

Михаил Михеев

Третье
издание
бестселлера!

Администрирование VMware vSphere 5

Виртуализация для профессионалов

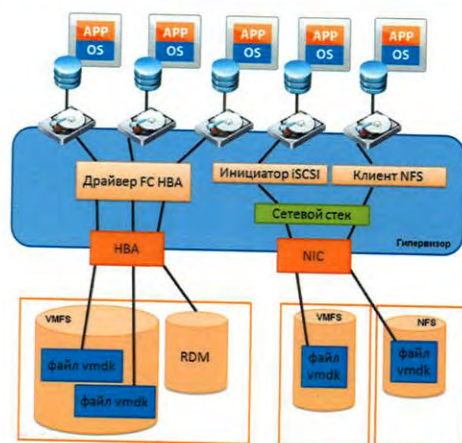
Настройка сети виртуальной инфраструктуры

Системы хранения данных

Управление ресурсами сервера

Мониторинг достаточности ресурсов

Защита данных и доступность виртуальных машин

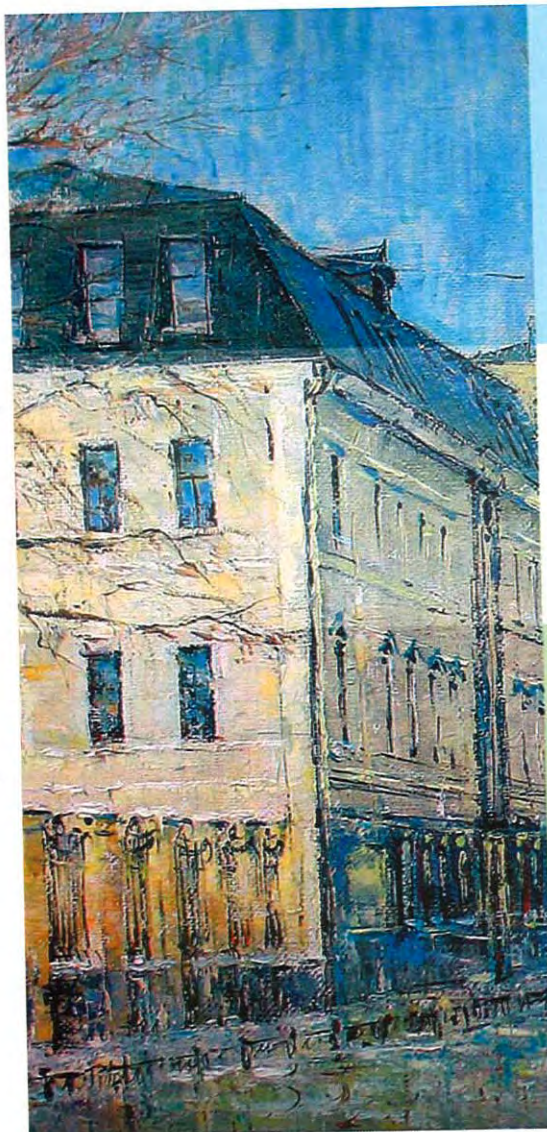


АДМИНИСТРИРОВАНИЕ СЕРВЕРА

Микроинформ — авторизованный учебный центр VMware

vmware®
PARTNER

AUTHORIZED
TRAINING CENTER



- первый в России авторизованный учебный центр VMware (с 2005 г.);
 - предлагает самый широкий спектр курсов VMware (как для заказчиков, так и для партнеров):
 - VMware vSphere;
 - VMware View;
 - VMware Site Recovery Management;
 - VMware vCenter Manager
- Учебный центр «Микроинформ» работает на ИТ-рынке России с 1988 г.;
 - специализируется в области авторизованного и корпоративного обучения;
 - проводит обучение в Москве и 40 регионах России, Украине и Казахстане;
 - Лицензия на образовательную деятельность;
 - Свидетельство о Государственной аккредитации.



Москва, 115184, ул. Малая Ордынка, 44
Тел. +7 (495) 953-0006

www.microinform.ru, www.facebook.com/microinform

Михеев М. О.

Администрирование VMware vSphere 5



Москва, 2013

УДК 005:004VMware vSphere 5
ББК 65.290c515
М69

Михеев М. О.

М69 Администрирование VMware vSphere 5. – М.: ДМК Пресс, 2012. – 504 с.: ил.
ISBN 978-5-94074-569-3

Книга посвящена вопросу работы с семейством продуктов VMware vSphere 5. В книге я постарался рассмотреть самые разнообразные моменты, с которыми можно столкнуться при работе с продуктом: здесь вы встретите описание требований и возможностей продуктов VMware, варианты настроек, необходимую для работы с продуктом информацию, в том числе из смежных областей знаний.

Начинается книга с описания того, что из себя представляет семейство продуктов vSphere, все компоненты этого набора продуктов и их возможности. Далее приводится информация о том, как этими возможностями воспользоваться: с точки зрения требований к инфраструктуре и необходимых настроек. Кроме того, приводятся глубокие технические подробности о принципах работы, способах мониторинга и диагностики неполадок. Наконец, дается информация по дополняющим сторонним продуктам, которые могут помочь в работе или решении проблем. Материал книги подается в виде пошаговых инструкций с достаточно подробной детализацией.

Издание будет полезно как начинающим, так и опытным системным администраторам. Последние могут использовать книгу как справочное пособие, позволяющее оперативно уточнить нюансы работы тех или иных механизмов, найти необходимые параметры и команды командной строки.

УДК 005:004VMware vSphere 5
ББК 65.290c515

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-94074-569-3

© Михеев М. О., 2012
© Оформление, ДМК Пресс, 2012

Содержание

Введение	12
Для кого?	12
О какой версии продукта?	12
Как книга организована?	12
Обратная связь	14
Предисловие	15
Глава 1	
Установка vSphere	16
1.1. Обзор	16
1.2. Установка и начало работы с ESXi	17
1.2.1. До установки	18
Варианты дистрибутивов	20
1.2.2. Установка ESXi	21
1.2.3. Автоматическая установка ESXi	25
1.2.4. Особенности установки ESXi	27
1.2.5. Auto Deploy	28
Установка Windows-версии Auto Deploy	29
Настройка vCenter	30
Настройка TFTP и DHCP сервера	30
Настройка Auto Deploy для первого сервера	31
Настройка Auto Deploy для последующих серверов	34
Обновление образа, загружаемого при помощи Auto Deploy	35
1.3. Вспомогательные компоненты vSphere	36
1.3.1. Image Builder	36
1.3.2. VMware Syslog Collector	39
1.3.3. VMware Core Dump Collector	41
1.4. Начало работы	44
1.4.1. Начало работы без vCenter	44
1.4.2. Установка Windows-версии vCenter Server	46
Системные требования vCenter	46
БД для vCenter Server	48
Совместимость vCenter Server 5 и vSphere Client с предыдущими версиями ESX(i) и vCenter	49
Установка vCenter Server	49

Linked Mode	51
1.4.3. vCenter Virtual Appliance	54
Различия между Windows- и Linux-версиями vCenter	54
Установка и настройка vCSA	55
1.5. Интерфейс клиента vSphere, vCenter, ESXi. Веб-интерфейс	56
1.5.1. Элементы интерфейса клиента vSphere при подключении к vCenter.....	56
Базовые шаги для решения проблем с клиентом vSphere	64
1.5.2. Первоначальная настройка vCenter и ESXi	65
Добавление серверов в консоль vCenter	65
Настройка лицензирования	66
Рекомендуемые начальные настройки ESXi	67
1.5.3. Работа через веб-интерфейс vSphere Web Client	69
Установка Web Client Server	70
1.5.4. vCenter Mobile Appliance, клиент для iPad, веб-интерфейс администратора.....	71
1.6. Основы работы из командной строки	71
1.6.1. Локальная командная строка ESXi и доступ по SSH	72
1.6.2. Microsoft PowerShell + VMware PowerCLI.....	76
Настройка PowerCLI.....	77
1.6.3. vSphere CLI, работа с vMA	78
1.6.4. Полезные команды	80
1.6.5. Полезные сторонние утилиты	81
Командная строка, SSH	81
Файловый менеджер	83
Вспомогательные утилиты.....	84
1.7. Сайзинг и планирование	85
1.7.1. Процессор.....	86
Выбор процессоров с точки зрения функционала.....	86
Что такое и зачем надо Intel-VT / AMD-V. Аппаратная поддержка виртуализации.....	88
1.7.2. Память.....	89
1.7.3. Дисковая подсистема	90
Расчет требуемого места на системе хранения	91
Производительность дисковой подсистемы	93
Выбор количества LUN	95
1.7.4. Сетевая подсистема	96
1.7.5. Масштабируемость: мало мощных серверов или много небольших?.....	98

Глава 2

Настройка сети виртуальной инфраструктуры	101
2.1. Основы сети ESXi, объекты виртуальной сети.....	101
2.1.1. Физические сетевые контроллеры, vmnic	104



2.1.2. Виртуальные контроллеры VMkernel	106
2.2. Стандартные виртуальные коммутаторы VMware – vNetwork Switch	110
2.3. Распределенные коммутаторы – vNetwork Distributed Switch, dvSwitch. Настройки	113
2.3.1. Основа понятия «распределенный виртуальный коммутатор VMware»	113
Сравнение стандартных и распределенных виртуальных коммутаторов	114
2.3.2. Добавление сервера в dvSwitch, настройки подключения vmnic	118
Нюансы задействования внешних подключений (Uplinks) dvSwitch	118
2.3.3. Группы портов на dvSwitch, добавление интерфейсов VMkernel	122
Добавление интерфейса VMkernel на dvSwitch	123
2.3.4. Уникальные настройки распределенных виртуальных коммутаторов	124
NetFlow	125
Port Mirroring	126
2.3.5. Уникальные настройки портов dvSwitch: Miscellaneous и Advanced	128
2.3.6. Миграция со стандартных виртуальных коммутаторов на распределенные	130
2.3.7. Технические особенности распределенных виртуальных коммутаторов VMware	134
2.3.8. Основы решения проблем dvSwitch	135
2.4. Настройки Security, VLAN, Traffic shaping и NIC Teaming	136
2.4.1. VLAN, виртуальные локальные сети. Настройка VLAN для стандартных виртуальных коммутаторов	136
EST, external switch tagging	139
VST, virtual switch tagging	139
VGT, virtual guest tagging	140
2.4.2. Настройка VLAN для dvSwitch. Private VLAN	141
Private VLAN, PVLAN	143
2.4.3. Security	145
2.4.4. Ограничение пропускной способности (Traffic Shaping)	147
2.4.5. NIC Teaming. Группировка сетевых контроллеров	148
2.4.6. Cisco Discovery Protocol, CDP и Link Layer Discovery Protocol (LLDP)	154
Настройка CDP для стандартных виртуальных коммутаторов	154
Настройка CDP и LLDP для распределенных виртуальных коммутаторов	155
2.5. Разное	156

2.5.1. Jumbo Frames	156
Настройка Jumbo Frames для виртуальных машин	156
Настройка Jumbo Frames для VMkernel	157
2.5.2. TSO – TCP Segmentation Offload, или TOE – TCP offload engine	158
2.5.3. VMDirectPath	159
2.5.4. Standalone (отдельные) порты	159
2.6. Рекомендации для сети	160

Глава 3

Системы хранения данных и vSphere	162
3.1. Обзор типов СХД	164
3.2. DAS	166
3.3. NAS (NFS)	167
3.3.1. Настройка и подключение ресурса NFS к ESXi	169
3.4. SAN, Fibre Channel	172
3.4.1. Адресация и multipathing	175
3.4.2. Про модули multipathing. PSA, NMP, MMP, SATP, PSP	177
3.4.3. Про зонирование (Zoning) и маскировку (LUN masking, LUN presentation)	183
3.5. SAN, iSCSI	184
3.5.1. Как настроить программный инициатор или аппаратный зависимый iSCSI на ESXi	186
Настройка сети для iSCSI	187
Включение iSCSI-инициатора и настройка Discovery	188
3.5.2. iSCSI Multipathing	191
3.6. VMFS, Virtual Machine File System	195
Корректное отключение LUN или удаление раздела VMFS	198
Технические особенности VMFS	198
3.6.1. Увеличение размера хранилища	
VMFS. Grow и Extent	201
VMFS Grow	201
VMFS Extent	202
3.6.2. Доступ к клонированному разделу VMFS, или к разделу VMFS с изменившимся номером LUN	203
3.7. RDM, Raw Device Mapping	206
3.8. NPIV	209
3.9. Адресация SCSI	211
3.10. vSphere API for Array Integration, VAAI. Интеграция и делегирование некоторых операций системам хранения данных	214
3.11. Profile-Driven Storage	216
3.12. VMware vSphere APIs for Storage Awareness, VASA	220
3.13. Virtual Storage Appliance	221
3.13.1. Ввод в VSA в эксплуатацию	222

3.13.2. Эксплуатация VSA	224
3.13.3. Размышления про применимость	225

Глава 4

Расширенные настройки, безопасность,

профили настроек, решение проблем	227
4.1. Расширенные настройки (Advanced settings)	227
4.2. Безопасность	229
4.2.1. Общие соображения безопасности	229
4.2.2. Брандмауэр ESXi	231
4.2.3. Аутентификация на серверах ESXi, в том числе через Active Directory	234
Вариант 1 – вам требуется подключаться напрямую	234
Вариант 2 – вам не требуется работать напрямую с ESX	235
Вариант 3 – вам требуется жестко запретить работу напрямую	235
4.2.4. Контроль доступа, раздача прав при работе через vCenter ...	236
Общие соображения по разграничению прав доступа	242
4.3. Настройка сертификатов SSL	243
4.4. Host Profiles	244
4.5. Использование SNMP	252
4.5.1. Настройка SNMP для vCenter	252
4.5.2. Настройка SNMP для серверов ESXi	254
4.6. Рекомендации по решению проблем	256
4.6.1. Статусные сообщения и файлы журналов (Logs&Events)	256
Events	256
Журналы	257
Экспорт журналов	257
Syslog	258
4.6.2. Онлайн-источники информации	259
4.6.3. Поддержка VMware	259
4.6.4. Core Dump, дампы	261
4.7. Время на сервере ESXi	261

Глава 5

Виртуальные машины	262
5.1. Создание ВМ. Начало работы с ней	262
5.2. Клонирование и шаблоны ВМ (Clone и Template)	268
5.2.1. Клонирование виртуальных машин	268
5.2.2. Шаблоны виртуальных машин (template)	269
5.2.3. Обезличивание гостевых ОС, SysPrep	271
5.2.4. Рекомендации для эталонных ВМ	275
5.3. Виртуальное оборудование ВМ	277
5.3.1. Memory	277

5.3.2. CPUs	278
5.3.3. IDE, PS2 controller, PCI controller, SIO controller, Keyboard, Pointing device	279
5.3.4. Video card	279
5.3.5. VMCI device, VM Communication Interface	280
5.3.6. Floppy drive	280
5.3.7. CD/DVD Drive	281
5.3.8. Network Adapter	281
TSO	283
Jumbo Frames	283
Large Ring Sizes	284
RSS	285
MSI-X	285
Резюме	286
MAC-адреса виртуальных машин	287
5.3.9. SCSI controller	288
5.3.10. Hard Disk	291
5.3.11. Parallel port	291
5.3.12. Serial port	291
5.3.13. SCSI device	292
5.3.14. USB controller и USB Device	293
5.3.15. VMDirectPath	294
5.4. Все про диски VM	297
5.4.1. Виртуальные диски – файлы vmdk	297
Тонкие диски и интерфейс	301
5.4.2. Изменение размеров дисков VM	302
Увеличение размера диска	302
Уменьшение номинального размера thin- или thick-диска	302
Уменьшение реального размера thin-диска	305
Удаление диска	307
5.4.3. Выравнивание (alligment)	308
Выравнивание VMFS	309
Выравнивание файловой системы гостевой ОС	310
5.4.4. Raw Device Mapping, RDM	311
5.5. Настройки VM	314
General Options	314
vApp Options	314
VMware tools	314
Advanced ⇒ General	315
Advanced ⇒ CPUID Mask	316
Advanced ⇒ Memory / CPU hotplug	316
Advanced ⇒ Boot Options	316
Advanced ⇒ Fibre Channel NPIV	316
CPU/MMU Virtualization	317
Swapfile Location	317

5.6. Файлы ВМ, перемещение файлов между хранилищами	317
Файл VMX	318
Файл NVRAM	319
Файл подкачки VSWP	319
Файлы VMDK	319
Перемещение файлов ВМ	324
5.7. Снимки состояния (Snapshot)	325
Файлы VMDS	328
Файлы vmsn	329
Файлы -delta.vmdk	329
Плюсы и минусы снимков состояния	331
5.8. VMware tools	333
5.9. vAPP	337
Резюме	339

Глава 6

Управление ресурсами сервера.

Мониторинг достаточности ресурсов.

Живая миграция ВМ. Кластер DRS

6.1. Настройки распределения ресурсов для ВМ. Пулы ресурсов	340
6.1.1. Настройки limit, reservation и shares для процессоров и памяти	340
Limit, reservation и shares для процессора	340
Limit, reservation и shares для памяти	342
Иллюстрация работы механизма распределения ресурсов на примере памяти	347
6.1.2. Пулы ресурсов	349
6.1.3. Рекомендации по настройкам Limit, Reservation и Shares	353
Когда ресурсов в достатке	355
Когда ресурсов не хватает	355
6.1.4. Storage IO Control, SIOC для дисковой подсистемы	356
6.1.5. Network IO Control, NIOC и traffic shaping для сети	360
6.2. Механизмы перераспределения ресурсов в ESXi	362
6.2.1. CPU	362
NUMA	366
6.2.2. Memory	367
Несколько общих слов	367
Memory Overcommitment	367
Выделение по запросу	369
Transparent Memory Page Sharing	370
Перераспределение памяти. Balloon driver, memory compression и vmkernel swap	376
Balloon Driver	376
Memory compression	378

VMkernel swap.....	378
Host Cache Configuration	380
Насколько часто три описанных механизма применяются к разным группам виртуальных машин?	380
Нехватка памяти на всех – какой механизм будет использован?	382
6.2.3. Disk.....	385
6.2.4. Net.....	386
6.3. Мониторинг достаточности ресурсов	386
6.3.1. Источники информации о нагрузке	386
Вкладка Performance и другие источники информации через клиент vSphere	387
Пулы ресурсов.....	389
Хранилища, Storage Views.....	391
esxstop и resxstop.....	391
Анализ информации от (r)esxstop	395
Perfmon «внутри» гостевой ОС.....	396
6.3.2. Какие счетчики нас интересуют и пороговые значения	398
CPU	399
MEMORY.....	402
DISK	403
Network.....	404
6.4. Механизм Alarm	404
6.5. Миграция выключенной (или suspend) виртуальной машины.....	410
6.6. Storage vMotion – живая миграция файлов ВМ между хранилищами	411
6.7. vMotion – живая миграция ВМ между серверами.....	412
6.8. Кластер DRS. DPM.....	418
DMP, Distributed Power Management	429
Удаление DRS.....	432
Advanced Settings	432
6.9. Кластер Storage DRS	433

Глава 7

Защита данных и повышение доступности

виртуальных машин	436
7.1. Высокая доступность виртуальных машин.....	436
7.1.1. VMware High Availability, HA.....	437
Условия для HA	437
Какие настройки доступны для кластера HA.....	438
Admission Control	441
Как работает HA.....	448
Изоляция. Datastore Heartbeating	450
VM Monitoring	454

Advanced Options	455
Использование HA и DRS вместе	457
7.1.2. VMware Fault Tolerance, FT	458
Настройка VMware FT	459
Настройка инфраструктуры и включение FT	461
Как работает VMware FT	466
7.2. Управление обновлениями виртуальной инфраструктуры, VMware Update Manager	469
7.2.1. Установка обновлений в командной строке локальной, удаленной и PowerCLI	469
Локальная командная строка	469
PowerCLI	470
vSphere CLI	470
7.2.2. VMware Update Manager	471
Установка VUM	471
Как работает VUM	472
7.3. Резервное копирование и восстановление	481
7.3.1. Резервное копирование ESXi и vCenter	481
Резервное копирование vCenter	481
Резервное копирование настроек ESXi	482
7.3.2. Резервное копирование виртуальных машин	482
Типы данных для резервного копирования	482
Подходы к организации резервного копирования	485
Агент резервного копирования в гостевой ОС	485
Бесплатные средства или сценарии	486
Средства, поддерживающие vStorage API for Data Protection	486
7.3.3. VMware Data Recovery	488
Первоначальная настройка	489
Настройка задания резервного копирования	491
Восстановление виртуальной машины из резервной копии VMware Data Recovery	492
Восстановление файлов гостевой ОС из резервной копии VMware Data Recovery	492
Факты про VDR	494
Предметный указатель	496



Введение

Последние несколько лет тема серверной виртуализации привлекает внимание все большего количества компаний и технических специалистов. Виртуализация позволяет добиться финансовых выгод для компании, значительного упрощения работы для системных администраторов. Сегодня самым интересным решением для виртуализации серверов является флагманское семейство продуктов компании VMware – VMware vSphere 5.

Гипервизор ESXi, часть vSphere, обладает очень интересными возможностями по виртуализации, балансировке нагрузки на подсистемы одного сервера и балансировке нагрузки между серверами, а также повышению доступности приложений, выполняемых в виртуальной среде. Однако чтобы начать в полной мере пользоваться всеми функциями vSphere, понадобятся определенные знания. Еще до того, как даже начать установку ESXi на сервер, стоит задуматься о многих вещах, например об ограничениях по выбору оборудования, и от чего зависят требования к производительности.

Кроме того, нелишними будут знания из некоторых смежных областей, таких как системы хранения данных, сети, особенности серверного оборудования. Все эти темы в достаточной мере раскрываются в данной книге простым и понятным языком.

Для кого?

Данная книга касается большинства аспектов серверной виртуализации, подача материала рассчитана на неподготовленных системных администраторов. В силу полноты описываемых тем интересна она будет и администраторам с опытом работы в области виртуализации, в частности как справочное пособие.

О какой версии продукта?

На момент написания данной книги актуальной версией являлась vSphere 5. Тем не менее большая доля и, скорее всего, вся информация книги будет актуальна для всех обновлений пятой версии виртуальной инфраструктуры VMware.

Как книга организована?

Глава 1. Установка vSphere. Первая глава посвящена самому началу – что такое VMware vSphere 5? Какие продукты входят в это семейство? Какие вспомогательные продукты предлагает нам VMware? Какие существуют сторонние продукты, способные облегчить жизнь администратору? Объясняется, каким нюансам следует уделить внимание при выборе оборудования для vSphere. Даются основные ответы на один из наиболее популярных вопросов – «Сервер какой про-

изводительности (или сколько серверов) необходимо для запуска на нем ESXi?». Разбирается установка с нуля. Автоматизация установки для упрощения массового развертывания. После установки – этап первоначальной настройки. Дается основная информация по элементам графического интерфейса и выполнению манипуляций с vSphere из командной строки.

Глава 2. Настройка сети виртуальной инфраструктуры. В этой главе приводится полная информация об организации сети для виртуальной инфраструктуры. Что собой представляют для гипервизора физические сетевые контроллеры? Виртуальные коммутаторы, стандартные и распределенные – что необходимо знать для уверенного использования этих объектов? Какие настройки для них возможны? Рассказывается про виртуальные сетевые контроллеры, принадлежащие самому гипервизору. Дается необходимая информация для планирования схемы сети виртуальной инфраструктуры.

Глава 3. Системы хранения данных и vSphere. Для большинства инфраструктур VMware vSphere используется внешнее хранилище данных, SAN или NAS. Администратору vSphere следует понимать, какими возможностями обладают системы хранения Fibre Channel, iSCSI и NFS относительно ESXi. Есть нюансы, которые необходимо знать для планирования и начала работы с системой хранения того или иного типа. Это возможности и настройки multipathing, программный инициатор iSCSI, нюансы адресации SCSI.

ESXi размещает виртуальные машины на своей собственной файловой системе VMFS. В этой главе приводится подробная информация по нюансам, возможностям и ограничениям этой файловой системы.

Глава 4. Расширенные настройки, профили настроек и безопасность. Достаточно важной темой является безопасность. В данной главе описываются основные аспекты обеспечения безопасности виртуальной инфраструктуры. Приводится процедура настройки брандмауэра, описывается модель контроля доступа и раздачи прав. Также приводится основная информация касательно сертификатов и их установки для ESXi, vCenter Server и Update Manager.

Отдельным подразделом описывается механизм Host Profiles, задачами которого являются тиражирование настроек и отслеживание соответствия назначенному профилю настроек для серверов ESXi.

Глава 5. Виртуальные машины. В данной главе приводится вся информация о виртуальных машинах. Способы их создания, в первую очередь механизмы vCenter для работы с шаблонами и клонирования. Дана подробная информация о виртуальном оборудовании, его возможностях и ограничениях. Особенно подробно разбираются возможности виртуальных дисков, в частности thin provisioning.

Виртуальная машина – это набор файлов. Разумеется, в этой главе есть отдельный раздел, посвященный описанию того, из каких файлов состоит виртуальная машина.

Приводятся список доступных для виртуальной машины настроек и их описание. Дается подробная информация о том, что такое снимки состояния виртуальной машины и в каких ситуациях ими стоит пользоваться, а в каких – избегать.

Глава 6. Управление ресурсами сервера. Мониторинг достаточности ресурсов. Живая миграция VM. Кластер DRS. В этой главе подробно рассматривается потребление ресурсов инфраструктуры, притом со всевозможных сторон.

Сильной стороной продуктов vSphere являются очень гибкие возможности по работе с ресурсами. Притом существуют как механизмы эффективной утилизации и перераспределения ресурсов одного сервера, так и возможность создать кластер DRS, который будет балансировать нагрузку между серверами ESXi при помощи живой миграции виртуальных машин между ними. У администраторов существуют весьма гибкие настройки того, как ESXi должен перераспределять ресурсы сервера или серверов. Наконец, vSphere предоставляет весьма гибкие возможности по анализу текущей ситуации потребления ресурсов и нахождению узких мест.

Все эти темы последовательно и подробно разбираются в данной главе.

Глава 7. Защита данных и повышение доступности виртуальных машин. Защита данных и повышение доступности – это те темы, без обсуждения которых обойтись невозможно. И для того, и для другого администраторы виртуальной инфраструктуры могут применять разнообразные средства.

В данной главе приводится подробная информация по настройке, использованию и нюансам работы с теми средствами повышения доступности, что предлагает компания VMware. Кроме того, разбираются разнообразные решения и подходы к резервному копированию.

Обратная связь

Адрес моей электронной почты – Mikhail.Mikheev@vm4.ru. Смело пишите.

Предисловие

С момента первого чтения курса по VMware ESX Server (еще второй тогда версии) в 2005 году я наблюдаю все более широкий интерес к теме виртуализации. В сентябре 2007 года я начал вести свой блог (<http://vm4.ru>), с помощью которого делился новой информацией, особенностями и нюансами работы с виртуальной инфраструктурой VMware. Этот опыт получился достаточно удачным, росли и посещаемость блога, и число специалистов, с которыми устанавливался контакт, как онлайн, так и офлайн. Однако, несмотря на хорошую посещаемость блога и постоянную переписку с читателями блога и слушателями курсов, я видел, что существует нехватка доступного и полного источника информации по данной теме. Так родилась идея написать книгу, которая смогла бы стать как средством знакомства с виртуализацией для новичков, так и настольным справочником для профессионалов. Собственно, ее вы и держите в руках.

Первый тираж книги назывался «Администрирование VMware vSphere 4.0». Затем произошло большое обновление vSphere, появились новые возможности, и второй тираж вышел уже исправленным и дополненным, по новой версии vSphere 4.1.

А сейчас я обновил материал в соответствии с изменениями в vSphere версии 5. Так что это уже третье издание, исправленное и дополненное.

Я хочу выразить благодарность людям, чьи отзывы помогли мне сделать эту книгу лучше: Родиону Тульскому, Андрею Цыганку, Виталию Савченко, Владиславу Кирилину, Дмитрию Тиховичу, Антону Жбанкову, Евгению Ковальскому, Евгению Киселеву, Сергею Щадных, Марии Сидоровой.

Особенно хочу выразить благодарность:

Артему Проничкину, труд которого был поистине титаническим. Он был первым человеком, которого я попросил вычитать книгу и высказать комментарии и рекомендации. И сразу попадание в яблочко! Артем вдумчиво прочел все и обратил мое внимание на множество проблемных мест. Более того, некоторые материалы были написаны им самим. Огромное персональное спасибо.

Роману Хмелевскому, автору блога blog.aboutnetapp.ru. Роман — крупный специалист в области систем хранения данных, очень много и по делу помог мне с написанием соответствующего материала.

Дмитрию Прокудину — человеку, который активнее всех откликнулся на мой призыв сообщать об ошибках, опечатках и неточностях в вышедшей книге. Дмитрий, в этой книге вашему меткому глазу брошены новые вызовы ☺.

Отдельное спасибо хочу выразить моей супруге Анюте, которая была вынуждена делить меня с работой над книгой в течение года. Потом еще несколько месяцев на обновление перед вторым тиражом. Затем еще несколько месяцев перед последним обновлением. Без помощи семьи у меня не получилось бы все это сделать.

Глава 1. Установка vSphere

Эта часть книги посвящена установке *VMware vSphere 5*, некоторых сопутствующих программ и связанным с установкой вопросам.

1.1. Обзор

Под vSphere понимаются следующие продукты: *VMware ESXi* и *VMware vCenter Server*.

ESXi – это *гипервизор*. Так называется программное обеспечение, создающее виртуализацию.

vCenter Server – это приложение, являющееся средством централизованного управления виртуальной инфраструктурой, то есть всеми ESXi, созданными на них сетями, виртуальными машинами и прочим.

В пятой версии vSphere есть два варианта vCenter Server – в виде привычного Windows-приложения и в виде так называемого vCenter Virtual Appliance – предустановленной виртуальной машины с предустановленной Linux-версией vCenter Server. Разница между этими вариантами, инструкции по развертыванию и рекомендации будут даны в посвященном vCenter разделе.

Под сопутствующими программами в первую очередь понимается разного рода официальное ПО VMware под vSphere. Некоторые приложения поставляются прямо в дистрибутиве vSphere, некоторые доступны отдельно.

Поставляющиеся в комплекте:

- ❑ vCenter Update Manager – утилита для удобного обновления ESXi, гостевых ОС и приложений в гостевых ОС;
- ❑ vSphere Web Client Server – приложение, обеспечивающее веб-интерфейс для взаимодействия с виртуальными машинами на vSphere;
- ❑ ESXi Dump Collector – служба сбора диагностической информации (дампов) с серверов ESXi после критического сбоя (PSOD, пурпурный экран смерти);
- ❑ Syslog Collector – Windows-служба централизованного сбора файлов журналов с серверов ESXi;
- ❑ Auto Deploy – служба организации PXE загрузки серверов ESXi;
- ❑ vSphere Authentication Proxy – пригодится, если вы выберете вариант PXE-загрузки ESXi, и эти ESXi надо будет ввести в домен Active Directory.

Поставляющиеся отдельно:

- ❑ vSphere CLI (Command Line Interface) – удаленная командная строка. Предоставляет централизованный интерфейс командной строки к ESXi и ESX. Командная строка может понадобиться для решения проблем, для

автоматизации каких-то действий через сценарии. vSphere CLI доступен в вариантах под Linux и под Windows;

- ❑ vSphere Management Assistant (vMA) – это Virtual Appliance, то есть готовая к работе виртуальная машина с Linux, которая содержит в себе разного рода компоненты, призванные упростить некоторые задачи администрирования виртуальной инфраструктуры. В частности, в ее состав входит vSphere CLI;
- ❑ Power CLI – дополнение к Microsoft PowerShell, позволяющее управлять виртуальной инфраструктурой при помощи этого мощного языка сценариев;
- ❑ VMware Data Recovery – решение VMware для резервного копирования;
- ❑ VMware Converter – эта программа поможет нам получить VM из:
 - другой VM, в формате другого продукта VMware, не ESXi;
 - другой VM, в формате продукта виртуализации другого производителя;
 - другой VM, в формате ESX. Это может потребоваться для удобного изменения некоторых свойств данной VM. Например, для уменьшения размера ее диска;
 - образа диска, снятого с физического сервера. Поддерживаются образы, созданные при помощи Norton Ghost, Symantec LiveState, Symantec Backup Exec System Recovery, StorageCraft ShadowProtect, Acronis True Image;
 - резервной копии VM, полученной с помощью VCB;
 - наконец, из физического сервера. То есть осуществить его миграцию в VM. Такой процесс часто называют p2v – physical to virtual. (По аналогии существуют процессы v2v – первые три пункта этого списка – и v2p – для миграции с виртуальных машин на физические. Средства последнего рода VMware не предоставляет.)

Также упомяну про некоторые сторонние продукты и утилиты, которые кажутся интересными лично мне.

Наконец, приведу соображения по сайзингу – подбору конфигурации сервера (и не только, еще коснемся СХД).

У vSphere существуют несколько вариантов лицензирования, в том числе бесплатная лицензия для ESXi. В книге я рассказываю про все существующие функции, так что делайте поправку на ограничения используемой вами лицензии.

Иногда я буду упоминать как первоисточник информации документацию в общем или конкретные документы. Подборка основных источников информации доступна по ссылке <http://link.vm4.ru/docs>. Я настоятельно рекомендую взять эти источники на вооружение.

1.2. Установка и начало работы с ESXi

Здесь мы поговорим про требования к оборудованию и дистрибутивы – этими вопросами следует озаботиться до установки как таковой. Далее разберем важные шаги установки ESXi, начало работы. Затем – более сложные варианты установки: обновление с предыдущих версий и автоматическую установку ESXi.

ESXi – это операционная система. Установка ее мало чем отличается от установки других ОС, разве что своей простотой – вследствие узкой специализации этой ОС. Тем не менее на некоторые моменты обратить внимание стоит.

Для справки:

- ❑ давайте договоримся называть физический сервер, на котором установлен ESXi, как **Host, Хост**;
- ❑ вам в изобилии будет попадаться название VMkernel. VMkernel – это название компонента ESXi, который «делает виртуализацию». В каком-то смысле оправданно сказать, что ESXi состоит из VMkernel и Linux. VMkernel занимается абсолютно всем, что связано с виртуализацией; Linux занимается всякой прочей мелочевкой. VMkernel – неотъемлемая часть ESXi, поэтому нередко VMkernel и ESXi можно воспринимать как синонимы. Например, если в тексте встретилось «интерфейс управления VMkernel», это означает то же самое, что и «интерфейс управления ESXi».

1.2.1. До установки

Перед разговором об установке ESXi имеет смысл поговорить об оборудовании, на котором он будет работать.

Первое – неплохо, если ESXi будет поддерживать это оборудование. Это гарантирует наличие драйверов и возможность обращаться за поддержкой в случае возникновения проблем. На сайте VMware легко находятся HCG – Hardware Compatibility Guides, списки совместимости (<http://vmware.com/go/hcl>). Таких списков несколько, например:

- ❑ Systems/Server – перечисление поддерживаемых моделей серверов;
- ❑ I/O Devices – список поддерживаемых контроллеров;
- ❑ Storage/SAN – список поддерживаемых систем хранения.

Большая часть «брендового» оборудования в этих списках присутствует, проблемы обычно возникают при желании сэкономить. Кроме того, в Интернете можно отыскать неофициальные списки совместимости. Они не могут повлиять на поддержку производителя, но помогут не выбрать заведомо несовместимых компонентов.

Основная проблема – в поддержке дискового контроллера. Здесь надо иметь в виду: сам ESXi можно установить на разнообразные контроллеры ATA, SATA, SAS и SCSI, а также HBA FC и iSCSI. Заметьте, не «на любые», а на «разнообразные». Список поддерживаемых легко найти в документе по вышеупомянутой ссылке <http://vmware.com/go/hcl>.

Обратите внимание. Если очень-очень надо установить ESXi на сервер, к оборудованию которого нет штатных драйверов, можно попробовать найти официальные драйверы и интегрировать их в дистрибутив при помощи Image builder (см. раздел 1.3.1). Кроме того, существуют неофициальные драйверы и неофициальные способы добавить их в дистрибутив – см. утилиту ESXi Customizer (<http://esxi-customizer.v-front.de>). Разумеется, последнее – на свой страх и риск.

Однако использовать дисковые ресурсы для работы ВМ можно на более ограниченном количестве моделей контроллеров. То есть возможна ситуация, когда у вас сам ESXi установлен на локальные диски сервера, подключенные к дешевому и/или встроенному контроллеру. Но оставшееся свободным место на этих дисках задействовать под ВМ не получится. Так что поддержка контроллеров ATA и SATA, появившаяся еще в ESXi 4, не означает, что ими можно ограничиться.

В большинстве случаев при использовании встроенных контроллеров ESXi заработает с ними как с дисковыми контроллерами, но не как с контроллерами RAID – то есть увидит отдельные диски, не RAID-массив.

Вывод: читайте внимательно списки совместимости и руководства для начинающих. Или заранее попробуйте, если образец комплектующих есть под рукой.

Я не привожу конкретных списков лишь потому, что такие списки имеют обыкновение меняться с выходом обновлений ESXi. Напомню, что искать следует по адресу <http://vmware.com/go/hcl>.

Особняком стоит возможность установить ESXi на флэш-накопитель. Такой вариант интересен тем, что все место на дисках мы отводим под виртуальные машины, сама ОС (ESXi) установлена отдельно. Локальных дисков вообще может не быть – достаточно флэш-накопителя. Еще такой вариант иногда интересен для обслуживания удаленных площадок. При необходимости установить там сервера ESXi можно: вначале установить ESXi локально на накопитель USB (например, в ВМ под управлением VMware Workstation или Player), а затем отправить на удаленную площадку лишь флэшку с установленным и настроенным ESXi.

Перед тем как устанавливать ESXi на сервер, имеет смысл обновить всевозможные BIOS и firmware сервера и всех контроллеров. Это действительно может помочь решить (или избежать) проблем. В идеале, конечно, имеет смысл обратиться на сайт производителя сервера и посмотреть – вдруг есть рекомендации использовать (или ни в коем случае не использовать) какую-то конкретную версию прошивки под вашу версию ESXi.

Еще один небольшой совет: в моей практике были ситуации, когда на вроде бы совместимом сервере ESXi работал не так, как ожидалось (на этапе установки в том числе). Несколько раз в таких ситуациях помогал сброс настроек BIOS на значения по умолчанию. Иногда помогал только аппаратный сброс настроек, перестановкой джамперов.

Еще несколько слов следует сказать про процессор. Если мы хотим использовать на сервере ESXi 5, то процессоры этого сервера должны быть 64-битными (x86-64). Это неактуально для новых серверов (последние годы все процессоры Intel и AMD поддерживают работу в 64-битном режиме), но если вы планируете задействовать какой-то сервер в возрасте – этот момент необходимо учесть. Проверить 64-битность процессора можно несколькими путями:

- ☐ узнать его модель и посмотреть описание на сайте производителя;
- ☐ попробовать запустить на этом сервере установку – если процессор не подходит, установщик сообщит нам об этом;
- ☐ наконец, с сайта VMware можно загрузить небольшую утилиту под названием CPU Identification Utility. Найти ее можно в разделе **Download**

⇒ **Drivers and tools.** Эта утилита сообщит вам о возможности работы процессора в 64-битном режиме, поможет узнать, совместимы ли процессоры нескольких серверов для vMotion, о поддержке EVC (Enhanced vMotion Compatibility).

Кроме 64-битного режима, процессоры могут обладать аппаратной поддержкой виртуализации – Intel-VT или AMD-V. Она является необходимой для запуска 64-битных гостевых ОС. Небольшой нюанс здесь в следующем: поддержка этой функции включается и выключается в BIOS сервера, так что возможна ситуация, когда процессор ее поддерживает, но запустить 64-битную ВМ вы не можете из-за того, что эта функция выключена. Разные производители в разных BIOS называют ее по-разному. Обычно «Hardware Virtualization», «Intel-VT», «AMD-V».

Проверить состояние аппаратной поддержки виртуализации можно, выполнив в локальной командной строке ESXi команду

```
esxcfg-info | grep HV
```

Выводы интерпретируются следующим образом:

- 0 – поддержка Intel VT/AMD-V недоступна на данном сервере;
- 1 – технология Intel VT/AMD-V доступна, но не поддерживается на данном сервере;
- 2 – поддержка Intel VT/AMD-V доступна для использования, но не включена в BIOS;
- 3 – поддержка Intel VT/AMD-V включена в BIOS и доступна для использования.

Еще нюанс: для функции VMware Fault Tolerance необходимы процессоры из списка <http://kb.vmware.com/kb/1008027>. Далее для Fault Tolerance и vMotion нужно, чтобы набор поддерживаемых процессорами инструкций был одинаков для серверов, между которыми мы хотим использовать эти функции. Более подробно данные вопросы будут разобраны в разделе, посвященном сайзингу.

Варианты дистрибутивов

Правильное место для обзаведения дистрибутивами продуктов – сайт VMware. Доступ к ним можно получить после бесплатной регистрации. Ссылки на загрузку придут на указанный адрес электронной почты.

Эти дистрибутивы полнофункциональны, то есть никаких ограничений по сроку действия в дистрибутив как таковой не встроено. Однако для того, чтобы хоть что-то заработало, нам потребуется лицензия.

Для ознакомительных и демонстрационных целей можно воспользоваться временной лицензией. Она «встроена» в дистрибутив как ESXi, так и vCenter. Это означает, что для ESXi или vCenter можно указать тип лицензии «Evaluation». И 60 дней абсолютно все функции будут работать (как если бы к этим продуктам была применена максимальная лицензия Enterprise Plus).

По истечении этих 60 дней необходимо указать купленную лицензию или переустановить ESXi или vCenter Server (они лицензируются независимо друг от

друга). Когда для ESXi не указано работающей лицензии – все установится и почти все настроится (кроме функций, требующих отдельных лицензий, типа DRS). Вы сможете создать и настроить VM на ESXi без действующей лицензии – но не сможете эти VM включить.

Общее представление о том, какие функции в каких лицензиях доступны, можно получить на официальном сайте VMware (подборка ссылок доступна тут: <http://link.vm4.ru/lic>).

Еще одна небольшая тонкость, касающаяся ESXi. С сайта вы можете загрузить «Installable» версию ESXi, которая предназначена для установки на HDD/LUN/Flash. Но еще один вариант – приобрести сервер со встроенной флэшкой или отдельно флэш-накопитель, где ESXi уже установлен. Такой вариант называется «ESXi Embedded».

При выборе между вариантами Installable и Embedded (то есть между установкой ESXi на HDD/LUN/флэш-накопитель или готовый флэш-накопитель) ориентируйтесь на свои вкусы и привычки.

Хочу акцентировать ваше внимание на то, что версия Installable устанавливается не только на локальные диски, но и на флэш-накопитель или USB-HDD. Также из версии Installable можно самостоятельно извлечь образ, который затем залить на флэшку и загружать ESXi с нее. Последний вариант является официально не поддерживаемой конфигурацией. Но с его помощью можно подготовить загрузочную флэшку, даже если целевой сервер недоступен (подробности можно найти по ссылке <http://www.vm4.ru/2010/01/all-about-esxi.html>).

Еще одним вариантом является загрузка ESXi по PXE. В пятой версии vSphere появился отдельный продукт vSphere Auto Deploy, реализующий эту задачу. О нем будет рассказано в соответствующем разделе.

Последний нюанс – в списке дистрибутивов на сайте VMware или на сайтах таких производителей оборудования, как Dell, IBM и HP, можно найти что-то вроде «ESXi HP (IBM, DELL и прочее) Edition». Кратко это означает сборку ESXi, в которую входят нестандартные драйверы и CIM Provider – компоненты, предоставляющие интерфейс для мониторинга оборудования серверов конкретного производителя. Благодаря этому такой ESXi может заработать на серверах, на которых не заработает стандартный ESXi. Кроме того, данных мониторинга будет больше, чем если на сервере будет установлена обычная версия ESXi от VMware, и эти данные могут быть собраны центральным сервером управления и мониторинга оборудования (если таковой используется в вашей инфраструктуре).

1.2.2. Установка ESXi

Здесь мы разберем все аспекты обычной установки ESXi.

ESXi – это операционная система. Установка ее мало чем отличается от установки других ОС, разве что своей простотой – вследствие узкоспециализированности ESXi. Тем не менее на некоторые моменты обратить внимание стоит.

Для установки ESXi вам потребуются диск с дистрибутивом и доступ к локальной консоли сервера.

Загружаем сервер с этого диска, запускается мастер установки. Вопросов он задаст всего ничего:

1. Формально говоря, первый вопрос – принимаем ли мы лицензионное соглашение. С вашего позволения, я не буду давать рекомендаций по ответу на этот вопрос.
2. Выбор диска для установки. Тут есть нюансы, но в большей степени нюансы планирования – хотим ли мы устанавливать ESXi на локальный диск, на диск в системе хранения данных или флэш-накопитель? Ну а технический нюанс только один – выбрать на этом шаге мастера правильный, именно тот диск, куда ESXi должен быть установлен.
3. Выбор раскладки клавиатуры. Варианты, отличные от предлагаемого по умолчанию, нас не интересуют.
4. Указание пароля для пользователя root. Именно с этим пользователем и его паролем мы будем взаимодействовать с сервером на первых этапах.

Остановимся на некоторых моментах подробнее.

Первый из них – это выбор диска, на который будем устанавливать ESXi. Проблема здесь может возникнуть в том случае, если инсталлятору видно больше одного диска/LUN. Такое обычно происходит в инфраструктурах покрупнее, когда сервер, на который мы устанавливаем ESXi, подключен к системе хранения данных, СХД. В силу условий работы vMotion и других функций vSphere часть LUN системы хранения должна быть доступна всем серверам (рис. 1.1).

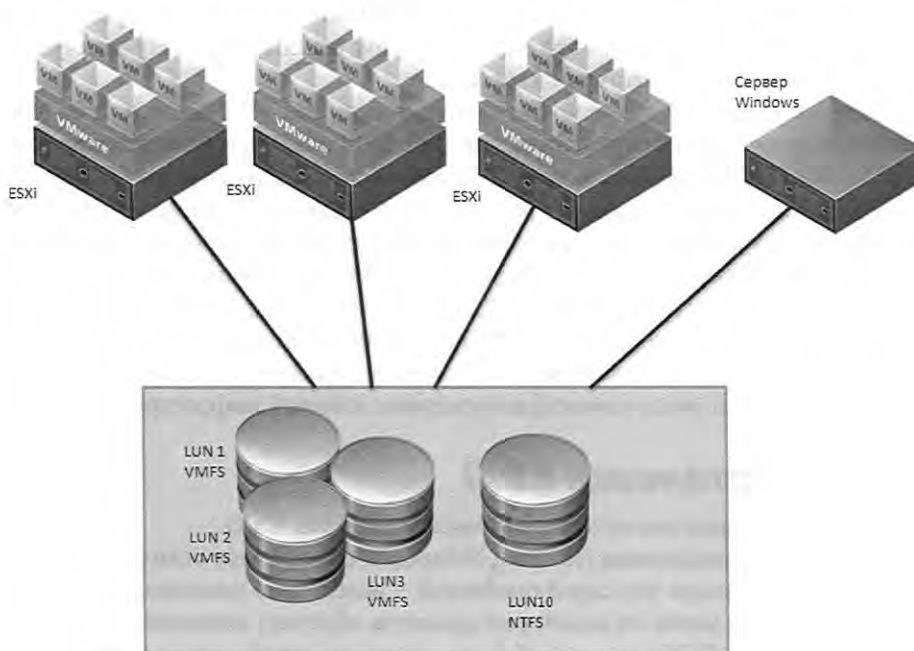


Рис. 1.1. Схема использования СХД несколькими серверами

СХД используется несколькими серверами ESXi и сервером Windows. Притом в некоторых ситуациях мы вынуждены будем сделать так, что серверам ESXi будут доступны все четыре LUN с этой СХД – даже тот, с которым работает физический Windows-сервер. Например, такая конфигурация потребуется, если используется отказоустойчивый кластер Майкрософт в конфигурации physical-2-virtual. Таким образом, при установке ESXi на один из этих серверов он увидит все четыре LUN (а еще и локальные диски сервера, если они есть) – и минимум один (тот, который используется Windows-сервером) инсталлятору покажется пустым.

Так вот, если инсталлятору ESXi доступны несколько дисков, то не всегда бывает тривиально на этапе установки определить, на какой же из них нам необходимо проинсталлировать ESXi, а на какой/какие ставить нельзя, потому что это разрушит данные на них, а нам оторвут за это голову.

Обратите внимание: ESXi на этом этапе покажет нам физический адрес устройства – вида «vmhba0:C0:T0:L0». Последняя цифра здесь – номер LUN, по которому, как правило, и можно сориентироваться.

Плюс к тому в пятой версии ESXi установщик делит видимые диски на локальные (Local) и внешние (Remote), так что в актуальной версии не ошибиться намного проще.

В итоге проблема в том, что по ошибке мы можем выбрать диск, на котором уже лежат данные, и эти данные затереть.

Эта проблема актуальна в подавляющем большинстве случаев, если мы используем внешнюю СХД.

Отсюда вывод:

Если доступных для ESXi дисков/LUN мало и мы гарантированно отличим предназначенный для установки – просто устанавливаем на него ESXi.

Иначе лучше подстраховаться. Если у нас внешняя СХД и слова «зонирование» и «маскировка» (zoning, LUN masking/presentation) нам ни о чем не говорят, то надежнее физически отключить сервер от всех LUN, кроме того одного, на который будем устанавливать ESXi.

Если маскировку мы используем, то можно с ее помощью спрятать все LUN, кроме нужного, на время установки.

Сразу после установки ESXi необходимо настроить сеть. Делается это очень просто: нажимаем **F2**, попадаем в меню а-ля BIOS. Нам нужен пункт **Configure Management Network**. Там выставляем правильные настройки IP, DNS-имя и домен. Но самое главное – нам нужен пункт **Network Adapters**, в котором мы выберем один сетевой контроллер, через который будет выходить наружу управляющий интерфейс ESXi (рис. 1.2).

Ошибиться мы можем, в случае если в сервере несколько сетевых карт и они смотрят в разные сети, – см. рис. 1.3.

Нам необходимо выбрать тот физический сетевой контроллер (они именуются vmnic#), который ведет в сеть управления. На рис. 1.3 я изобразил, что в этой сети находится ноутбук – с которого, предполагается, вы будете управлять ESXi. Или в этой сети находится vCenter, через который вы будете управлять ESXi.

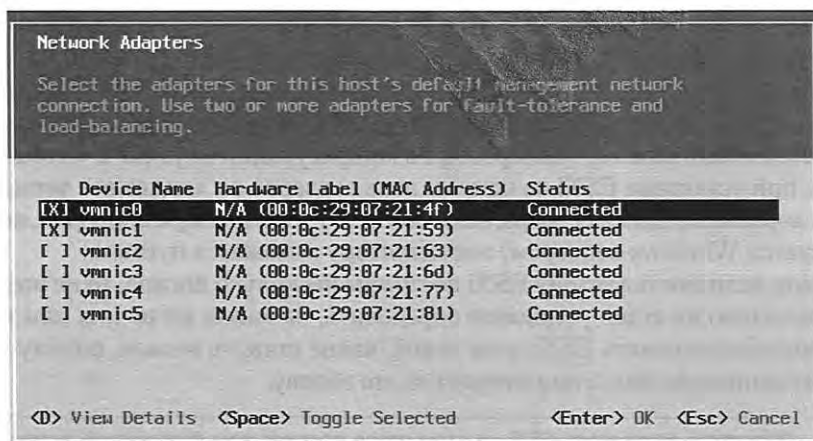


Рис. 1.2. Выбор контроллера для сети управления ESXi

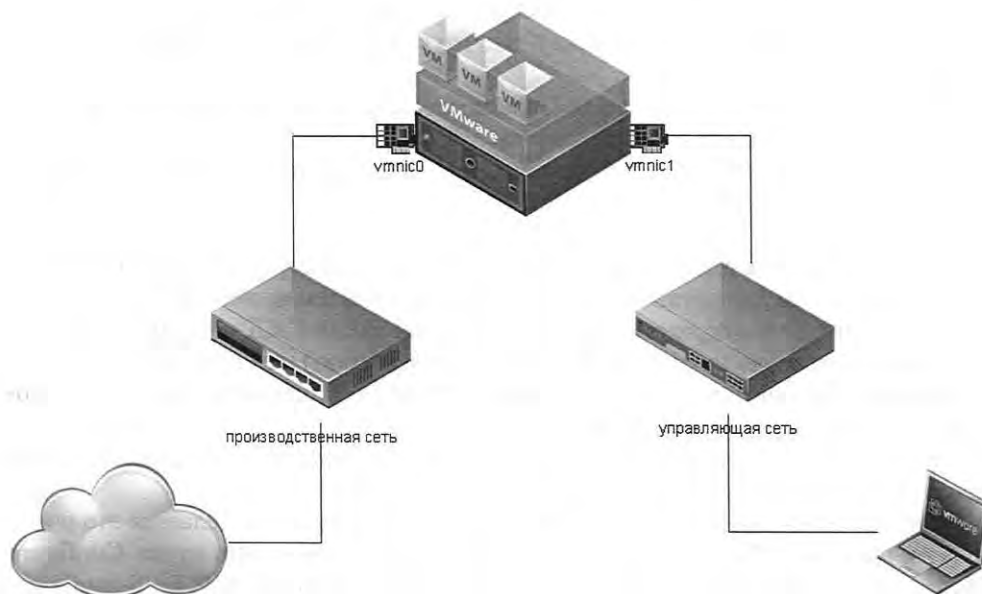


Рис. 1.3. Схема подключения физических сетевых контроллеров ESXi к разным сетям

VLAN – если вы не знаете, зачем нужен этот механизм, то проконсультируйтесь со специалистами, которые обслуживают вашу сеть. Если VLAN вам не нужны, то ничего не вводите в соответствующее поле. Немного теории про VLAN будет дано в посвященном сетям разделе.

Далее переходите к разделу **Начало работы**.

1.2.3. Автоматическая установка ESXi

Если вам предстоит установка большого количества серверов ESXi, то, возможно, вам будет интересна возможность автоматизировать этот процесс.

Для этого потребуется создать файл ответов и сослаться на него в начале установки.

Чтобы указать файл ответов, следует на первом шаге установки нажать **Shift+O** и указать путь к файлу ответов в появившейся командной строке.

В дистрибутиве ESXi есть файл ответов по умолчанию. Он установит ESXi на первый диск, пароль пользователя root будет «mypassword», все остальные настройки будут по умолчанию.

Чтобы воспользоваться файлом ответов по умолчанию, в командной строке, доступной при старте по **Shift+O**, введите

```
ks=file:///etc/vmware/weasel/ks.cfg
```

А как быть, если вариант по умолчанию нам не подходит?

Можно файл ответов сделать доступным по http или nfs и указать ссылку к этому файлу примерно в таком же виде.

А можно файл ответов добавить в дистрибутив.

Я приведу пример использования собственного файла ответа, а точнее – даже их набора. Кстати, листинги можно скопировать по ссылке <http://www.vm4.ru/2011/10/esxi-5-kickstart.html>.

Итак, чего хочется добиться: загрузочного флэш-накопителя с дистрибутивом ESXi, набором файлов ответов и меню для выбора этих файлов. Допустим, для ситуации, когда необходимо установить ESXi на 10 серверов, притом отличия установок только в имени сервера (ESXi01..ESXi10) и в IP-адресе.

Загрузите программу «UNetbootin, Universal Netboot Installer» и с ее помощью создайте загрузочную флэшку с дистрибутивом ESXi. Для этого вам потребуется запустить загруженную утилиту, выбрать iso-образ дистрибутива и флэшку с ранее созданным разделом файловой системы FAT.

После завершения копирования файлов (на вопрос о перезаписи файлов отвечайте утвердительно) вы становитесь обладателем загрузочного usb-накопителя, с которого можно производить установку ESXi.

Теперь добавим файлы ответов, например:

```
# Принять лицензионное соглашение
vmaccepteula
```

```
# Указать пароль пользователя root
rootpw VMware2012
```

```
# использовать первый диск, если там уже есть VMFS - переформатировать
install --firstdisk=local --overwritevmfs
```

```
# настройки сети
```



```
network --bootproto=static --device=vmnic0 --ip=1.1.1.1 --netmask=255.255.255.0
--gateway=1.1.1.253 --hostname=esxi-01.vm4ru.local --vlanid=0 --nameserver=1.1.1.252
```

```
# перезагрузить сервер после окончания установки, без выдвижения лотка cd-rom
Reboot --noeject
```

Жирным выделены поля, которые следует изменять под себя. Все варианты параметров файла ответов см. в документации – vSphere Installation and Setup ⇒ Installing, Upgrading, or Migrating Hosts using a Script (<http://link.vm4.ru/ksf>).

Создав несколько файлов ответов с необходимыми различиями, скопируйте их в отдельный каталог на ранее созданной флэшке (в моем примере название каталога kickstart).

Теперь отредактируем загрузочное меню. Для этого найдите в корне нашей флэшки файл syslinux.cfg и отредактируйте его следующим образом:

```
default menu.c32
prompt 0
menu title VMware VMvisor Boot Menu
timeout 300

label -
    menu label ^ESXi kickstart install:
    menu disable

label esx-01
    menu label ^Install esx-01
    menu indent 1
    kernel mboot.c32
    append vmkboot.gz ks=usb:/kickstart/esx01.cfg --- vmkernel.gz --- sys.vgz ---
    cim.vgz --- ienviron.vgz --- install.vgz

label esx-02
    menu label ^Install esx-02
    menu indent 1
    kernel mboot.c32
    append vmkboot.gz ks=usb:/kickstart/esx02.cfg --- vmkernel.gz --- sys.vgz ---
    cim.vgz --- ienviron.vgz --- install.vgz

label esx-03
    menu label ^Install esx-03
    menu indent 1
    kernel mboot.c32
    append vmkboot.gz ks=usb:/kickstart/esx03.cfg --- vmkernel.gz --- sys.vgz ---
    cim.vgz --- ienviron.vgz --- install.vgz

label ESXi Installer
    menu label ^ESXi Installer
    kernel mboot.c32
    append vmkboot.gz --- vmkernel.gz --- sys.vgz --- cim.vgz --- ienviron.vgz ---
    install.vgz
```

Здесь пути вида `:/kickstart/esx01.cfg` – это пути к файлам ответа (предыдущий листинг), помещенным в каталог `kickstart` в корне флэшки.

Теперь при загрузке сервера с этой флэшки вы увидите меню выбора – с каким файлом ответов произвести установку; и последний вариант – запустить стандартную установку с мастером.

Также обратите внимание на возможность PXE-загрузки установщика – <http://link.vm4.ru/uda>.

1.2.4. Особенности установки ESXi

Если ознакомиться с таблицей разделов на диске с установленным ESXi, то этих разделов мы обнаружим несколько:

1. Первым будет небольшой загрузочный раздел размером порядка 4 Мб.
2. Вторым и третьим идут одинаковые разделы размером в 250 Мб. В них хранятся основная и резервная копии ESXi.
3. Четвертым идет технический раздел для сохранения диагностической информации при падениях гипервизора в случае критических ошибок (PSOD, Purple Screen of Death, пурпурный экран смерти). Его размер порядка 110 Мб.
4. Раздел для хранения образов VMware tools размером порядка 286 Мб.
5. Если позволяют размеры диска (5 и более гигабайт), будет создан так называемый «scratch» раздел. Он используется для хранения файлов журналов и прочих временных данных.
6. На всем остальном пространстве будет создан раздел VMFS (исключая флэш-накопители, где это невозможно, даже если позволяет объем).

Что полезного можно отсюда извлечь?

На диске есть две копии ESXi, два раздела. Один из них активный (назовем его «первый»). Если мы устанавливаем обновление, то оно записывается в неактивный раздел (назовем его «второй») и помечает его как активный, чтобы дальнейшие загрузки происходили уже с него. Однако в бывшем активном разделе хранится старая версия ESXi. Это означает, что в случае неудачного обновления нам достаточно загрузиться с «первого» раздела – и вуаля, мы откатились на проверенную временем версию. Для этого следует нажать **Shift+R** при загрузке ESXi.

Обратите внимание. Загрузочные разделы используются практически только для первоначальной загрузки ESXi. Дело в том, что при старте он создает для себя RAM-диск, загружает туда необходимое и в дальнейшем на свой диск обращается только для периодической записи настроек.

Если не создался «scratch»-раздел, то все файлы журналов хранятся только в оперативной памяти ESXi, в его RAM-диске. Если не настроен syslog-сервер, то это сводит ценность файлов журналов как средства диагностики проблем к нулю. Однако scratch-раздел можно заменить на каталог с раздела VMFS – см. <http://kb.vmware.com/kb/1033696>. Это может быть полезно еще и из соображения эко-

номии места – иногда бывает неудобно расходовать по 4 Гб места на каждый сервер под временные данные.

1.2.5. Auto Deploy

VMware Auto Deploy – продукт, позволяющий организовать PXE-загрузку серверов. При такой организации бездисковой загрузки серверов мы получаем несколько преимуществ:

- ❑ сам факт того, что в серверах не нужны локальные носители под гипервизор, что слегка удешевляет эти сервера и упрощает управление и ввод в эксплуатацию;
- ❑ при необходимости обновить ESXi достаточно обновить используемый для PXE-загрузки образ, расположенный на сервере Auto Deploy, – и при следующем старте сервер загрузится уже с обновлениями;
- ❑ манипуляции для организации такого типа загрузки для сотни серверов практически не отличаются от манипуляций для десятка серверов – то есть в больших инфраструктурах Auto Deploy позволяет упростить развертывание и обновление серверов ESXi.

Загрузка ESXi при помощи Auto Deploy работает по следующей схеме:

1. Сервер включается, начинает загружаться с сетевого контроллера. Получает адрес IP по DHCP, там же получает адрес сервера TFTP и имя загружаемого файла.
2. По TFTP на сервер загружается маленький загрузчик gPXE. Теперь уже он обращается к серверу Auto Deploy по HTTP, сообщает информацию о сервере (такую как производитель, MAC-адреса сетевых контроллеров и др.). Основываясь на этих данных, Auto Deploy указывает образ ESXi, который следует загрузить на этот сервер. Auto Deploy основывается на правилах, которые администратор должен предварительно настроить. Образов может быть несколько разных, оптимизированных, например, под разные модели серверов или сервера разных производителей.
3. ESXi загружается, инициируется его добавление в vCenter. Однако загружаемый с сервера Auto Deploy образ ESXi не содержит настроек. Для настройки серверов используется механизм Host Profiles – то есть к добавленному в консоль vCenter серверу применяется профиль настроек, после чего сервер готов к эксплуатации. Этот механизм и сами настройки также должны быть настроены предварительно – один раз для каждого сервера.

Какие шаги необходимы для эксплуатации этого продукта:

1. Установка Windows-версии Auto Deploy и интеграции ее с Windows-версией vCenter, или развертывание vCenter Server Virtual Appliance, где служба Auto Deploy уже предустановлена. Также возможно использование VCSA с отдельно установленной службой Auto Deploy.
2. Настройка vCenter. Создание контейнера для серверов, назначение ключа лицензии.

3. Настройка DHCP и TFTP.
4. Установка PowerCLI – настройка Auto Deploy сегодня производится только командлетами.
5. Настройка Auto Deploy для первого сервера, настройка Host profiles.
6. Настройка Auto Deploy для последующих серверов.

Установка Windows-версии Auto Deploy

Установить Auto Deploy можно как на выделенный сервер (физический или виртуальный), так и на один сервер с vCenter. Однако для производственных сред VMware рекомендует использовать выделенный физический сервер.

Вам потребуется запустить autorun.exe в корне дистрибутива vCenter и в появившемся меню выбрать установку VMware Auto Deploy.

Сама настройка не представляет интереса и крайне проста.

После завершения установки на странице **Home** клиента vSphere должна появиться иконка **Auto Deploy**.

Больше вопросов вызывает планирование использования Auto Deploy. Данный сервис представляет из себя точку отказа – в случае его недоступности мы не сможем запустить сервера ESXi. К сожалению, на момент написания VMware не предоставляла каких-либо средств повышения доступности для Auto Deploy.

В остальном рекомендации по планированию виртуальной инфраструктуры с Auto Deploy следующие:

- ☐ устанавливать vCenter и Auto Deploy на один сервер;
- ☐ использовать виртуальный сервер под эти цели и размещать его в кластере HA для повышения доступности. Приоритет рестарта для этой VM следует выставить в «High»;
- ☐ выделить пару серверов, которые будут стартовать не при помощи Auto Deploy. При помощи правил DRS привязать VM с vCenter и Auto Deploy к этим серверам. Те же рекомендации следует применить к инфраструктурным серверам, здесь это в первую очередь DHCP и TFTP;
- ☐ для vCenter не будет лишней дополнительная защита, особенно если будут использоваться распределенные виртуальные коммутаторы. Если так, то VMware рекомендует развернуть службу отказоустойчивой кластеризации vCenter – vCenter Server Heartbeat (недоступна для Linux-версии vCenter на момент написания);
- ☐ Auto Deploy – это веб-сервер. В момент загрузки большого числа серверов ESXi этот веб-сервер будет испытывать значительную нагрузку. Для снижения нагрузки можно использовать стандартные решения такой задачи для веб-серверов. Плюс к тому, если вам требуется одномоментно загрузить большое число серверов, VMware рекомендует стартовать сервера группами – покластерно;
- ☐ с точки зрения безопасности сеть Auto Deploy следует максимально изолировать. Разумеется, следует изолировать и защищать доступными методами сам сервер Auto Deploy – его компрометация делает скомпрометированным каждый ESXi, загружаемый с этого сервера Auto Deploy.

Настройка vCenter

Загруженные при помощи Auto Deploy сервера попадут в какой-то датацентр (Datacenter, объект иерархии vCenter). Необходимо создать этот датацентр, если его еще нет.

Кроме того, уже внутри датацентра вы можете захотеть разместить сервера в каком-то кластере (Cluster) или папке (Folder). Если так, то создайте требуемые объекты заранее.

В своих примерах я буду использовать датацентр и кластер с именами «AutoDeployDC» и «AutoDeployCluster».

Для того чтобы загружаемые при помощи Auto Deploy сервера были лицензированы, необходимо выполнить специальную операцию – назначить лицензионный ключ на тот Datacenter, куда сервера будут помещаться. В документации это действие называется «Bulk licensing».

Тогда данный ключ автоматически будет присвоен этим серверам.

Вам потребуется следующий скрипт:

```
## Подключение к вашему серверу vCenter
Connect-VIServer vcenter.vm4ru.local

## здесь мы указываем, какому датацентру будем назначать ключ
$hostContainer = Get-Datacenter -Name AutoDeployDC

$licenseDataManager = Get-LicenseDataManager
$licenseData = New-Object VMware.VimAutomation.License.Types.LicenseData
$licenseKeyEntry = New-Object VMware.VimAutomation.License.Types.LicenseKeyEntry
$licenseKeyEntry.TypeId = "vmware-vmware"

## Указываем сам ключ
$licenseKeyEntry.LicenseKey = "24435-J809K-H8841-0U1K0-09W05"

$licenseData.LicenseKeys += $licenseKeyEntry

$licenseDataManager.UpdateAssociatedLicenseData($hostContainer.Uid, $licenseData)
$licenseDataManager.QueryAssociatedLicenseData($hostContainer.Uid)
```

Настройка TFTP и DHCP сервера

Нам потребуется настроить доступность некоторых файлов Auto Deploy по TFTP и некоторые параметры DHCP.

Для настройки TFTP следует в клиенте vSphere кликнуть по иконке **Auto Deploy** на странице **Home** и загрузить с открывшегося окна архив по ссылке «Download TFTP Boot Zip». Этот архив нужно распаковать в каталог, доступный по TFTP (этот каталог обычно указывается в настройках сервера TFTP).

Для DHCP следует указать две опции:

- ☐ 066 – IP-адрес сервера TFTP;
- ☐ 067 – имя загружаемого файла. Для серверов с BIOS это undionly.kpxe.vmw-hardwired. Несмотря на то что доступен отдельный загрузчик для сер-

веров с EFI, на момент написания такие сервера не поддерживаются для развертывания при помощи Auto Deploy, и это развертывание не работает.

Если необходимо сохранение IP-адресов управляющих интерфейсов серверов ESXi между перезагрузками – настройте DHCP reservation.

Настройка Auto Deploy для первого сервера

Вам потребуется дистрибутив ESXi, пригодный для Auto Deploy. Им может быть предоставляемый VMware вариант – ESXi Software Depot. Это архив zip, его можно загрузить с сайта VMware. На момент написания имя файла было следующим: VMware-ESXi-5.0.0-469512-depot.zip.

Или вы можете создать измененный дистрибутив при помощи Image Profile (см. посвященный этому инструменту раздел). Измененный дистрибутив может содержать дополнительные или обновленные драйверы, а также такие компоненты, как CIM-провайдеры или модули расширения ESXi (например, модуль виртуального коммутатора Cisco Nexus 1000V или модуль multipathing EMC PowerPath).

Дистрибутив (или несколько разных) следует поместить в какой-нибудь каталог, доступный с машины, откуда вы будете настраивать Auto Deploy. Например, это будет каталог d:\depot.

Обратите внимание. На момент написания был представлен графический интерфейс для Auto Deploy и Image Builder. Правда, в статусе экспериментального – <http://labs.vmware.com/flings/autodeploygui>.

На этой машине должны быть установлены PowerShell и PowerCLI.
Потребуется следующий код на PowerCLI:

```
## Подключение к вашему серверу vCenter
Connect-VIServer vcenter vcenter.vm4ru.local

## Регистрация дистрибутива ESXi
Add-EsxSoftwareDepot D:\depot\VMware-ESXi-5.0.0-469512-depot.zip

## Список вариантов его загрузки, они могут отличаться модулями
Get-EsxImageProfile

## загрузка дистрибутива на AutoDeploy. Name - на свое усмотрение.
## Item - указание базовых настроек этого дистрибутива.
## Во первых - Image Profile, см. предыдущую команду.
## То есть мы можем указывать, какие модули должны или не должны быть загружены.
## Далее - контейнер в vCenter, куда следует поместить свежезагруженный хост.
## Это объект типа Datacenter\Cluster\Folder.
## Pattern - признак сервера, на котором запускать этот дистрибутив с этим профилем
## Вместо паттерна для выборки серверов можно указать -AllHosts,
## тогда правило действует для всех серверов
New-DeployRule -Name "FirstBoot" -Item "With_LSI", AutoDeployCluster -Pattern
"model=VMware Virtual Platform"

## Регистрация правила, созданного ранее
Add-DeployRule -DeployRule FirstBoot
```

Обратите внимание. Если на той машине, где вы настраиваете Auto Deploy, есть доступ в Интернет, то вы можете получить список актуальных версий ESXi и загрузить требуемую прямо с сайта VMware. Вам потребуется команда

```
Add-EsxSoftwareDepot -DepotUrl https://hostupdate.vmware.com/software/VUM/PRODUCTION/main/vmw-depot-index.xml
```

Затем можно посмотреть доступную там версию ESXi командой

```
Get-EsxImageProfile
```

В приведенном примере самая главная команда – это New-DeployRule, и ее параметр –Item. В моем примере:

- ❑ «WithLSI» – это так называемый Image Profile, описание дистрибутива ESXi. При старте с одного и того же дистрибутива, но с разным профилем мы можем получить отличающиеся варианты ESXi. Отличаться они могут списком загруженных модулей, таких как драйверы, CIM-провайдеры, сторонние компоненты;
- ❑ «AutoDeployCluster» – это объект-«кластер» в иерархии vCenter. Этот кластер предварительно был создан администратором. Теперь все сервера, загружаемые по данному правилу, попадают в этот кластер.

Также важен параметр –Pattern. Auto Deploy определяет идентификаторы сервера и в зависимости от этих идентификаторов назначает то или иное правило на каждый загружаемый сервер. Именно данным параметром мы и указываем серверам, какому идентификатору данное правило соответствует. Если у вас будут разные правила для разных серверов – вам потребуется несколько правил.

Если у вас уже есть установленный сервер ESXi, то ознакомиться с вариантами идентификаторов легче всего на его примере. Вам поможет следующая команда (на примере сервера esxi01.vm4ru.local), и сразу я приведу список и примеры вывода, то есть идентификаторов:

```
get-vmhostattributes -vmhost "esxi01.vm4ru.local"
```

```
name value
----
vendor VMware, Inc.
uuid 423fecf3-eea1-a89b-5213-aaaa6f50bf60
model VMware Virtual Platform
gatewayv4 192.168.10.100
ipv4 192.168.10.51
hostname esxi01
domain vm4ru.local
ipv4 192.168.111.1
oemstring Welcome to the Virtual Machine
asset No Asset Tag
oemstring [MS_VM_CERT/SHA1/27d66596a61c48dd3dc721...
mac 00:50:56:bf:03:1b
mac 00:50:56:bf:03:1c
mac 00:50:56:bf:03:1d
mac 00:50:56:bf:03:1e
```

Если же вы планируете использовать единственное правило для всех ваших серверов, то вместо параметра `-Pattern` укажите параметр `-AllHosts`.

После создания правила Auto Deploy готов к первому старту вашего первого сервера.

После включения сервера и начала загрузки его по сети вы довольно быстро должны увидеть стандартный экран старта ESXi с индикатором загрузки и перечислением загружаемых модулей. После окончания загрузки сервер автоматически должен появиться в иерархии vCenter, в том датацентре, кластере или папке, что вы указали в правиле.

Теперь ваша задача – выполнить все необходимые настройки. Создание виртуальных коммутаторов, подключение iSCSI/NFS, изменение Advanced Settings и все прочее, что требуется в вашей инфраструктуре.

Есть несколько настроек, особенных для серверов, загружаемых по Auto Deploy.

Это настройки Syslog и Core Dump Collector.

Так как в случае Auto Deploy ESXi работает только в RAM-диске, по умолчанию там же хранятся все файлы журналов. Это означает, что существуют они только до первой перезагрузки. Это иногда не очень удобно и всегда не очень правильно.

Поэтому имеет смысл установить в сети сервер Syslog (можно в варианте от VMware, см. соответствующий раздел) и настроить сервера ESXi на пересылку файлов журналов на этот удаленный сервер. Всю необходимую информацию см. в разделе VMware Syslog Collector.

Примерно та же ситуация с информацией, доступной при отказе гипервизора (PSOD, Purple Screen of Death). Для того чтобы в случае такого отказа сервера нам была доступна для анализа отладочная информация, следует установить и настроить соответствующий продукт от VMware. Вся необходимая информация доступна в разделе VMware Dump Collector.

Итак, на данный момент у вас есть первый сервер, первый раз загруженный при помощи Auto Deploy. Вы произвели для него необходимые настройки, однако эти настройки будут потеряны при перезагрузке.

Чтобы этого не произошло, создадим профиль настроек (Host Profile), взяв за основу этот сервер. В контекстном меню сервера выберите **Host Profiles ⇒ Create Profile from this host**.

Теперь перейдите в интерфейс управления профилями настроек: **Home ⇒ Host Profiles**. Найдите там созданный только что профиль, на вкладке **Hosts and Clusters** назначьте (attach) на тот же сервер. После назначения профиля необходимо проверить соответствие настроек сервера профилю (check compliance) – выполните эту процедуру, если она не произошла автоматически.

На текущий момент у вас должен быть профиль настроек, снятый с первого сервера. Этот профиль назначен на этот же сервер, проверено соответствие настроек сервера профилю – и сервер профилю удовлетворяет.

Кстати говоря, возможно, вы захотите изменить профиль настроек вручную – например, указав пароль пользователя root.

Последний шаг здесь – сделать так, чтобы этот профиль применялся к нашему серверу после каждой перезагрузки. Однако есть проблема – профиль содержит универсальные настройки (например, что необходимо создать интерфейс VM-kernel для vMotion). А вот уникальных для сервера настроек (таких как IP-адрес этого интерфейса) профиль не содержит. Поэтому наше следующее действие – создать файл ответов (answer file).

В контекстном меню сервера (по-прежнему находясь в интерфейсе профиля настроек) выберите пункт **Update Answer File**. У вас запросят всю уникальную информацию и сохранят ее в базе данных vCenter.

После завершения данной последовательности действий вы почти закончили настройку Auto Deploy. Вернемся к правилу – нам необходимо пересоздать правило Auto Deploy, чтобы с его помощью еще и профиль настроек назначать на сервер.

```
Connect-VIServer vcenter
```

```
## Регистрация дистрибутива ESXi
```

```
Add-EsxSoftwareDepot D:\depot\VMware-ESXi-5.0.0-469512-depot.zip
```

```
## Список вариантов его загрузки, они могут отличаться модулями
```

```
Get-EsxImageProfile
```

```
## Удалим ранее созданное правило
```

```
Remove-DeployRule "FirstBoot" -delete
```

```
## Создадим заново, указав еще и название профиля.
```

```
New-DeployRule -Name "NormalBoot" -Item "With_LSI", AutoDeployCluster, HostProfileAuto  
Deploy -AllHosts
```

```
## Активация правила, созданного ранее
```

```
Add-DeployRule -DeployRule NormalBoot
```

Здесь «HostProfileAutoDeploy» – имя профиля настроек, которые мы создали чуть ранее.

Теперь ваш первый сервер должен быть еще и автоматически настроен после перезагрузки – ведь при старте он будет добавлен в vCenter, к нему будет применен профиль настроек с уже созданным нами файлом ответов.

Настройка Auto Deploy для последующих серверов

Теперь включим второй сервер. На нем будет загружен ESXi, он будет добавлен в нужный контейнер в vCenter, к нему будет привязан профиль настроек – в соответствии с правилом чуть выше.

Однако автоматически применить настройки из профиля не получится – файла ответов-то нет.

Поэтому для каждого впервые запущенного сервера вам необходимо один раз создать файл ответов (**Home** ⇒ **Host Profiles** ⇒ профиль для Auto Deploy ⇒ вкладка **Hosts and Clusters** ⇒ контекстное меню нового сервера ⇒ **Update Answer File**).

Все.

Обновление образа, загружаемого при помощи Auto Deploy

В какой-то момент времени перед вами встанет задача заменить загрузочный образ ESXi.

Вы загрузите новый образ с сайта VMware или создадите свой собственный при помощи Image Builder. Останется настроить Auto Deploy на использование этого нового образа.

Например, на данный момент я обладаю следующим:

- ☐ дистрибутивом ESXi. Имя файла VMware-ESXi-5.0.0-469512-depot.zip. Каталог D:\depot;
- ☐ обновлением для ESXi. Имя файла ESXi500-201109001.zip. Каталог D:\depot;
- ☐ правилом для Auto Deploy с именем «NormalBoot». Правило основано на исходном дистрибутиве ESXi.

Вам потребуется примерно следующий скрипт:

```
## добавляем дистрибутивы
## исходный дистрибутив esxi
Add-EsxSoftwareDepot D:\depot\VMware-ESXi-5.0.0-469512-depot.zip
## дистрибутив обновления
Add-EsxSoftwareDepot D:\depot\ESXi500-201109001.zip
## дистрибутив агента HA
Add-EsxSoftwareDepot http://<адрес сервера vCenter>:80/vSphere-HA-depot

## в правило с именем NormalBoot добавляем обновление.
## список правил, из которого узнаем имя правила - Get-DeployRule
## ESXi-5.0.0-20110904001-standard - здесь это имя Image Profile из
## дистрибутива обновления
Copy-DeployRule -DeployRule NormalBoot -ReplaceItem "ESXi-5.0.0-20110904001-standard"

## добавляем агент HA
Add-EsxSoftwarePackage -ImageProfile "ESXi-5.0.0-20110904001-standard" -SoftwarePackage "vmware-fdm"

## Еще раз обновляем правило
Copy-DeployRule -DeployRule NormalBoot -ReplaceItem "ESXi-5.0.0-20110904001-standard"

## проверяем сервера на соответствие правилу - из-за того, что в правиле указан
## дистрибутив с обновлением, соответствия не будет
## здесь "AutoDeployCluster" - кластер, куда помещаются загружаемые через Auto Deploy сервера
Get-VMHost -Location AutoDeployCluster | Test-DeployRuleSetCompliance
## применяем обновления к серверам
Get-VMHost -Location AutoDeployCluster | Test-DeployRuleSetCompliance | Repair-DeployRuleSetCompliance
```

После этой процедуры при следующем старте сервера должны загружаться в новую версию ESXi.

1.3. Вспомогательные компоненты vSphere

В этом разделе я упомяну о нескольких дополнительных компонентах vSphere, которые важны уже на этапе установки.

1.3.1. Image Builder

В пятой версии vSphere появилась штатная возможность изменять дистрибутив ESXi путем внесения в него дополнительных драйверов и компонентов – с целью получить дистрибутив, наиболее подходящий под конкретный набор оборудования и инфраструктуру. Для этих целей предназначен компонент Image Builder.

При помощи Image Builder возможно создать:

- ☐ образ iso с дистрибутивом ESXi для ручной установки его на сервера;
- ☐ архив zip для загрузки ESXi по PXE, при помощи Auto Deploy;
- ☐ архив zip, который может быть использован для обновления серверов ESXi (в том числе с версии 4.x до версии 5), при помощи VMware Update Manager или локальной командной строки.

При работе с Image Builder следует выделить три компонента:

- ☐ VIB, или vSphere Installation Bundle, – это пакет с ПО. VIB-файл может содержать драйверы, CIM-провайдеры, приложения для ESXi. Поставляться VIB-пакеты могут как самой VMware, так и компаниями-партнерами. В контексте Image Builder – мы добавляем откуда-то полученные VIB-пакеты в стандартный дистрибутив ESXi, получая таким образом необходимые нам возможности;
- ☐ Image Profile – набор VIB-пакетов. При помощи Image Builder мы создаем «описание образа» (Image Profile), затем отдельной командой «собираем» образ из указанных в «описании» пакетов;
- ☐ Software Depot – тип VIB-пакетов, подходящих для интеграции в дистрибутив при помощи Image Builder. То есть теоретически мы можем загрузить некий драйвер в виде VIB или в виде Software Depot. Первое будет подходить для установки этого драйвера на уже существующие сервера ESXi, второе – для интеграции драйвера в дистрибутив.

VIB поставляется в виде одного файла, но это архив, содержащий в себе непосредственно данные (например, драйвер). Кроме того, в этом архиве находится xml-описание. Оно используется для обработки добавления VIB к образу ESXi или удалению из образа. Кроме того, существует цифровая подпись этого VIB-пакета. Она указывает на легитимность данного VIB, на авторство и на уровень поддерживаемости.

Возможные варианты:

- ☐ VMwareCertified – VIB был создан и протестирован VMware. VMware оказывает поддержку в случае проблем с этими модулями. На момент написания таким уровнем проверки обладают только драйверы.

- ❑ **VMwareAccepted** – VIB был создан доверенным партнером, VMware проверяла качество. Запросы на поддержку будут перенаправлены партнеру.
- ❑ **PartnerSupported** – VIB был создан и проверен партнером VMware, VMware не проверяла качество. Запросы на поддержку будут перенаправлены партнеру.
- ❑ **CommunitySupported** – VIB был создан частным лицом или компанией вне партнерской программы VMware. VMware не несет ответственности за качество работы и не оказывает поддержки.

Все пакеты, проверенные VMware или партнером, имеют цифровую подпись – это облегчает использование измененных дистрибутивов ESXi в инфраструктурах с повышенными требованиями к безопасности.

Во время создания «описания образа» (Image Profile) мы можем явно указать желаемый уровень доверенности (выбирая из тех же самых четырех вариантов выше). VIB с менее доверенным статусом не могут быть включены в Image Profile.

Отдельные VIB могут быть загружены администратором и установлены на уже имеющиеся ESXi. Однако в текущей версии эту операцию можно произвести только из командной строки. VMware Update Manager не работает с индивидуальными VIB-пакетами. Также с отдельными VIB-пакетами не работает Image Builder.

При желании использовать Image Builder или Update manager для внедрения VIB в дистрибутив или установки их на уже развернутые ESXi необходимо использовать так называемый Software Depot. Это архивы с одним или несколькими VIB-пакетами и дополнительными метаданными.

Для того чтобы воспользоваться Image Builder, нам потребуются следующие шаги:

1. Загрузить необходимые VIB-пакеты и дистрибутив ESXi в специальном формате, так называемом «ESXi offline bundle», обычный образ iso не подойдет.
2. Создать «Image profile».
3. Добавить к нему VIB.
4. Применить ранее созданный Image Profile к дистрибутиву.

Обнаружить подходящий для интеграции пакет VIB можно на сайте VMware, в разделе Download. Ссылки на загрузку разделены по нескольким вкладкам, одна из них – **Drivers&Tools** и раздел на ней **Driver CDs**.

Обратите внимание. Сторонние компании также могут предоставлять веб-ресурсы с разнообразными vib для ESXi. Например, HP – <http://vibsdepot.hp.com>.

Итак, вами загружен один или несколько пакетов с ПО, которые вы хотите добавить в дистрибутив, и сам дистрибутив в виде «ESXi offline bundle». Создайте каталог на диске, скопируйте в него дистрибутив ESXi и сюда же (или в подкаталог для удобства) распакуйте пакеты VIB.

Запустите PowerCLI (если вы незнакомы с PowerShell/PowerCLI, см. соответствующий раздел. Вкратце – для этих манипуляций вам необходимо просто установить PowerCLI на тот компьютер, где вы будете заниматься этой интеграцией).

Обратите внимание. На момент написания был представлен графический интерфейс для Auto Deploy и Image Builder. Правда, в статусе экспериментального – <http://labs.vmware.com/flings/autodeploygui>.

Далее вам потребуется выполнить несколько команд для осуществления действий по списку чуть выше:

```
# Подключение к vCenter
Connect-VIServer "имя или IP сервера vCenter"

# Регистрация модулей VIB
Add-EsxSoftwareDepot D:\depot\LSI_5_34-offline_bundle-455140.zip

# Просмотр названия и прочих данных зарегистрированного ПО
# Эти названия пригодятся немного позже
Get-EsxSoftwarePackage

# Регистрация самого дистрибутива ESXi
add-esxsoftwaredepot D:\depot\VMware-ESXi-5.0.0-469512-depot.zip

# Просмотр списка профилей. Пока только существующие по умолчанию
get-esximageprofile

# Создание своего профиля. Имя – на свой выбор.
# Создаем не с нуля, а клонируя профиль по умолчанию
new-esximageprofile -cloneprofile ESXi-5.0.0-469512-standard -name "With_LSI"

# Опять просмотр списка – чтобы убедиться, что наш профиль появился
get-esximageprofile

# Добавление VIB в созданный профиль
PowerCLI C:\> add-esxsoftwarepackage -imageprofile «With_LSI» -softwarepackage scsi-
megaraid-sas

# Экспорт дистрибутива с добавленными VIB в отдельный образ ISO, пригодный к установке
с него ESXi
export-esximageprofile -imageprofile "With_LSI" -filepath d:\depot\esxi5.0.0-with_LSI-
469512.iso -exporttoiso -force
```

Полученный образ можно использовать для установки ESXi и для импорта в VMware Update Manager с целью обновления серверов ESXi.

Кроме того, если в последней команде поменять параметр –ExportToIso на ExportToBundle, то на выходе получится архив zip, пригодный для использования с VMware Auto Deploy.

Если возникнет задача сохранять профили между сессиями PowerShell, то сохранить и загрузить их можно следующими командами:

```
Export-EsxImageProfile -ImageProfile "my_profile" -ExportToBundle -FilePath "C:\isos\temp-base-plus-vib25.zip"
Add-EsxSoftwareDepot "C:\isos\temp-base-plus-vib25.zip"
```

Если вы готовите образ дистрибутива для Auto Deploy и загружаемые с этого дистрибутива сервера будут входить в кластер VMware HA, то образ по умолчанию имеет смысл расширить агентом VMware HA, или, более правильное название, компонентом FDM (Fault Domain Manager, новое имя VMware HA в пятой версии vSphere).

```
## Регистрация источника модулей
Add-EsxSoftwareDepot http://<адрес сервера vCenter>:80/vSphere-HA-depot

## Создание нового профиля образа, то есть набора загружаемых компонентов
New-EsxImageProfile -CloneProfile ESXi-5.0.0-469512-standard -name "ESXi_with_HA"

## добавляем в этот профиль компонент «Агент HA»
## теперь при создании правила AutoDeploy можем указывать Image profile с именем ESXi_with_HA
Add-EsxSoftwarePackage -ImageProfile "ESXi_with_HA" -SoftwarePackage vmware-fdm
```

Без такого изменения дистрибутива установка агента HA будет происходить каждый раз, когда сервер подключается к vCenter после перезагрузки.

1.3.2. VMware Syslog Collector

ESXi сохраняет свои файлы журналов на свой диск, в каталог «/var/log» по умолчанию. Но такое положение вещей не всегда удобно, особенно в тех случаях, когда этот каталог сохраняется в RAM-диске и очищается при перезагрузке сервера.

Так как для записей в файлы журналов ESXi использует сервис syslog – есть возможность развернуть и настроить централизованный сервер сбора файлов журналов.

Таким сервером может быть любая реализация службы syslog, в частности реализация VMware под названием VMware Syslog Collector.

В составе дистрибутива vCenter поставляется Windows-вариант сервера syslog. (А в составе vCenter Virtual Appliance эта служба предустановлена.)

Запуск мастера установки доступен из меню **Autorun** дистрибутива vCenter. Эта служба может быть установлена на любой Windows-сервер, в том числе на сам vCenter Server. Мне такой вариант кажется достаточно удобным, противоположений к нему я не знаю.

Установка Windows-версии сервера VMware Syslog Collector тривиальна. Вопросы, которые задаст нам мастер:

- ☐ каталог для хранения файлов журналов;
- ☐ максимальный объем одного файла и количество файлов журналов для каждого сервера;
- ☐ надо ли интегрировать этот syslog-сервер с vCenter.

К каталогу для хранения файлов журналов только один критерий – чтобы вам было удобно его найти. Я предпочитаю путь вида «C:\Syslog».

Все сообщения с каждого сервера попадают в отдельную папку, но в один-единственный файл. Когда размер этого файла достигает указанного значения (2 Мб по умолчанию), этот файл перестает использоваться, но сохраняется. Все дальнейшие записи идут в следующий файл. Когда количество этих файлов достигает указанного нами максимума (по умолчанию – 8), самый старый файл удаляется.

Таким образом, от количества и размера файлов зависит количество времени, за которое нам доступны записи в файлах журналов. (В моем тестовом окружении один сервер с десятком виртуальных машин заполнял двухмегабайтный файл примерно за три часа).

Интеграция Syslog Collector с vCenter, к сожалению, далека от идеала. Появляющаяся на странице **Home** пиктограмма позволяет вспомнить параметры сервера Syslog и увидеть, какие сервера ESXi используют этот Syslog-сервер (рис. 1.4). Но просмотр файлов журналов из этого интерфейса невозможен.

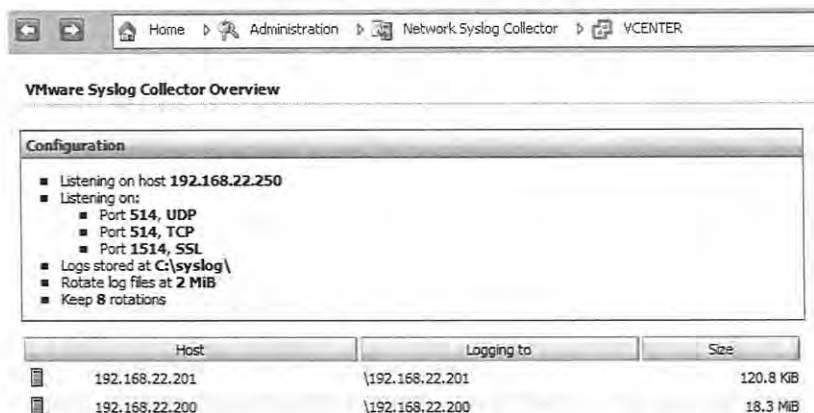


Рис. 1.4. Данные о Syslog-сервере в клиенте vSphere

Для того чтобы сервер ESXi начал отправлять свои журналы на удаленный сервер syslog, вам потребуется пройти на вкладку **Configuration** ⇒ **Advanced Settings** ⇒ **Syslog** и прописать в поле **Syslog.global.loghost** адрес сервера syslog примерно таким образом: `udp://192.168.22.250:514`.

Кроме UDP, поддерживаются протоколы TCP и SSL.

Или из командной строки PowerCLI:

```
Get-VMHost | Set-VMHostAdvancedConfiguration -Name Syslog.global.logHost -Value
udp://192.168.10.50:514
```

Такая команда выполнит данную настройку сразу для всех серверов ESXi в vCenter, к которому открыта сессия PowerCLI.

Кроме того, настройка удаленного сервера syslog может быть произведена при помощи Host Profiles.

В качестве альтернативы серверу syslog есть возможность указать каталог для записи туда файлов журнала на любом VMFS- или NFS-хранилище. Для указания каталога и хранилища вам нужна вкладка **Configuration** ⇒ **Advanced Settings** ⇒ **Syslog**. В поле **Syslog.global.LogDir** укажите требуемый путь в формате [имя хранилища] /logs.

Обратите внимание на квадратные скобки вокруг имени хранилища и пробел перед путем к каталогу. Обычно бывает удобно указать один и тот же путь для всех серверов ESXi и на всех поставить флажок **Syslog.global.logDirUnique** – в этом случае каждый сервер создаст по указанному пути подкаталог со своим IP-адресом в качестве имени и свои файлы журналов разместит уже в личном подкаталоге.

1.3.3. VMware Core Dump Collector

Если произойдет критический сбой ESXi – PSOD, Purple Screen of Death, пурпурный экран смерти, то гипервизор сохранит диагностическую информацию в специально предназначенном для этого разделе диска (так называемый «Dump partition»), куда гипервизор установлен. Этот дамп призван помочь обнаружить причину сбоя.

Однако в случае загрузки серверов ESXi по сети при помощи AutoDeploy этого раздела для диагностической информации нет, так как гипервизор в принципе не использует для себя диски. В таком случае для доступа к диагностической информации следует развернуть службу VMware Core Dump Collector, которая сможет принять дамп от ESXi по сети.

Это приложение поставляется вместе с vCenter, запуск его установки доступен из меню **Autorun** дистрибутива vCenter, и Dump Collector может быть установлен на самом vCenter.

А если вы используете vCenter Virtual Appliance, то в таком случае Dump Collector даже предустановлен в этой виртуальной машине.

Установка его крайне проста, интерес представляют только два вопроса: в каком каталоге сохранять полученную от ESXi информацию и интегрироваться ли с vCenter.

Каталог я предпочитаю выбирать более быстро доступный, чем в варианте по умолчанию, что-нибудь вроде «C:\Dumps». (Для vCenter Virtual Appliance путь по умолчанию «/var/core/netdumps/».)

Интеграция с vCenter, к сожалению, фиктивная. Если мы укажем данные для интеграции при установке Dump Collector, то на странице **Home** клиента vSphere появится соответствующая пиктограмма, но, выбрав ее, мы лишь увидим информацию о параметрах Dump Collector (адрес сервера, где тот работает, и используемый сетевой порт). Так что (на момент написания) эта интеграция выполняет роль памятки для администратора – где установлена эта утилита, но не более того.

Еще мастер установки позволяет изменить порт по умолчанию – указанный UDP-порт должен быть открыт в брандмауэрах, если таковые будут между серверами ESXi и сервером Dump Collector.

По умолчанию для хранения диагностической информации предполагается использовать до 2 Гб. По достижении этого объема дампы начнут удаляться, начиная с самых старых. Насколько я могу судить, размер дампов крайне мал, и предлагаемые по умолчанию 2 Гб должны быть достаточны в большинстве случаев.

Изменить путь для хранения информации, максимальный размер и сетевой порт для уже установленного Dump Collector можно в файле `vmconfig-netdump.xml`, расположенном «%ALLUSERSPROFILE%\VMware\VMware ESXi Dump Collector». После редактирования файла требуется рестарт службы Dump Collector.

На vCenter Virtual Appliance эти настройки хранятся в файле `/etc/sysconfig/netdumper`.

После завершения установки Dump Collector следует указать серверам ESXi-адрес и порт машины, где он установлен. Это можно выполнить при помощи Host Profiles или из командной строки.

В варианте Host Profiles следует изменить уже существующий профиль настроек: **Network Configuration** ⇒ **Network CoreDump Settings**.

Единственная альтернатива – это воспользоваться командной строкой, но удобно, что подойдут и локальная командная строка, и удаленная командная строка, и PowerCLI.

Для локальной командной строки и для удаленной командной строки (vSphere CLI):

```
esxcli system coredump network set --interface-name vmk0 --server-ipv4=1-XX.XXX -
    ыукмук-порт=6500
esxcli system coredump network set --enable=true
esxcli system coredump network get
```

Для PowerCLI:

```
## выбрать сервер для настройки
$esx= Get-VMHost "host00.micro.local"

$esxcli = Get-EsxCli -VMHost $esx
## осуществить настройку Dump Collector
## здесь vmk0 – тот интерфейс гипервизора, через который будет производиться
## обращение на Dump Collector
## IP-адрес и порт – адрес сервера Dump Collector и порт
$esxcli.system.coredump.network.set($null,"vmk0","192.168.22.250",6500)
## активация использования сети для отправки диагностической информации
$esxcli.system.coredump.network.set(1)
## просмотр сделанных настроек
$esxcli.system.coredump.network.get()
```

В случае использования Dump Collector для серверов, разворачиваемых через AutoDeploy, эту настройку следует выполнить только для первого сервера – остальные сервера возьмут эту настройку из Host Profile.

Однако с сохранением диагностической информации по сети есть несколько ограничений:

- ❑ управляющий интерфейс vmkernel, который выбран для обращения на Dump Collector, не должен использовать Etherchannel/LACP;
- ❑ протокол netdump, используемый данным решением, поддерживает только IPv4;
- ❑ не поддерживаются механизмы авторизации и шифрования. Таким образом, единственным способом повысить безопасность данного механизма является изоляция сети, используемой для пересылки диагностической информации. К сожалению, изоляция возможна лишь средствами физической сети – см. следующий пункт;
- ❑ при обращении на Dump Collector игнорируются настройки VLAN на группах портов виртуальных коммутаторов;
- ❑ выбранный для отсылки дампов интерфейс vmkernel не может располагаться на распределенном виртуальном коммутаторе (в том числе на стороннем распределенном виртуальном коммутаторе, таком как Cisco Nexus 1000V).

Как альтернативу сетевому сборщику диагностической информации можно рассмотреть создание общего LUN на системе хранения и настройку сброса дампа в раздел на этом общем LUN для каждого сервера.

Если настроен и сетевой сбор дампов, и сброс дампа на раздел диска (локального или с системы хранения), то использоваться будут оба механизма.

Для проверки корректности настроек можно использовать два способа.

Выполнив в локальной командной строке ESXi команду

```
vsish -e set /reliability/crashMe/Panic
```

вы вызовете тот самый пурпурный экран смерти. Прямо на нем вы должны увидеть отчет об отправке дампа по сети и обнаружить новый файл дампа в соответствующей серверу папке на машине Dump Collector. Очевидно, что использовать этот способ следует на тестовом сервере ESXi, без работающих на нем VM.

Второй способ менее нагляден, но более безопасен. В локальной командной строке ESXi выполните команду

```
echo testmessage | nc -w 1 -s <IP-адрес интерфейса vmkernel> -u <IP-адрес Dump  
Collector> 6500
```

После этого, если все работает так, как ожидается, в файле журнала Dump Collector («%ALLUSERSPROFILE%\VMware\VMware ESXi Dump Collector\logs\netdumper.txt») вы должны обнаружить строки вида:

```
2011-11-19T15:29:03.068+04:00| vthread-4| Bad magic:0xa656761. Expected:0xadeca1bf  
2011-11-19T15:29:03.068+04:00| vthread-4| Skipping bad packet.
```

Зачем эти дампы нам пригодятся?

Самое главное – их может запросить поддержка VMware в случае инцидента.

А также их можно попробовать проанализировать самостоятельно. Единственный известный мне способ их анализа – это распаковка файла дампа командой `vmkdump_extract`, доступной в локальной командной строке ESXi 5. Вам может пригодиться статья базы знаний VMware № 1004250 (<http://kb.vmware.com/kb/1004250>).

1.4. Начало работы

В пятой версии vSphere у нас не очень много вариантов, как начать работать. Вся разница – у нас есть сервер vCenter или нет (еще нет).

В любом случае нам потребуется клиент vSphere (vSphere Client) – это Windows-приложение, предоставляющее графический интерфейс для управления как отдельным ESXi, так и vCenter.

1.4.1. Начало работы без vCenter

Предполагается, на этом этапе у нас есть как минимум один установленный сервер ESXi. vCenter у нас, может быть, нет – и все дальнейшие манипуляции будут осуществляться с каждым ESXi отдельно.

Или vCenter пока нет – мы хотим развернуть на имеющемся ESXi виртуальную машину, туда установить vCenter и в дальнейшем работать уже при его помощи.

В любом случае, нам потребуется установить на свое рабочее место приложение vSphere Client, клиент vSphere.

С локальной консоли сервера мы не сможем управлять ESXi. На локальном мониторе отображается лишь немного информации, в частности IP-адрес сервера ESXi, и практически ничего больше. Для работы нам понадобится отдельная машина с Windows, куда необходимо установить клиента vSphere. Поддерживаются следующие версии Windows: XP, Vista, 7, Server 2003, Server 2008 в 32- и 64-битных версиях.

Этой внешней машиной, допустим, будет ваш компьютер. Далее я его буду называть клиентским компьютером.

На него устанавливаем vSphere Client (клиент vSphere). Взять дистрибутив можно с веб-интерфейса ESXi. Браузером обращаемся на IP-адрес сервера, на страничке щелкаем по ссылке **Download vSphere Client**. Загружаем, устанавливаем. Обратите внимание на то, что начиная с версии 4.1 (и в версии 5 в том числе) ссылка на загрузку клиента ведет на сайт VMware, а в составе ESXi дистрибутива клиента больше не поставляется. Это значит, что такой вариант вам может быть неудобен, если с вашего клиентского компьютера нет достаточно быстрого доступа в Интернет для загрузки дистрибутива клиента vSphere (размером порядка 250 Мб).

Этот клиент может быть найден на сайте VMware не только по прямой ссылке с описываемой странички, но и в доступном после регистрации разделе down-

load – таким образом вы можете загрузить этот дистрибутив отдельно, при необходимости с другой машины.

Если вы будете использовать vCenter Server, то дистрибутив клиента vSphere проще всего найти в дистрибутиве или на веб-странице сервера vCenter. При обращении на веб-интерфейс vCenter загрузка клиента vSphere происходит с сервера vCenter, не из Интернета.

Установка клиента тривиальна, мастер не задаст ни одного вопроса.

После завершения установки запускаем установленный клиент vSphere, указываем имя или IP-адрес сервера ESXi и подключаемся. Авторизоваться вы должны пользователем root с тем паролем, что вы указали при установке сервера ESXi.

В первый раз мы увидим предупреждающее сообщение о неподписанном сертификате. Если вы не понимаете, что это такое, проконсультируйтесь со специалистом по безопасности вашей компании. Нужен этот сертификат для подтверждения легитимности сервера. Если вопросы выполнения рекомендаций безопасности стоят не настолько остро, что вы будете заниматься генерацией и распространением доверенных сертификатов ssl по серверам ESXi, то достаточно поставить флажок **Ignore**.

Подключившись первый раз, следует выполнить минимальную настройку. После подключения вы попадете на страничку **Home**. На ней выберите иконку **Inventory** и выделите свой сервер. Затем пройдите на вкладку **Configuration** для сервера (рис. 1.5):

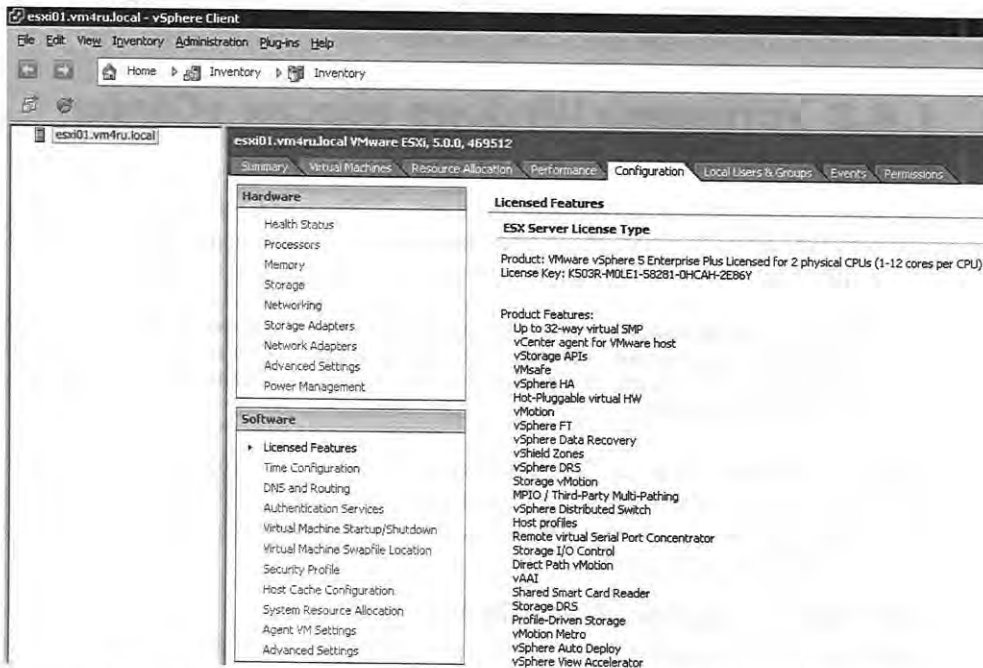


Рис. 1.5. Подключение клиентом vSphere напрямую к серверу ESXi

Из начальных настроек представляют наибольший интерес:

1. **Licensing Features** – здесь необходимо указать ключ продукта. Напомню, что без него на сервере будут доступны все функции в течение 60 дней ознакомительного периода.
2. **Time Configuration** – настройки NTP. Настроек минимум – включить клиент NTP и указать адрес сервера NTP.
3. **DNS and Routing** – настройки DNS и шлюзов по умолчанию.
4. **Networking** – сеть. Подробности по настройке сети см. в соответствующем большом разделе.
5. **Storage** – система хранения. Подробности по настройке дисковой подсистемы см. в соответствующем большом разделе.

После этого можно создавать и включать виртуальные машины.

Если мы планируем использовать vCenter Server, то так никогда не делаем ☺.

Под «так» имеется в виду «подключаясь напрямую к серверу». Если есть vCenter, стараемся все действия производить через него. И только если совсем не получается, лишь в этом случае подключаемся к ESXi напрямую. «Не получается» означает какого-то рода проблемы.

Установке и работе с vCenter посвящен один из следующих разделов. Обзор элементов интерфейса клиента vSphere при работе напрямую с ESXi и через vCenter – чуть далее.

Есть маленький список действий, которые доступны вам только при работе с сервером ESXi напрямую, а не через vCenter Server. В частности, это создание и управление локальными пользователями ESXi. Большинству из вас дополнительные локальные пользователи на ESXi не нужны.

1.4.2. Установка Windows-версии vCenter Server

Здесь поговорим про вторую часть vSphere – VMware vCenter Server, про интерфейс работы с ESXi через vCenter, а также про шаги первоначальной настройки.

Также последовательно поговорим про все шаги установки этого продукта.

Обратите внимание: в устанавливаемой с нуля версии vSphere 5 у вас есть выбор между двумя вариантами: vCenter – это «классический» vCenter в виде приложения под Windows и виртуальная машина с предустановленной Linux-версией vCenter – vCenter Server Appliance, vCSA.

Я в книге сделал упор на Windows-версию, но подавляющее большинство функций и настроек никак не зависят от выбранного варианта vCenter. Если перед вами стоит выбор, ознакомьтесь с разделом 1.3.3 – vCenter Virtual Appliance, там размещена информация по сравнению.

Системные требования vCenter

Системные требования текущей версии vCenter Server 5 имеет смысл смотреть в документации (здесь и далее документацию ищите по адресу <http://pubs.vmware.com>).

Информацию именно о совместимости между компонентами vSphere разных версий следует искать в документе VMware vSphere Compatibility Matrixes). Я же скажу, на что имеет смысл обратить внимание.

У сервера vCenter должно быть достаточно ресурсов: при десятках хостов и сотнях ВМ одного процессора может быть мало, от 2–4 Гб ОЗУ, несколько гигабайт места на диске.

Обратите внимание: это только для службы vCenter Server. Возможно, на той же машине вы установите некоторые дополнительные продукты. Например, сервер базы данных для vCenter.

vCenter – это приложение под Windows. В качестве ОС vCenter может использовать Windows Server 2003 SP2 – 2008 R2 (начиная с версии 4.1 – только 64-битные) и 64-битную Windows XP Pro. Впрочем, за точным списком для текущей версии имеет смысл обращаться все-таки в документацию. Windows XP я не рекомендую использовать даже для тестовых установок.

Также для работы vCenter необходима база данных. Если она будет работать на том же сервере или ВМ, что и vCenter, то для нее потребуются дополнительные ресурсы, включая место на диске.

vCenter Server требует физического сервера или ВМ, в которую мы будем его устанавливать. ВМ может работать на одном из ESXi серверов, который будет управляться этим vCenter Server.

Есть некоторые соображения по использованию для установки vCenter Server физической или виртуальной машины. Очевидно, что в первом случае потребуются сервер – это финансовые затраты. Какие соображения есть, кроме этого?

Плюсы использования для vCenter виртуальной машины на одном из ESXi:

- ☐ нет необходимости в выделенном сервере под vCenter;
- ☐ ВМ с vCenter может быть защищена механизмами HA (High Availability), FT (Fault Tolerance);
- ☐ ВМ с vCenter можно мигрировать на другой сервер в случае необходимости обслуживания сервера ESXi, где эта ВМ работает;
- ☐ к ВМ с vCenter применимы снимки состояния (snapshot), дающие очень простую возможность обеспечить точку возврата перед потенциально опасными действиями.

Минусы использования для vCenter виртуальной машины на одном из ESXi:

- ☐ в некоторых (на мой взгляд, маловероятных) ситуациях проблемы с виртуальной инфраструктурой могут вызвать недоступность vCenter, а без него не будет централизованного доступа к виртуальной инфраструктуре, что может усложнить решение исходной проблемы. Впрочем, сам я с трудом верю, что подобная ситуация реальна (в большинстве случаев, по крайней мере).

Наконец, возможны ситуации, что у вас не будет выбора. Очень характерный пример – если ваша инфраструктура подпадает под требования регуляторов и вы обязаны использовать дополнительные средства обеспечения безопасности для

vCenter. Например – решения для организации доверенной загрузки. С очень большой вероятностью такие средства могут не заработать в виртуальной машине, и вы должны будете обзавестись физическим сервером для vCenter.

Если вы приняли решение использовать под vCenter виртуальную машину, то каких-то особых нюансов у такой установки нет. Устанавливаем ESXi сервер, на свою машину устанавливаем клиент vSphere, подключаемся с его помощью к ESXi и создаем там ВМ. Устанавливаем в нее ОС, затем vCenter Server. Все.

БД для vCenter Server

В состав дистрибутива vCenter входит дистрибутив SQL Server 2008 R2 Express, бесплатной версии СУБД от Microsoft. Его идеально использовать для тестовых и демонстрационных стендов, а также можно применять и в производственной среде. Правда, сама VMware не рекомендует использовать SQL Server Express в случае, если размер вашей инфраструктуры превышает 5 серверов и 50 ВМ.

Насколько я могу судить, единственным реально ограничивающим нас фактором является ограничение в 10 Гб на размер базы данных. Для справки – при настройках по умолчанию для сбора событий и данных о производительности с серверов ESXi ожидаемый размер базы для инфраструктуры 100 хостов/1000 ВМ – порядка 8 Гб. Так что с технической точки зрения ограничение 5/50 – весьма не жесткое, но оно важно для тех из вас, кому важен «поддерживаемый» статус инфраструктуры. Дело в том, что даже для инфраструктуры размера на порядок большего, чем 5 хостов / 50 ВМ, база будет размером менее 10 Гб – то есть SQL Express технически будет улаживать. Однако мне встречались ситуации, например, когда при штатной работе большой инфраструктуры все было хорошо, а потом инфраструктуру обновляли на новую версию vSphere – и внезапно размер БД заметно увеличивался, доходил до предельного, и vCenter переставал работать из-за остановившейся базы. Имея перед глазами такие примеры, я серьезно отношусь к ограничению 5/50 для бесплатных БД сервера vCenter.

Из коммерческих СУБД поддерживаются разные версии следующих продуктов:

- ☐ IBM DB2;
- ☐ Microsoft SQL Server 2005;
- ☐ Microsoft SQL Server 2008;
- ☐ Oracle 10g;
- ☐ Oracle 11g.

Точный список для вашей версии ищите в документации. БД может располагаться как на одном сервере с vCenter, так и на выделенном сервере. Из стандартных соображений надежности и производительности рекомендуется второй вариант – впрочем, такая рекомендация начинает иметь смысл, лишь если размер инфраструктуры достигает хотя бы нескольких десятков физических серверов.

Все версии БД, кроме SQL Server 2008 Express, требуют дополнительной настройки – см. документ **vSphere Installation and Setup ⇒ Preparing vCenter Server Databases**. Если сервер БД не выделен под хранение базы vCenter, то рекомендуется перед установкой vCenter сделать резервную копию остальных БД.

Обратите внимание. Операции по обслуживанию базы данных vCenter Server выполняются средствами самой БД. В случае Microsoft SQL-сервер см. подробности в базе знаний, статьи <http://kb.vmware.com/kb/1004382> и <http://kb.vmware.com/kb/2006097>.

Совместимость vCenter Server 5 и vSphere Client с предыдущими версиями ESX(i) и vCenter

С помощью vCenter Server 5 вы можете управлять серверами ESXi большинства версий (не всех!), начиная с ESX 3.0 Update 1.

С помощью vSphere Client вы можете напрямую управлять ESX-серверами версий, начиная с 2.5 включительно, и Virtual Center, начиная с версии 2.5 включительно. Напомню, что соответствие версий ESXi и Virtual Center следующее:

1. ESX 2.0 – Virtual Center 1;
2. ESX 2.5 – Virtual Center 1.5;
3. ESX 3.0 – Virtual Center 2.0;
4. ESXi 3.5 – Virtual Center 2.5;
5. ESXi 4.# – vCenter Server 4.#;
6. ESXi 5.# – vCenter Server 5.#.

При подключении к ESX(i)/vCenter версий до 5 вам предложат загрузить с него клиент vSphere его версии и управлять этим ESX(i) через клиент родной для него версии. Клиент предыдущей версии установится в вашей системе параллельно с vSphere Client 5.

Для получения полной и актуальной информации о совместимости компонентов vSphere разных версий рекомендую обратиться к инструменту **VMware Product Interoperability Matrixes**, доступному на сайте VMware.

Установка vCenter Server

Сервер для установки vCenter может быть участником как домена, так и рабочей группы. В последнем случае не будет работать vCenter Linked Mode.

Для сервера vCenter рекомендуется настроить статический IP-адрес. Статичность адреса для сервера vCenter обусловлена тем, что агенты vCenter на серверах ESXi сохраняют IP-адрес сервера vCenter в своем конфигурационном файле (vrxa.cfg). Если адрес сервера vCenter изменится, сервера ESXi будут отображаться как недоступные, и придется для каждого выполнить операцию connect, после чего агенты vCenter узнают и запомнят уже новый адрес.

Обратите внимание. Если вам потребовалось сменить IP-адрес для уже установленного и используемого vCenter Server и сервера ESXi отключились со статусом Disconnected, то правильная последовательность шагов для возвращения их в строй описана в статье базы знаний <http://kb.vmware.com/kb/1001493>. Если меняется IP-адрес для Update Manager, то см. <http://kb.vmware.com/kb/1013222>.

Между серверами ESXi и vCenter не должен использоваться NAT.

Далее я перечислю, что вам следует знать для прохождения мастера установки vCenter.

Для запуска службы vCenter можно использовать как пользовательскую учетную запись, так и встроенную учетную запись Local System. Использование пользовательской учетной записи нужно в основном при применении Windows authentication на SQL Server.

Запустив autoran.exe из корня дистрибутива, вы увидите меню выбора продукта для установки. Обратите внимание на то, что в правой части этого меню есть ссылки для установки дополнительных вспомогательных компонентов. Для vCenter Server это Microsoft .NET 3.5 SP1 и Windows Installer 4.5 – эти компоненты может потребоваться установить отдельно при установке на некоторые версии ОС. Если будет необходимо, вам об этом сообщит установщик компонента vSphere.

Компоненты, входящие в состав дистрибутива vCenter Server:

- ☐ **VMware vCenter Server** – сама служба vCenter, именно ее мы устанавливаем;
- ☐ **vSphere Client** – клиент для подключения к vCenter и доступа ко всем настройкам и функциям. Его следует установить на те машины, с которых мы будем непосредственно администрировать виртуальную среду (это то же приложение, что используется для управления ESXi напрямую, без vCenter);
- ☐ **Microsoft.NET Framework** – необходимый компонент для работы vCenter и vSphere Client. Будет установлен автоматически, если отсутствует в системе;
- ☐ **Microsoft SQL Server 2008 R2 Express** – бесплатная версия СУБД от Microsoft, может использоваться в качестве БД для vCenter. Если указать ее использование в мастере установки – будет установлена автоматически. Если вы планируете использовать существующую БД, то выбирать использование SQL Server 2008 Express не надо;
- ☐ **VMware vCenter Orchestrator** – дополнительный продукт, предназначенный для автоматизации задач администрирования виртуальной инфраструктуры. Устанавливается автоматически, в случае если ОС использует IPv4, и недоступен в случае использования IPv6. Начиная с версии 4.1, vCenter Orchestrator предлагает новые предустановленные процессы для автоматизации задач администрирования, такие как управление снимками состояния, забытыми VMDK-файлами, множественные операции конвертирования thick-дисков в thin и отключение съемных устройств через vCenter. Об этом продукте в книге я писать не буду, базовую информацию можно почерпнуть по ссылке <http://link.vm4.ru/orchestrator>;
- ☐ **vCenter Update Manager** – дополнительный компонент vCenter, предназначенный для обновления серверов ESXi и VM (некоторых гостевых ОС и приложений). См. посвященный ему раздел. Может быть установлен как на машину с vCenter, так и на выделенный сервер или VM;
- ☐ **VMware vSphere Web Client (Server)** – веб-сервер, предоставляющий веб-интерфейс для администрирования и доступа к виртуальным машинам на vSphere;

- ❑ **VMware ESXi Dump Collector** – сервис для сбора через сеть диагностической информации в случае критического сбоя ESXi;
- ❑ **VMware Syslog Collector** – сервис сбора файлов журналов ESXi;
- ❑ **VMware Auto Deploy** – сервис организации PXE-загрузки серверов ESXi;
- ❑ **VMware Authentication Proxy** – сервис, позволяющий добавить сервера ESXi в домен AD без сохранения учетных данных на самом сервере ESXi (востребован вместе с Auto Deploy).

vCenter может использоваться сам по себе, в независимом (Standalone) варианте, или в составе группы серверов vCenter – последний режим называется «Linked Mode». Данный режим может быть интересен, если в вашей компании эксплуатируются несколько серверов vCenter – для разных групп серверов ESXi. В случае использования Linked Mode вы сможете управлять всей инфраструктурой из одного окна. Варианту Linked Mode посвящен следующий раздел, сейчас мы говорим про установку в варианте Standalone.

Из важных вопросов установщика – выбор БД. Или используйте SQL Server 2008 Express (тогда она будет установлена автоматически) либо какую-то существующую БД. В последнем случае необходимо отдельно настроить подключение к ней – см. подробности для вашей версии БД в документе **vSphere Installation and Setup ⇒ Preparing vCenter Server Databases**. SQL Express не рекомендуется использовать для инфраструктуры размером более 5 серверов/50 виртуальных машин. Обусловлено это техническими ограничениями для данной версии БД: один процессор, 1 Гб памяти, размер базы до 10 Гб.

Укажите учетную запись – System или пользовательскую, если вы используете Windows Authentication для SQL Server.

Выберите, будет ли это отдельный vCenter Server или он будет входить в состав группы – Linked Mode. Второй вариант разбирается в следующем разделе.

После установки vCenter для работы с ним необходим клиент vSphere. Дистрибутив клиента можно взять из дистрибутива vCenter Server или с веб-интерфейса vCenter.

Если этот сервер vCenter должен управлять ESX(i) серверами версий 3.x, то необходимо сохранить (или установить) сервер лицензий (тот, что использовался в VI 3).

Linked Mode

vCenter версии 5 позволяет объединять несколько серверов vCenter в группу. Подключившись к одному серверу vCenter из такой группы, мы можем видеть и управлять объектами каждого сервера vCenter из группы. Это очень удобно, если у вас есть несколько серверов vCenter. Например, отдельные сервера для производственной и тестовой инфраструктур или разные vCenter в разных ЦОД компании.

Подключить сервер vCenter к группе можно как на этапе установки, так и в произвольный момент позже.

Сервера vCenter, объединенные в группу, используют децентрализованную систему совместной работы (Peer-to-Peer). Если один сервер vCenter может об-

служивать до 1000 серверов ESXi и 10 000 включенных ВМ (рекомендательное ограничение), то до десяти vCenter в одной группе Linked Mode позволят вам мониторить и управлять до 3000 серверами ESXi и 30 000 ВМ.

Обратите внимание: эту возможность включает только vCenter Server с лицензией «Standard». То есть vCenter с лицензией Foundation или Essentials не получится добавить в группу Linked Mode. Linked Mode недоступен для Linux-версии vCenter, vCenter Virtual Appliance.

В объединенной таким образом инфраструктуре вам доступны:

- ☐ назначение глобальных ролей – то есть раздача прав по всей инфраструктуре;
- ☐ глобальный поиск объектов;
- ☐ управление всеми лицензиями.

Выглядит это так (рис. 1.6):

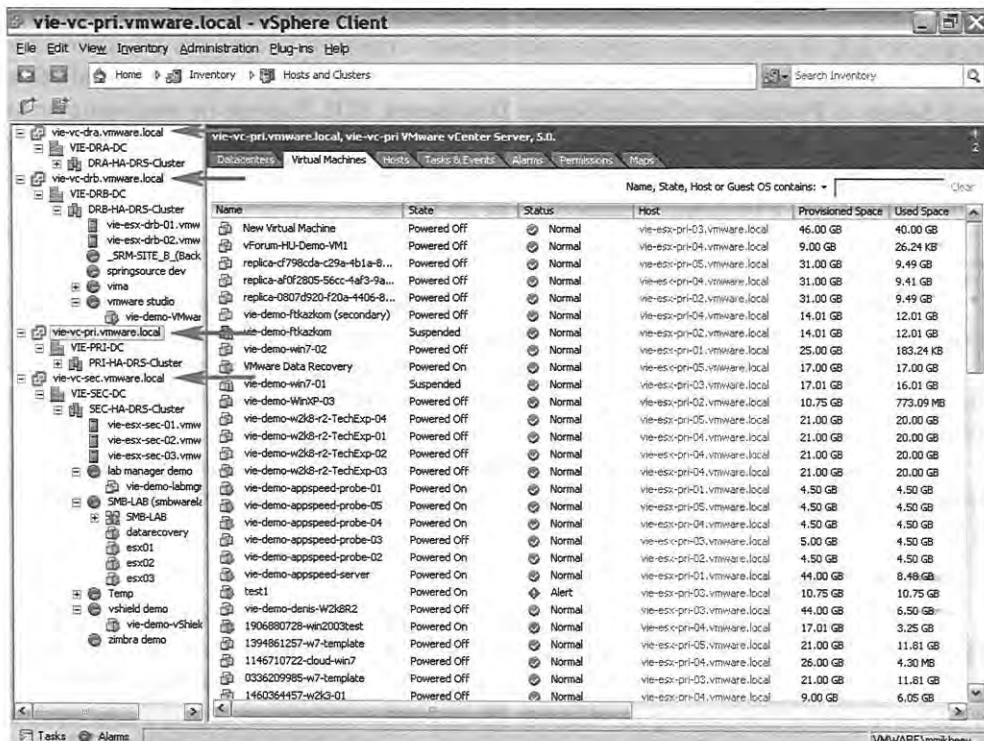


Рис. 1.6. Работа с несколькими vCenter в одном окне

Для хранения и синхронизации данных между экземплярами vCenter Server в группе Linked Mode использует Microsoft Active Directory Application Mode (ADAM). Сегодня ADAM известен как Microsoft Active Directory Lightweight Di-

rectory Services (AD LDS). ADAM устанавливается автоматически при установке vCenter Server. Каждый экземпляр ADAM хранит данные всех серверов vCenter одной группы. Эти данные содержат:

- ☐ информацию для соединения – IP-адреса и номера портов;
- ☐ сертификаты и их отпечатки (Thumbprints);
- ☐ лицензионную информацию;
- ☐ роли пользователей.

Один раз в день данные из ADAM выгружаются в резервную копию, которая хранится в БД сервера vCenter. В случае повреждения ADAM vCenter будет использовать данные из последней резервной копии для его восстановления.

Подключить vCenter к группе Linked Mode можно как на этапе установки, так и в любой другой момент. Пользователь, который запускает процесс присоединения сервера vCenter к группе, должен обладать правами локального администратора и на локальной системе, и на системе (системах), где установлены прочие сервера vCenter этой группы.

Требования к инфраструктуре следующие.

Все сервера vCenter должны находиться в одном домене AD или в разных доменах с двухсторонними доверительными отношениями. Само собой, важна правильная настройка синхронизации времени и DNS.

Но vCenter Server не должен быть установлен на контроллере домена – это воспрепятствует его добавлению в группу. Также не получится добавить vCenter в группу, если он установлен на терминальном сервере.

Если вы хотите объединить в группу несколько (например, три) серверов vCenter на этапе их установки, то ваши действия следующие.

1. Установить первый из них. Так как на этом этапе он является единственным, никакой группы для него не указываем.
2. Установить второй, когда установщик задаст вопрос про Linked Mode – указать FQDN сервера с первым vCenter. Теперь первые два vCenter в группе.
3. Третий vCenter, предположим, мы не устанавливаем с нуля, а обновляем с Virtual Center 4 до vCenter 5. Наши действия – сначала обновить, затем добавить к группе, указав FQDN первого или второго сервера vCenter. Нельзя включать в одну группу сервера vCenter версии 5 и версии 4 одновременно.

Теперь все три сервера vCenter принадлежат одной группе.

Если вы хотите добавить к группе уже установленный vCenter, то это также весьма не сложно: **Start ⇒ All Programs ⇒ VMware ⇒ vCenter Server Linked Mode Configuration**.

Выберите пункт **Modify Linked-Mode configuration** и укажите FQDN любого сервера vCenter, в группу с которым вы хотите добавить текущий.

Для удаления сервера vCenter из группы следует воспользоваться приложением «Programs and Features» (Программы и компоненты) или «Add or Remove program» (Установка и удаление программ) для Windows Server 2008 или более ранних версий соответственно. Выбрать там VMware vCenter Server, нажать **Change** и в мастере отказаться от членства в группе Linked Mode.

Прочие подробности, рекомендации и инструкции см. в актуальной версии **vSphere Installation and Setup** ⇒ **Preparing vCenter Server Databases** ⇒ **After You Install vCenter Server** ⇒ **Creating vCenter Server Linked Mode Groups**.

1.4.3. vCenter Virtual Appliance

В vSphere версии 5 появился кардинально новый вариант сервера vCenter – Linux-версия, поставляемая предустановленной в виртуальной машине. Эта виртуальная машина называется vCenter Server Virtual Appliance, vCSA.

В данной книге я сделаю упор на Windows-версию vCenter, а в этом разделе постараюсь дать достаточную информацию по развертыванию vCSA и по отличиям этого варианта vCenter от «классического».

Различия между Windows- и Linux-версиями vCenter

Первое, о чем кажется важным сказать, – практически все функции vSphere работают абсолютно одинаково с любой версией vCSA. Если вы выполняете какие-либо действия при помощи клиента vSphere, то не сможете отличить, к какому варианту vCenter подключен ваш клиент. Таким образом, отличия между версиями больше организационные.

Различие первоочередное – операционная система Windows или SUSE Linux Enterprise Server 11 x64. То, какая ОС вам больше знакома, – первый аргумент для выбора. Кроме того, в небольших инфраструктурах лишняя лицензия для Windows может быть заметна (но в силу специфики правил лицензирования Microsoft это не будет аргументом в инфраструктурах покрупнее).

Различие номер два – поддерживаемая база данных. Если ваша инфраструктура превышает по размерам 5 хостов или 50 ВМ, то VMware рекомендует не использовать идущую в комплекте с vCSA базу данных DB2. Но из коммерческих БД поддерживается лишь Oracle (на момент написания), что делает vCSA плохим выбором, когда в вашей компании не используется эта база данных. Впрочем, насколько знаю я, предустановленная версия DB2 не обладает ограничением на размер базы, поэтому упомянутый размер инфраструктуры 5/50 – это ограничение на поддерживаемость инфраструктуры, не на работоспособность.

Более мелкие отличия:

- ☐ в случае Windows-версии vCenter есть возможность установить VMware Update Manager на тот же сервер что и vCenter, – в случае vCSA обязательно потребуется отдельный сервер (ВМ);
- ☐ vCSA не поддерживает Linked Mode;
- ☐ vCSA не поддерживает IPv6;
- ☐ vCSA не поддерживает vSphere Virtual Storage Appliance, VSA;
- ☐ vCSA нельзя защитить при помощи vCenter Server Heartbeat, кластерной службы для vCenter.

Могут быть проблемы совместимости с дополнительными продуктами. Все, что относится непосредственно к vSphere с vCSA, совместимо (а то и предустановлено), большинство связанных с vSphere продуктов VMware совместимо – но

не 100%. Например, VMware View требует установки на vCenter компонента View Composer для использования Linked Clones – сегодня этот продукт есть только в Windows-версии.

Установка и настройка vCSA

Ввод vCSA в эксплуатацию состоит из нескольких этапов:

1. Развертывание этого Virtual Appliance.
2. Настройка БД и старт службы vCenter.
3. Дополнительные настройки.

О чем мне кажется важным упомянуть с точки зрения планирования перед развертыванием:

- ☐ vCSA может потребить до 80 Гб места на хранилище;
- ☐ в зависимости от размеров инфраструктуры вам может потребоваться увеличить количество памяти для виртуальной машины vCSA. VMware выделяет следующие градации (хостов/BM – ОЗУ):
 - до 10/100 – от 4 Гб;
 - между 10/100 и 100/1000 – от 8 Гб;
 - от 100/400 до 1000/4000 – от 13 Гб;
 - свыше 400/4000 – от 17 Гб.

Развертывание vCSA крайне несложно – подключившись к серверу ESXi клиентом vSphere, выберите пункт меню **File** ⇒ **Deploy OVF Template**. Выбрав в открывшемся мастере заранее загруженный OVF-файл vCSA, пройдите мастер до конца, выбрав хранилище, на которое следует скопировать vCSA из ovf-пакета.

После завершения импорта vCenter Appliance включите эту VM. Наша следующая задача – настроить сеть. Если не устраивает получение настроек IP по DHCP, выполните эти настройки вручную, открыв консоль к этой VM и выбрав пункт **Configure Network**.

Кроме того, здесь же можно и стоит настроить часовой пояс.

После настройки сети вам потребуется браузер – обратитесь на VCSA по https на порт 5480. Для авторизации используйте root/vmware.

Программа-минимум для настройки:

1. Принять EULA.
2. Настроить БД.
3. Стартовать службу vCenter.

Принятие лицензионного соглашения – первое, что вы увидите после первой авторизации.

Для настройки БД вам потребуется перейти на вкладку **Database** (оставаясь на вкладке более высокого уровня **vCenter Server**).

Для использования комплектной базы данных достаточно выбрать «embedded» в выпадающем меню пункта **Database Type** и нажать кнопку **Save Settings**. Дожидаемся надписи **Operation was succesful**.

После этого на вкладке **Status** нажмите кнопку **Start vCenter**.

Все. Начиная с этого момента вы можете подключаться к этому vCenter клиентом vSphere и добавлять в него сервера ESXi.

Дальнейшие настройки уже необязательны для работы vCenter как такового, но могут быть полезны.

На вкладке **Authentication** ⇒ **Active Directory** вы можете указать настройки интеграции с AD и использовать доменные учетные записи для авторизации в vCenter Appliance. Только не забудьте дать права нужным группам (вкладка **Permissions** в клиенте vSphere). По умолчанию никакие доменные группы или пользователи не обладают правами на объекты иерархии vCenter.

Также поддерживается авторизация через службу каталогов NIS, Network Information Services.

Сменить пароль пользователя root можно на вкладке **vCenter Server** ⇒ **Administration**.

При необходимости использовать Syslog Collector, ESXi Dump Collector и Auto Deploy достаточно пробежаться по одноименным вкладкам, сделать настройки (сетевых портов в основном) и сохранить настройки. Последним шагом будет перезагрузка vCSA.

Можно настроить перенаправление файлов журналов vCSA на NFS-ресурс. Для этого в веб-интерфейсе vCSA перейдите на вкладку **vCenter Server** ⇒ **Storage**. Там поставьте флажки и укажите сервер и имя NFS-ресурса вида «192.168.22.249:/vCSA_NFS». **Test Settings** ⇒ **Save Settings** ⇒ перезагрузите vCSA для применения настроек.

1.5. Интерфейс клиента vSphere, vCenter, ESXi. Веб-интерфейс

Здесь я опишу основные элементы интерфейса. Упор будет сделан на работу через vCenter. Впрочем, при работе с ESXi напрямую все очень похоже, просто недоступна часть объектов (например, кластер HA/DRS) или функций (vMotion).

1.5.1. Элементы интерфейса клиента vSphere при подключении к vCenter

Итак, вы подключились к vCenter с помощью клиента vSphere. Дистрибутив клиента входит в дистрибутив vCenter. Также клиент vSphere доступен для загрузки на веб-интерфейсе vCenter и ESXi. Устанавливаем клиент на свой компьютер, получив дистрибутив из удобного источника.

В самом начале вам потребуется учетная запись, имеющая права локального администратора на системе, где установлен vCenter Server. Это может быть как локальная, так и доменная учетная запись. Впоследствии вы можете назначать различные роли и давать права в vCenter любым учетным записям (не обязательно имеющим административные права в ОС).

Подключив к vCenter сервер, вы увидите примерно такую картину (рис. 1.7):

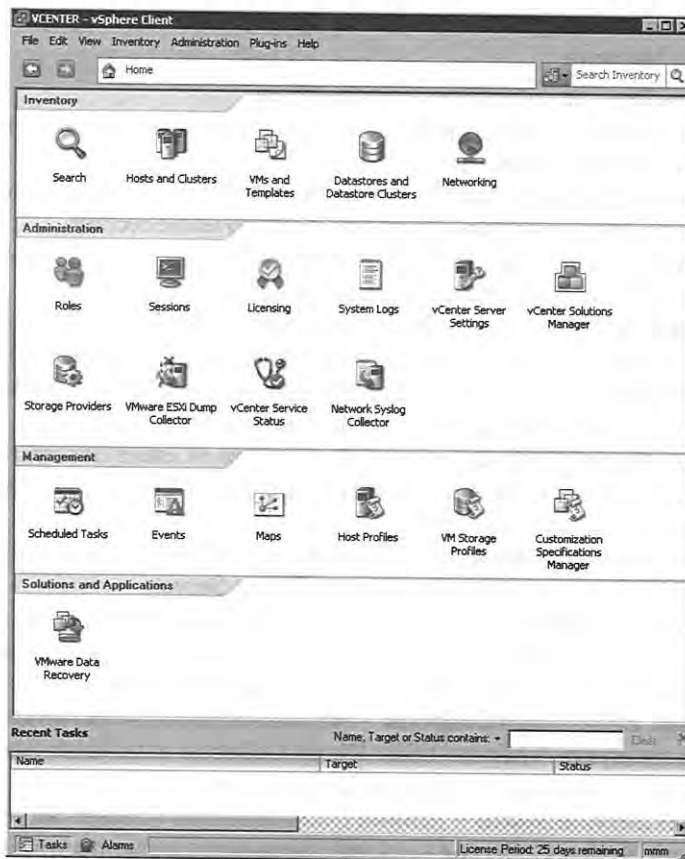


Рис. 1.7. Интерфейс vCenter

Как видите, элементы интерфейса поделены на четыре большие категории.

Inventory – здесь сгруппированы основные элементы интерфейса по работе со всеми объектами виртуальной инфраструктуры:

1. **Hosts and Clusters** – один из часто используемых пунктов. В нем мы настраиваем наши сервера, кластеры, виртуальные машины.
2. **VMs and Templates** – второй из часто используемых. В этом представлении интерфейса отображаются виртуальные машины и шаблоны, а также папки для них – иерархия виртуальных машин здесь не зависит от их физического размещения на тех или иных серверах и кластерах. Удобно, когда необходимо работать с виртуальными машинами, и только с ними, особенно если требуется их как-то сгруппировать по организационному признаку.
3. **Datastores** – здесь отображаются все хранилища, подключенные хотя бы к одному ESXi нашей инфраструктуры.
4. **Networking** – здесь отображаются группы портов для виртуальных машин на стандартных виртуальных коммутаторах, и, главное, здесь и только

здесь можно создавать и настраивать распределенные виртуальные коммутаторы VMware.

Administration – задачи администрирования:

1. **Roles** – отсюда настраиваем и получаем информацию о раздаче прав в виртуальной инфраструктуре.
2. **Sessions** – просмотр текущих сеансов работы с vCenter, рассылка сообщений подключенным пользователям и отключение сессий.
3. **Licensing** – настройка лицензирования vCenter и ESXi. Именно здесь указываем ключи продукта и назначаем их разным серверам.
4. **System Logs** – журналы событий vCenter (когда клиент vSphere подключен к vCenter) и ESXi (когда клиент подключен к ESXi напрямую). Пригодятся при решении проблем. Это намного удобнее, чем искать файлы журналов самостоятельно. Так доступны не все, но основные журналы.
5. **vCenter Server Settings** – настройки самого vCenter Server. Именно здесь можно указать, например, почтовый сервер для рассылки оповещений или настройки оповещения по SNMP.
6. **vCenter Solution Manager** – интерфейс получения информации о сторонних продуктах, интегрированных с vCenter Server.
7. **Storage Providers** – интерфейс получения информации о сторонних «VASA-провайдерах», продуктах сторонних фирм, которые обеспечивают интеграцию vCenter Server и системы хранения по протоколу VASA.
8. **VMware ESXi Dump Collector** – эта пиктограмма появляется после интеграции продукта VMware ESXi Dump Collector и vCenter Server. Под ней доступна информация о сервере Dump Collector.
9. **vCenter Service Status** – данные о компонентах vCenter Server.
10. **Network Syslog Collector** – эта пиктограмма появляется после интеграции продукта VMware Syslog Collector и vCenter Server. Под ней доступна информация о сервере Syslog.

Management – вспомогательные возможности vCenter:

1. **Scheduled Tasks** – планировщик задач vCenter Server. Обратите на него внимание – с его помощью можно запланировать на удобное время многие операции, такие как включение и выключение ВМ, развертывание ВМ из шаблона, миграция ВМ, снимки состояния ВМ и др.
2. **Events** – все события, которые vCenter получает с серверов и генерирует сам.
3. **Maps** – этот механизм позволяет строить графические схемы связей между объектами инфраструктуры. Если стоит задача посмотреть, какие ВМ лежат на тех или иных хранилищах или на каких серверах существуют те или иные сети, то вам сюда.
4. **Host Profiles** – однажды созданный профиль настроек доступен для редактирования в этом разделе интерфейса.
5. **VM Storage Profiles** – настройки механизма Profile Driven Storage, создание и изменение «профилей хранилищ».

6. **Customization Specification Manager** – vCenter позволяет обезличивать некоторые гостевые ОС при клонировании или развертывании из шаблона. Сохраненные файлы ответов мастера развертывания попадают сюда. Здесь же можно их импортировать или экспортировать, создавать и редактировать.

Solution and Application – сюда попадают функции, которые появляются в vSphere через установку дополнительных приложений-плагинов. На рис. 1.10 вы видите иконку для управления решением резервного копирования VMware – **VMware Data Recovery**.

Когда вы переходите к какому-то элементу интерфейса со страницы **Home**, то адресная строка меняется соответствующим образом. Также через нее можно получать быстрый доступ к другим частям интерфейса (рис. 1.8).

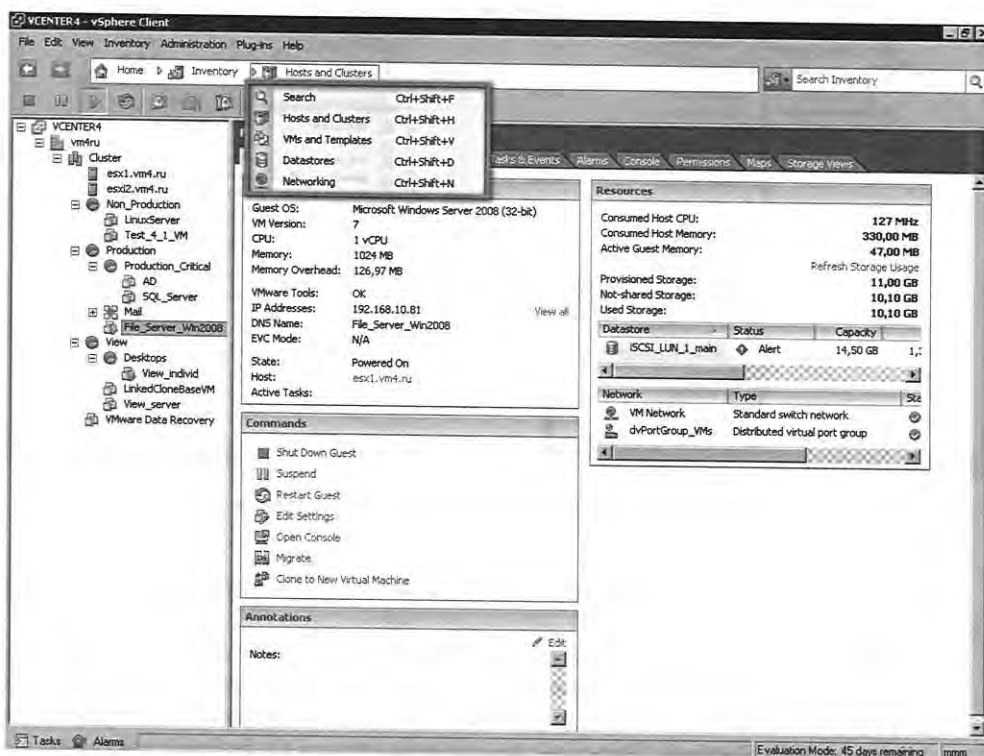


Рис. 1.8. Меню адресной строки

По умолчанию для каждого объекта клиент vSphere показывает вкладку **Getting Started**. Я рекомендую отключить эту вкладку, потому что без нее вам сразу будет показываться несравнимо более информативная вкладка **Summary**. Отключить показ вкладок **Getting Started** для объектов всех типов зараз можно в меню **Edit ⇒ Client Settings ⇒** снять флажок **Show Getting Started Tab**.

Еще немного подробностей про интерфейс на примере отдельного сервера. Пройдите в раздел **Hosts and Clusters**, выделите сервер и обратите внимание на вкладки (рис. 1.9).

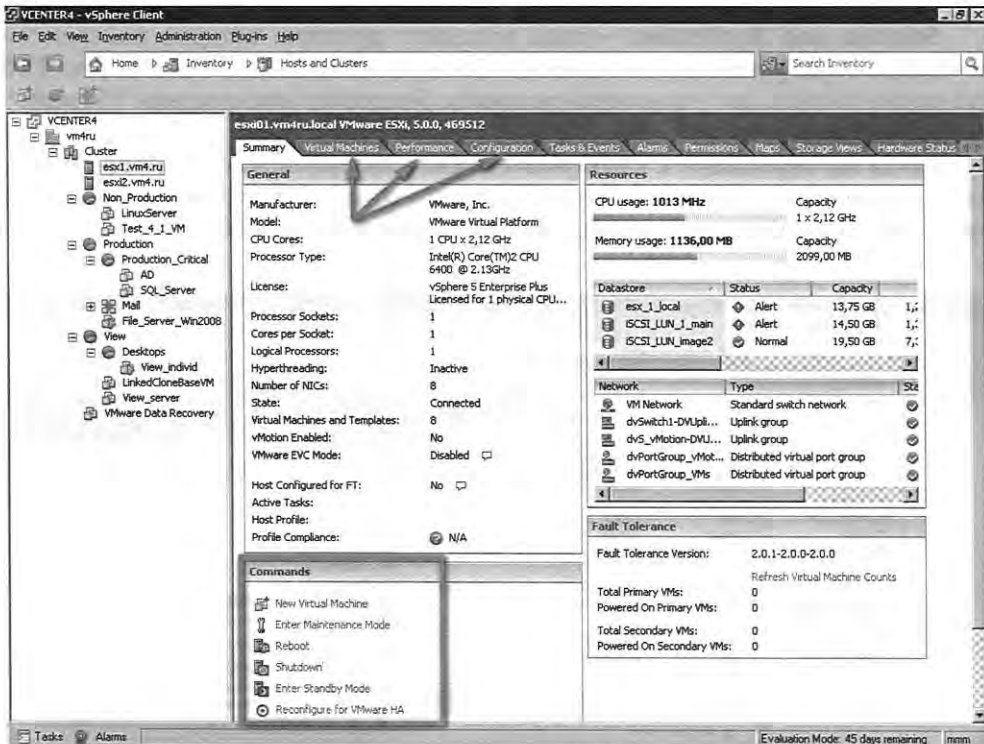


Рис. 1.9. Элементы интерфейса для сервера

Большинство вкладок доступны для объекта vCenter любого типа. Перечислю вкладки и их функции:

1. **Summary** – сводная информации об объекте. Содержит секцию **Commands**, в которой доступны основные манипуляции с объектом. Например, для виртуальных машин отсюда удобно попадать в настройки, доступные по строке **Edit Settings**. Впрочем, все пункты отсюда (и другие) доступны через контекстное меню объекта.
2. **Virtual Machines** – отображает виртуальные машины и шаблоны, являющиеся дочерними к данному объекту. Вкладка доступна для большинства типов объектов vCenter. Для сервера на этой вкладке отображаются виртуальные машины, размещенные непосредственно на нем. Для хранилища – чьи файлы располагаются на данном хранилище. Для распределенной группы портов – подключенные к ней VM.

3. **Performance** – на этой вкладке мы смотрим графики нагрузки на объект по разнообразным счетчикам производительности. Для объектов разных типов доступны разные наборы счетчиков.
4. **Configuration** – здесь мы настраиваем сервера. Эта вкладка существует только для них.
5. **Tasks & Events** – события и задачи выбранного объекта или его дочерних объектов.
6. **Alarms** – на этой вкладке настраиваются и отображаются сработавшие предупреждения (alarm) vCenter.
7. **Permissions** – на этой вкладке отображаются пользователи и группы, имеющие права на данный объект.
8. **Maps** – отображается схема связей данного объекта с другими объектами vCenter. Для виртуальных машин на этой вкладке отображается так называемая vMotion map (карта vMotion), которая показывает возможность или невозможность живой миграции этой VM на другие сервера.
9. **Storage Views** – здесь отображается разнообразная информация про подсистему хранения в контексте выделенного сейчас объекта. То есть на этой вкладке для виртуальной машины отображается информация о хранилищах, занятых ее файлами. Для сервера – всех его хранилищ. Для кластера – всех хранилищ всех его узлов.

У меня частенько будут попадаться конструкции вида «Пройдите **Configuration** ⇒ куда-то далее и сделайте то-то и то-то». Это означает, что для описываемых манипуляций вам нужна вкладка **Configuration**. Существует она только для серверов; чтобы ее найти, пройдите **Home** ⇒ **Hosts and Clusters** ⇒ **Configuration**.

Также обратите внимание на кнопки **Tasks** и **Alarms** в нижней левой части экрана. Они отображают или скрывают панель текущих задач и активных предупреждений. Особенно полезно окно **Tasks** – здесь видно, закончилось или нет то или иное действие, успешно закончилось или нет, и некоторые действия можно отменить (рис. 1.10).

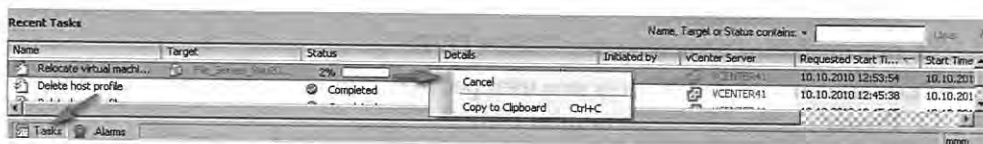


Рис. 1.10. Панель **Tasks**

Для виртуальных машин есть несколько специфических элементов интерфейса. В контекстном меню VM выберите пункт **Open Console**. Откроется консоль к этой виртуальной машине (рис. 1.11).

Здесь вы видите:

- ☐ пиктограммы управления питанием VM;

Обратите внимание: пиктограммы выключения и перезагрузки настроены на корректное выключение и корректную перезагрузку. Эти операции требуют VMware

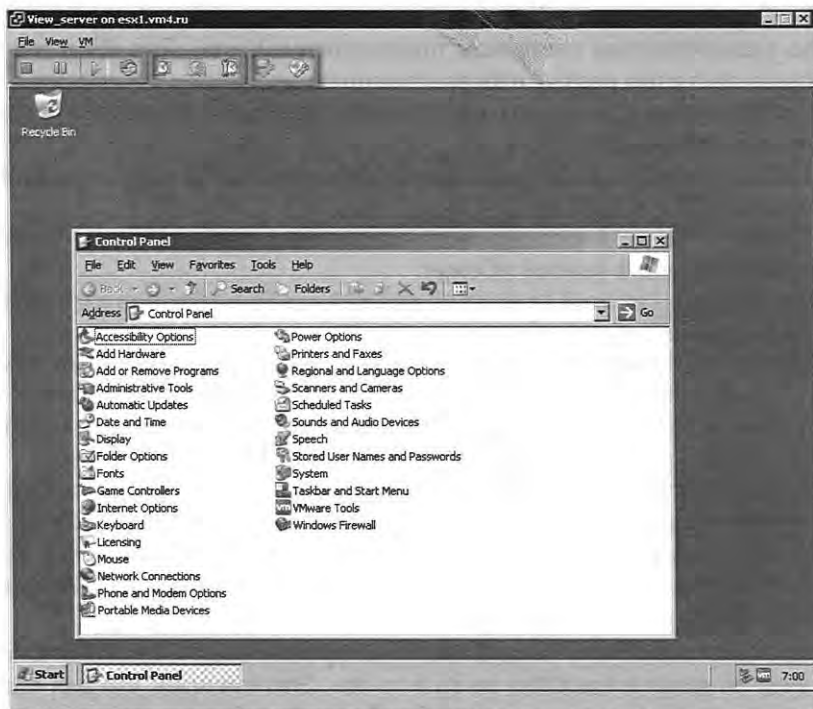


Рис. 1.11. Консоль ВМ

tools. Если их нет или они еще не загрузились, то для выполнения «жестких» операций с питанием пользуйтесь пунктами меню **VM** ⇒ **Power**.

- ☐ пиктограммы работы со снимками состояния (snapshot) – создание, возврат к ранее созданному снимку и запуск диспетчера снимков состояния (Snapshot Manager);
- ☐ пиктограммы подключения FDD и DVD.

Если вызвать меню **VM**, то в нем интерес представляют пункты:

- ☐ **Power** – управление питанием ВМ;
- ☐ **Guest** – из этого меню инициируется установка VMware Tools и посылается комбинация **Ctrl+Alt+Del**. Чтобы отправить внутрь ВМ такую комбинацию с клавиатуры, следует нажать **Ctrl+Alt+Ins**.

Когда в ВМ не установлены VMware Tools, то после щелчка мышью внутри консоли фокус ввода принадлежит уже ей, а не вашей «внешней» ОС. Чтобы «извлечь» курсор из консоли, нажмите **Ctrl+Alt**.

Отличительными чертами этой консоли являются то, что она «аппаратная», то есть ее работоспособность не зависит от ПО внутри ВМ, и то, что ее график идет по управляющей сети ESXi. Таким образом, чтобы открыть консоль к виртуальной машине, вам нет необходимости находиться в одной с ней сети. Однако даже

при работе через vCenter эта консоль открывается с конкретного сервера ESXi. Это означает, что сервера ESXi должны быть доступны с той машины, откуда вы открываете данную консоль. Если сервер ESXi добавлен в vCenter по имени, то необходимо еще, чтобы это имя разрешалось с машины клиента.

Еще несколько слов про работу в консоли – если обратиться к пункту меню **View**, то в нем обнаружится пара очень полезных пунктов:

- ☐ **Fit Windows Now** – сделать размер окна консоли соответствующим разрешению гостевой ОС;
- ☐ **Fit Guest Now** – сделать разрешение гостевой ОС соответствующим размеру окна консоли.

Эти пункты помогут в случае, если окно консоли заметно меньше или заметно больше разрешения гостя и/или вашего рабочего места.

Обратите внимание на **Home** ⇒ **Administration** ⇒ **System Logs**. С помощью этого элемента интерфейса вы можете получить доступ к журналам vCenter (когда клиент vSphere подключен к vCenter) и ESXi (когда клиент подключен к ESXi). Это намного удобнее, чем искать эти файлы журналов самостоятельно. Так доступны не все, но основные журналы.

Нажав в любом окне интерфейса **Ctrl+Shift+F**, вы попадете в окно расширенного поиска. Отсюда возможен поиск объектов любого типа, по любым критериям и с учетом условий. Например:

- ☐ все выключенные виртуальные машины;
- ☐ все виртуальные машины с устаревшей версией VMware tools;
- ☐ все виртуальные машины со строкой «х» в поле **Description** (Описание);
- ☐ все виртуальные машины, для которых в качестве гостевой ОС указан Linux;
- ☐ все сервера ESXi во включенном или выключенном состоянии;
- ☐ все хранилища, свободного места на которых меньше указанного значения.

Обратите внимание, что поиск возможен и по так называемым **Custom Attributes** (Произвольным атрибутам). Задать такие атрибуты можно использовать для хостов и виртуальных машин. Сначала в настройках vCenter: меню **Administration** ⇒ **Custom Attributes** ⇒ **Add**. А затем на вкладке **Summary** ⇒ ссылка **Edit** в поле **Annotations** (рис. 1.12).

Наконец, обращаю ваше внимание на то, что со списком объектов в правой части окна клиента vSphere (например, список виртуальных машин на вкладке **Virtual Machines**) можно осуществлять разные операции:

- ☐ сортировать по любому столбцу, кликом по его заголовку;
- ☐ изменять набор отображаемых столбцов. Для этого в контекстном меню пустого места выберите пункт **View Column**;
- ☐ фильтровать по подстроке в некоторых столбцах (например, имя, состояние, тип гостевой ОС). В этом поможет поле в правой верхней части окна клиента vSphere, присутствующее на соответствующих вкладках;
- ☐ экспортировать список объектов (предварительно отфильтрованный, упорядоченный и с нужным набором столбцов). Для экспорта обратитесь

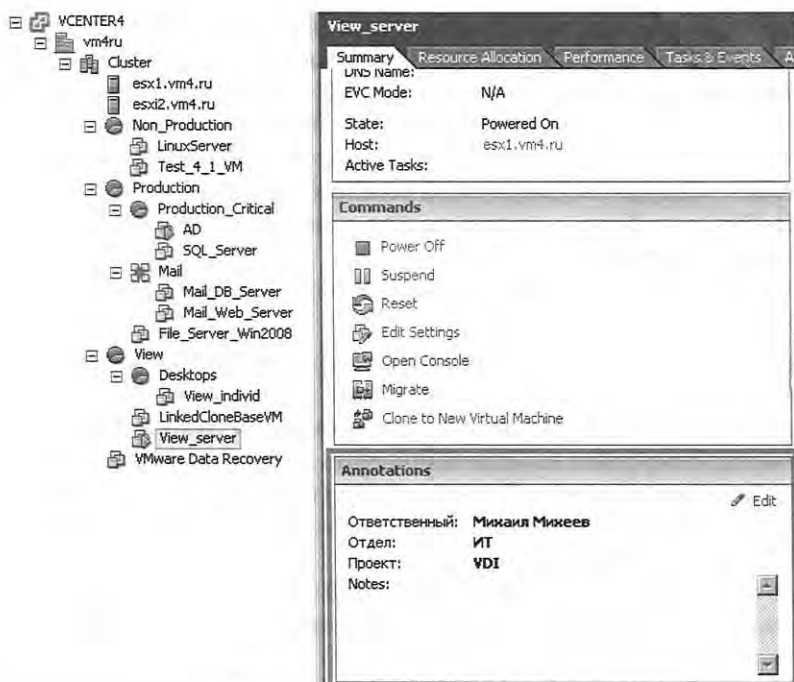


Рис. 1.12. Дополнительные поля для виртуальной машины

к пункту меню **File** ⇒ **Export** ⇒ **Export List**. В списке поддерживаемых форматов присутствуют html, csv, xls, xml.

Еще один заслуживающий внимания элемент интерфейса клиента vSphere – встроенный файловый менеджер. Он доступен в контекстном меню хранилища (datastore) ⇒ **Browse datastore**. С его помощью можно копировать, удалять, загружать и выгружать файлы с и на ESXi. А также регистрировать виртуальные машины и шаблоны, которые ранее были удалены из иерархии, но файлы которых остались на хранилищах.

Базовые шаги для решения проблем с клиентом vSphere

В случае возникновения ошибки при подключении клиентом vSphere к серверу vCenter первое, что необходимо проверить, – запущена ли служба VMware Virtual Center Server. Иногда при использовании локальной СУБД (особенно при использовании SQL Server Express) после перезагрузки сервера эта служба не запускается из-за того, что она пытается стартовать до того, как заработает БД. Решением проблемы является выставление зависимости службы vCenter от службы СУБД. К сожалению, даже выставление зависимости не всегда помогает.

Одно из решений для тестовой или демонстрационной инфраструктуры – настройка отложенного старта для служб vCenter и vCenter Management WebServices.

1.5.2. Первоначальная настройка vCenter и ESXi

Итак, у вас есть свежеставленный сервер или сервера ESXi и vCenter, вы установили на свою рабочую станцию клиент vSphere и подключились к vCenter, типовой список действий для подготовки виртуальной инфраструктуры к полноценной работе выглядит примерно так:

1. Добавление серверов ESXi в консоль vCenter.
2. Настройка лицензирования серверов ESXi и vCenter.
3. При необходимости настройки некоторых служб, таких как firewall, ntp, SSH, syslog. А также настройки DNS и шлюза по умолчанию.
4. Настройка сети. Это и настройка интерфейсов VMkernel, и настройка групп портов для виртуальных машин, и создание распределенных виртуальных коммутаторов.
5. Настройка хранилищ. Подключение систем хранения данных, создание разделов VMFS, проверка корректного обнаружения уже существующих VMFS.
6. Создание и настройка пулов ресурсов, кластеров HA и DRS.

Добавление серверов в консоль vCenter

Для добавления серверов в консоль vCenter необходимо, чтобы в ней существовал объект типа «Datacenter». Такой объект суть папка для объектов всех прочих типов – серверов, кластеров, виртуальных машин, хранилищ, сетей и прочего. Таким образом, с помощью папок Datacenter вы можете сгруппировать части инфраструктуры. Это пригодится в тех случаях, когда в вашей компании существуют несколько административных групп, управляющих независимыми виртуальными инфраструктурами. Все эти инфраструктуры управляются одним vCenter, из одной консоли, но на уровне прав можно ограничивать области видимости для разных пользователей. В тексте я буду называть объекты этого типа «датацентр».

Если у вас нет филиалов со своей инфраструктурой и администраторами, если у вас в компании нет отдельного отдела безопасности, который сам управляет своими ESXi, или подобных вариантов – то несколько Datacenter вам не нужно. Но хотя бы один создать придется – это требование vCenter.

Для создания вызовите контекстное меню для корневого объекта иерархии vCenter – его самого и выберите в нем пункт **New Datacenter**. Имя объекта выберите по своему усмотрению.

Теперь в контекстном меню уже созданного Datacenter выберите пункт **Add Host**. В запустившемся мастере укажите имя или IP-адрес сервера ESXi, пользователя root и его пароль. Предпочтительнее добавлять сервера по имени, причем по полному доменному имени (FQDN). Пароль пользователя root необходим vCenter, чтобы создать на этом ESXi своего собственного пользователя vpxuser, из-под которого в дальнейшем vCenter и будет подключаться к этому ESXi. Таким образом, последующая смена пароля root на ESXi-сервере не оказывает на vCenter никакого влияния.

Настройка лицензирования

При установке vCenter вы можете указать ключ продукта. А можете не указывать. Если ключ не указан, то vCenter начнет работать в «Evaluation» (оценочном) режиме. Для ESXi это вообще является единственно возможным вариантом – на этапе его установки ключ ввести нельзя. Но после установки даже для бесплатной версии ключ ввести необходимо.

Таким образом, если в вашей инфраструктуре есть объекты, для которых лицензия не была указана, то через 60 дней ознакомительная лицензия закончится, и они работать перестанут.

vCenter лицензируется поштучно, так что ключ для него должен содержать столько лицензий, сколько серверов vCenter вы планируете использовать. Часто один.

ESXi лицензируется по процессорам, и в ключе должно содержаться лицензий на столько процессоров (сокетов), сколько их совокупно во всех ваших ESXi. Разные сервера ESXi одной инфраструктуры могут лицензироваться разными ключами.

В пятой версии vSphere в правила лицензирования добавилось такое понятие, как vRAM. vRAM – это объем памяти, который выделен виртуальной машине (в частности, это тот объем, что вы указываете при создании VM). Каждая лицензия для ESXi дает право на использование какого-то объема vRAM. Например, если у вас 10 лицензий Enterprise Plus, то вы имеете право выдать VM $96 \text{ Гб} \times 10 = 960 \text{ Гб}$ памяти. Если этого вам недостаточно, то придется приобрести дополнительное количество лицензий. Получить информацию о ситуации с vRAM можно, пройдя **Home** $\Rightarrow$ **Licensing** $\Rightarrow$ **Reporting** (не работает без установленного Web Client Server).

Для указания лицензии пройдите **Home** $\Rightarrow$ **Licensing**. В этом окне вам необходимо добавить один или несколько ключей продукта. В каждом 25-символьном ключе зашифровано, какие лицензии он содержит и на какое количество объектов. В типовом случае у вас есть сервер vCenter и несколько серверов ESXi с лицензией какого-то одного типа. Значит, у вас будет минимум два ключа – для vCenter и для ESXi.

Запустите мастер Manage vSphere Licenses и пройдите по шагам:

1. **Add License Keys** – введите свои ключи здесь. Можно добавлять сразу несколько ключей, по одному в строку. Для каждого ключа можно ввести произвольную метку (**Label**) для упрощения управления лицензиями.
2. **Assign Licenses** – здесь вам покажут сервера vCenter и ESXi вашей инфраструктуры, и вы сможете указать, какой из них каким ключом необходимо отлицензировать.
3. **Remove License Keys** – здесь вы можете удалить какие-то ключи.

Одновременно под управлением одного сервера vCenter могут находиться сервера ESXi, лицензированные различными типами лицензий. Если вы обновляли какие-то лицензии, например со Standard на Enterprise Plus, то вам необходимо добавить новый ключ, указать его использование для серверов и удалить старый,

если он стал не нужен. Если вы перевели сервер с лицензии ознакомительной на какую-то из коммерческих с меньшим функционалом, то часть функций перестанет работать без предупреждения. Если сервер перестал быть лицензирован, например из-за окончания срока действия лицензии, то на таком сервере перестанут включаться виртуальные машины (хотя запущенные работать продолжат).

Также добавления отдельных ключей требуют некоторые дополнительные продукты, например Virtual Storage Appliance или компоненты vShield.

Рекомендуемые начальные настройки ESXi

Основные кандидаты на настройку из служб ESXi – это Firewall, клиент NTP и, возможно, сервер SSH.

В версии 5 все это настраивается из графического интерфейса. Пройдите **Home** ⇒ **Hosts and Clusters** ⇒ **Configuration** для настраиваемого сервера. В списке вас интересуют:

1. **Security Profile** – позволяет настраивать межсетевой экран и некоторые службы (нас в первую очередь будет интересовать ssh). Этот межсетевой экран защищает лишь интерфейсы гипервизора и никак не влияет на виртуальные машины.

Обратите внимание: для настройки из командной строки вам пригодится команда `esxcli network firewall`. Для централизованной настройки firewall вам пригодится механизм Host Profiles, см. посвященный ему раздел.

2. **Time Configuration** – в этом пункте вы можете включить клиент NTP на ESXi и указать ему настройки синхронизации времени.
3. **Licensed features** – здесь вы можете просмотреть информацию о действующих для этого сервера лицензиях. Также здесь можно настраивать лицензирование сервера, однако если сервер управляется через vCenter и лицензирование для этого сервера уже настроено через vCenter, то настройка, заданная на уровне сервера, будет отменена.
4. **DNS and Routing** – здесь можно указать имя сервера, суффикс домена DNS, адреса серверов DNS и шлюз по умолчанию.
5. **Virtual Machine Startup and Shutdown** – здесь настраиваются автозапуск виртуальных машин при включении сервера, порядок их включения и паузы между включениями разных VM. Также здесь можно указать, что делать с виртуальными машинами, когда сам сервер выключается. Варианты – корректное выключение, принудительное выключение, приостановка (Suspend).
6. **Authentication Services** – настройка аутентификации на ESXi при помощи учетных записей из Active Directory. Подробности см. в посвященной безопасности главе.

Пункт **Security Profile** довольно многофункциональный. Кроме брандмауэра, здесь вы сможете включить или выключить такие функции, как:

- ☐ **ESXi Shell** – доступность локальной командной строки в консоли ESXi;
- ☐ **SSH** – включение и отключение сервера SSH на ESXi;

- **Direct Console UI** – включение и отключение БИОС-подобного меню в локальной консоли ESXi (см. рис. 1.13).

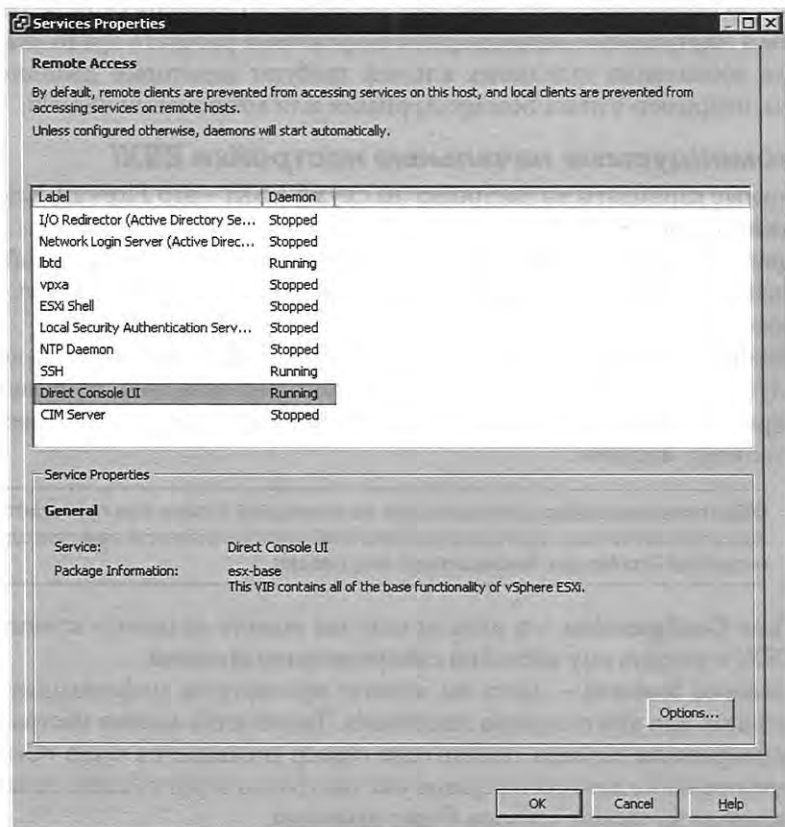


Рис. 1.13. Настройки Security Profile для ESXi 5

Пункт **Lockdown Mode** позволяет вам повысить безопасность ESXi. После включения этого флажка станет невозможно подключиться **напрямую** к ESXi каким бы то ни было интерфейсом или приложением и каким бы то ни было пользователем. Единственное исключение – пользователь `vxuser`, пароль от которого известен только vCenter. Таким образом, включение Lockdown-режима оставляет доступ к ESXi только из vCenter и локальной консоли. В локальной консоли нам доступны локальная командная строка и БИОС-подобное меню (его название – Direct Console User Interface, DCUI). Однако доступ к локальной командной строке по умолчанию выключен, DCUI может быть отключен (из тех же соображений повышения безопасности). Если локальная командная строка и DCUI выключены при включенном Lockdown Mode, то обязательно озаботьтесь повышением доступности vCenter Server. Когда в таких условиях vCenter потерял без возмож-

ности восстановления, единственный способ вернуть контроль над ESXi – это его переустановка.

Рискну предположить, что если перед вами не стоит задача обеспечить безопасность в соответствии с самыми жесткими требованиями – этот режим вам не нужен.

Возвращаясь к базовым настройкам серверов ESXi: пройдите **Configuration** ⇒ **Storage**. В этом окне отображаются разделы, отформатированные в VMFS. Скорее всего, здесь вы увидите раздел под названием Local1 или datastore1 – он был создан установщиком. Рекомендую переименовать этот раздел, например так: «local_esxi1». В дальнейшем это сильно поможет вам ориентироваться в сводных списках разделов VMFS в интерфейсе vCenter. Напоминаю, что в названиях лучше не использовать пробелы и спецсимволы, имена вида «local@esxi1» – это плохая идея, может вызвать проблемы совместимости со сторонними продуктами или сценариями.

Когда на ваших серверах уже созданы виртуальные машины, имеет смысл продуманно настроить порядок их автозапуска для служб и приложений, которые зависят друг от друга. Например, вот так: сначала ВМ с AD и DNS, затем СУБД, потом уже vCenter (если он установлен в ВМ). Однако если у вас будут использоваться кластеры HA и/или DRS, настройка автостарта лишается смысла, так как она выполняется для ВМ конкретного сервера – а в кластерах ВМ не привязаны к конкретному серверу.

Прочие упомянутые мной в начале раздела настройки – сети, системы хранения – подробно разбираются в соответствующих разделах позже.

1.5.3. Работа через веб-интерфейс vSphere Web Client

В vSphere 5 немного поменялась концепция веб-интерфейса, по сравнению с предыдущими версиями vSphere. Самое заметное отличие – служба веб-интерфейса теперь устанавливается отдельно от vCenter (хотя и может быть установлена на тот же сервер). На самом ESXi, как и раньше, веб-интерфейса просто нет.

Функционал этого веб-интерфейса такой же, как и раньше, – операторский – с ВМ можно делать все или почти все, с остальными объектами – практически ничего.

Когда серверная часть уже установлена (об установке ниже), то браузером заходим на <https://<адрес сервера Web Client>:9443/vsphere-client>.

И после авторизации получаем доступ к интерфейсу. Если не получаем – значит у нас не установлен Adobe Flash, этот интерфейс, так же как и у некоторых других продуктов VMware, построен на его основе.

В веб-интерфейсе vSphere 5 нам доступны все манипуляции с виртуальными машинами, включая:

- создание как с нуля, так и клонированием или разворачиванием из шаблонов;

- ☐ доступ к консоли ВМ прямо из браузера;
- ☐ изменение состояния питания;
- ☐ удаление;
- ☐ изменение большинства настроек (включая изменение виртуального оборудования);
- ☐ запуск миграции;
- ☐ работа со снимками состояния;
- ☐ получение информации о ВМ, в том числе информации о задачах, событиях, производительности.

Для серверов ESXi доступны просмотр данных, включая производительность, события и задачи, а также создание пулов ресурсов, vApp, ввод в режим обслуживания.

Для хранилищ – переименование и просмотр информации.

Для виртуальных коммутаторов – просмотр информации.

Также доступен поиск объектов.

Обратите внимание: vCenter 5 может нормально управлять серверами ESXi предыдущих версий, поэтому даже если вы не обновили всю инфраструктуру – можно обновить только vCenter и использовать web-client.

Интересный факт – помните про правила лицензирования vSphere 5? Появившееся ограничение по памяти выданной ВМ – vRAM? Вот данные по текущему потреблению становятся доступны в клиенте vSphere после установки Web Client Server.

Мы увидим эти данные, пройдя **Home** ⇒ **Licensing** ⇒ вкладка **Reporting**.

Установка Web Client Server

В виртуальной машине с Linux-версией vCenter серверный компонент веб-интерфейса предустановлен, его потребуется только запустить. Для Windows-версии его следует установить отдельно.

Для начала установки нужно запустить autorun.exe из корня дистрибутива vCenter и выбрать компонент **Web client (Server)**. Сама установка тривиальна. После завершения установки следует браузером обратиться на интерфейс администратора (обязательно непосредственно с того сервера, где вы установили Web client (Server)) <https://<адрес сервера Web Client>:9443/admin-app> и выполнить единственно возможное действие по ссылке **Register vCenter Server** – зарегистрировать наш vCenter (или несколько). В поле **vSphere Web Client server name or IP** указываем имя или IP той машины, где установлен Web Client (Server), или то имя/адрес, на которое будут обращаться пользователи этого веб-интерфейса (предполагается, что вы настроите перенаправление с этого адреса на реальный адрес сервера со службой веб-интерфейса).

На момент написания была причина не устанавливать Web Client (Server) на vCenter – происходит конфликт со службой Profile Driven Storage. Возможно, этим можно пренебречь, если ваша лицензия не позволяет использовать данную функцию.

Эта проблема (я думаю, что она будет решена в следующих обновлениях) – единственная веская причина не устанавливать сервер веб-клиента на одном сервере с vCenter.

Менее веская в большинстве случаев, но тоже причина установить его на отдельную от vCenter машину – это снижение нагрузки на vCenter. Если вы затрудняетесь предсказать, будет ли нагрузка на Web Client (Server) значительной – думаю, будет оправданно установить на vCenter, помониторить комфортность работы и в случае недостаточного комфорта переустановить уже на отдельную ВМ.

1.5.4. vCenter Mobile Appliance, клиент для iPad, веб-интерфейс администратора

Упомяну об экспериментальном дополнительном продукте, который позволяет осуществлять некоторые административные операции с vSphere через браузер (с упором на браузеры самих мобильных устройств, включая простейшие из мобильных телефонов) и с iPad.

Продукт называется VMware vCenter Mobile Access (vCMA), это предустановленная виртуальная машина. Загрузить ее можно на сайте экспериментальных продуктов VMware – <http://labs.vmware.com>.

Ввод в эксплуатацию описывать здесь не буду (продукт экспериментальный и весьма прост в обращении), дам лишь ссылку на более подробную информацию со скриншотами в блоге – <http://link.vm4.ru/vcma>.

1.6. Основы работы из командной строки

Большинство операций с виртуальной инфраструктурой производятся из графического интерфейса клиента vSphere. Однако и командная строка может нам пригодиться:

- ☐ для некоторых операций, которые невозможны из графического интерфейса;
- ☐ для автоматизации действий с помощью сценариев;
- ☐ для диагностики и решения проблем.

У нас есть несколько способов для получения интерфейса командной строки к серверу ESXi, но у каждого есть свои особенности и сферы применения:

- ☐ локальная командная строка, доступная с локальной консоли или через iLO/IP KVM. Этот инструмент не самый удобный, зато он доступен с самой большей вероятностью – остальные варианты требуют как минимум работающего доступа по сети к ESXi. Для решения проблем данный вариант командной строки может быть незаменим;
- ☐ сессия SSH к ESXi. Вариант хорош тогда, когда такое взаимодействие с сервером вам привычно. Менее удобен, чем некоторые альтернативы, от-

сутствием централизации – с каждым сервером работа возможна только независимо;

- ❑ PowerCLI – надстройка над PowerShell, добавляющая в posh командлеты для управления vSphere. Все плюсы PowerShell. Централизация, очень богатые возможности по управлению и составлению отчетов для виртуальной инфраструктуры. Лично мне данный вариант кажется самым интересным, когда речь идет про автоматизацию;
- ❑ vSphere CLI – интерфейс удаленной командной строки. В отличие от ssh, работает через API-интерфейсы, внутри тоннеля ssl, что дает право VMware рекомендовать этот инструмент вместо ssh для инфраструктур с большими требованиями к безопасности. Позволяет использовать стандартные механизмы авторизации vSphere. Дает возможность взаимодействовать с несколькими серверами ESXi централизованно. Я бы сказал, что этот вариант из той же категории, что PowerShell/PowerCLI, но для *nix-инфраструктуры.

1.6.1. Локальная командная строка ESXi и доступ по SSH

VMware не рекомендует открывать доступ к командной строке и SSH для ESXi – из общих соображений безопасности. Однако если вы приняли решение пренебречь данной рекомендацией, сделать это несложно.

Для доступа в командную строку в локальной консоли ESXi эта возможность должна быть разрешена. В интерфейсе клиента vSphere сделать это можно, пройдя **Configuration ⇒ Security Profile ⇒ Properties** для служб ⇒ **ESXi Shell**.

Через локальное БИОС-подобное меню также можно открыть доступ к локальной командной строке, пройдите **Troubleshooting Options ⇒ Enable ESXi Shell**. После нажатия **Enter** название пункта меню должно поменяться на **Disable ESXi Shell** – это значит, что локальная командная строка включена, а этим пунктом ее можно отключить обратно.

Так или иначе разрешив доступ к локальной командной строке, нажмите **Alt+F1** и авторизуйтесь.

Появится приглашение к вводу команд:

```
~#
```

Вы вошли в локальную консоль.

Включение SSH выполняется точно так же (в БИОС-подобном меню или в пункте настроек **Security Profile**), только сейчас вас интересует пункт **SSH**.

Теперь вы можете подключаться по SSH.

В состав ESXi входит маленький дистрибутив Linux под названием Busybox. Основные команды Linux (табл. 1.1) в нем работают.

Подсмотреть прочие доступные для Busybox команды можно, выполнив

```
/usr/bin/busybox
```

Таблица 1.1. Список основных команд Linux

Команда	Описание
cd	Смена текущей директории
cp	Копирование файла cp [файл 1] [файл2]
find	Поиск файлов по критериям
ls	Список файлов и директорий в текущей или явно указанной директории. ls /vmfs/volumes/ ключи: -l подробная информация -a отображение скрытых файлов
mkdir	Создание директории
mv	Перемещение файла. Переименование файла mv [путь и имя файла] [путь куда перемещать]
ps	Информация о запущенных процессах ps -ef
rm	Удаление файлов
shutdown	Выключение или перезагрузка сервера shutdown now shutdown -r now
vi	Текстовый редактор
cat	Вывод содержимого файла на экран cat /etc/hosts
more	Вывод содержимого файла на экран, по странице за раз more /etc/hosts
useradd	Создание пользователя useradd <имя пользователя>
passwd	Задание пароля пользователю passwd <имя пользователя>

В состав ESXi входят некоторые из команд, специфичных для ESXi. По не совсем мне понятной причине, существуют несколько групп таких специальных команд с частично перекликающимся функционалом. Некоторые из них слабо документированы.

Основными командами я бы назвал:

- ☐ семейство команд esxcfg-. Их можно назвать классическими – они без особых изменений существуют с третьей версии ESX;
- ☐ оболочку esxcli. Она получила развитие именно в пятой версии ESXi, и имеет смысл именно этот инструмент использовать.

Список большинства (но не всех) «классических» команд вы можете получить, набрав в командной строке

```
esxcfg-
```

и два раза нажав **Tab**.

Однако информацию об этих командах и их соотношении с командами удаленной командной строки см. в разделе 1.5.4.

Команды `esxcli` приходят на смену всем прочим командам ESXi. Притом VMware предоставляет унифицированный доступ к командам этой структуры через любой интерфейс командной строки – и локальной, и удаленной, и PowerCLI.

К примеру, вы увидите одно и то же в следующих двух случаях:

1. Если в локальной командной строке выполните команду

```
esxcli network vswitch standard list
```

2. И если в PowerCLI-сессии выполните код (особое внимание – на последнюю строку)

```
$esx= Get-VMHost "esxi01.vm4ru.local"
$esxcli = Get-EsxCli -VMHost $esx
$esxcli.network.vswitch.standard.list()
```

Структура команд `esxcli` довольно проста и легко обнаруживается – выполнив `esxcli`, вы получаете список возможных вариантов второго уровня. Например, вас заинтересовала команда второго уровня `network`. Теперь, выполнив команду `esxcli network`, вы получаете список возможных команд третьего уровня, и т. д. Самыми последними идут операторы действий – `get`, `set`, `list`, `add`, `remove` и др.

Вот иллюстрация. Итерация один:

```
~#esxcli
Available Namespaces:
  esxcli           Commands that operate on the esxcli system itself allowing
                   users to get additional information.
  fcoe             VMware FCoE commands.
  hardware         VMKernel hardware properties and commands for configuring
                   hardware.
  iscsi            VMware iSCSI commands.
  network          Operations that pertain to the maintenance of networking on an
                   ESXi host. This includes a wide variety of commands to
                   manipulate virtual networking components (vswitch,
                   portgroup, etc) as well as local host IP, DNS and general
                   host networking settings.
  software         Manage the ESXi software image and packages
  storage          VMware storage commands.
  system           VMKernel system properties and commands for configuring
                   properties of the kernel core system.
  vm               A small number of operations that allow a user to Control
                   Virtual Machine operations.
```

Итерация два:

```
~ # esxcli network
Available Namespaces:
  fence           Commands to list fence information
  firewall         A set of commands for firewall related operations
  ip               Operations that can be performed on vmknics
```

vswitch	Commands to list and manipulate Virtual Switches on an ESX host.
nic	Operations having to do with the configuration of Network Interface Card and getting and updating the NIC settings.

Итерация шесть:

```
~ # esxcli network ip interface ipv4 get
```

Name	IPv4 Address	IPv4 Netmask	IPv4 Broadcast	Address Type	DHCP	DNS
vmk0	192.168.22.201	255.255.255.0	192.168.22.255	STATIC	false	
vmk1	2.2.2.2	255.255.255.0	2.2.2.255	STATIC	false	

Здесь я хочу сделать акцент на следующем – выполнив команду `esxcli`, на выход мы получаем доступное пространство имен. Выбрав среди них интересное нас, например `network`, мы выполняем команду `esxcli network` – и получаем пространство имен следующего уровня, где выбираем требуемое. Двигаясь таким образом, мы, даже не зная нужной команды заранее, сможем ее подобрать. К примеру, команда `esxcli network ip interface ipv4 get` выводит список интерфейсов гипервизора и их сетевые настройки. Если на конце команды заменить `get` на `set` – настройки выбранного интерфейса можно будет поменять.

Команда `esxcli esxcli command list` выведет на экран список всех возможных команд `esxcli`.

Ключ `--help`, добавленный в конце любой команды `esxcli`, отобразит справку по этой команде, возможным пространствам имен и операциям на ее уровне.

Если вы только начинаете использовать командную строку – есть резон привыкать использовать именно `esxcli`. Если вы уже имеете опыт работы с «классическими» командами – они применимы в полный рост.

Чуть ниже я приведу основные «классические» команды. Обратите внимание, что у VMware доступна отличная документация по основным интерфейсам командной строки, в частности в документе «vSphere Command-Line Interface Concepts and Examples» в разделе «List of Available Commands» вы обнаружите соответствие между `esxcfg-` и `esxcli-` командами.

Но у `esxcli` есть интересное преимущество – этот инструмент является расширяемым. Например, если установить продукт `vCloud Director` (он предоставляет портал самообслуживания для создания и взаимодействия с ВМ на `vSphere`), то в списке `namespaces` для `esxcli` появится пространство имен `vcloud`. Если использовать дистрибутив `ESXi` от HP, то появятся пространства имен `hp` и `hpbootcfg`.

Напоследок перечислю слабо- или недокументированные команды `ESXi 5`, просто для справки:

❑ `vim-cmd`. Например:

```
vim-cmd vmvc/power.getstate <ID виртуальной машины>
```

вы узнаете статус питания виртуальной машины с указанным ID.

Увидеть список ВМ и их ID вы можете при помощи команды

```
vim-cmd vmvc/getallvms
```

Начать имеет смысл с выполнения этой команды без параметров;

- ❑ `vsish` – Vmkernel System Info Shell; выполните эту команду, затем выполните команду `help` и `ls` для получения базовой информации. Например, при помощи данного инструмента можно вызвать пурпурный экран смерти на сервере ESXi (может пригодиться в тестовых целях)

```
vsish -e set /reliability/crashMe/Panic
```

Здесь я их привожу, чтобы вы не удивлялись, найдя в Интернете какие-то конкретные инструкции с их участием. Есть маленький процент настроек, который можно выполнить только ими. Кроме того, если вам требуется выполнить что-то специфическое, что вы не смогли выполнить стандартными командами, – можно попробовать почитать справку и вывод этих команд, может быть, удастся обнаружить искомое.

1.6.2. Microsoft PowerShell + VMware PowerCLI

Мой личный фаворит в области автоматизации задач для vSphere – это PowerCLI. Мне он нравится сразу по нескольким причинам:

- ❑ универсальность. Понимая базовые принципы и конструкции PowerShell, я могу их применять как для управления виртуальной инфраструктурой, так и для управления многими другими продуктами (в первую очередь это продукты Microsoft, но не только);
- ❑ понятность и читаемость самого языка. Критерий спорный, во многом эти факторы – дело привычки, но все же я считаю, что начать овладевать PowerCLI с нуля весьма просто;
- ❑ заметная популярность – для большого процента задач возможно обнаружение готового или близкого к тому решения.

Для овладения этим инструментом следует сконцентрироваться на изучении PowerShell, а не PowerCLI, так как последний всего лишь добавляет специальные команды в язык PowerShell. А первичнее здесь понимание общих принципов, знание конструкций языка и т. п. По PowerShell существует большое количество статей, блогов и даже книг – соревноваться с ними или дублировать их смысла нет никакого. Поэтому в этой книге я про PowerShell в большей степени упомяну, чем приведу подробные инструкции.

Однако, во-первых, иногда я буду приводить примеры PowerCLI-кода для решения тех или иных задач; а во-вторых, дам ссылку на большой пост в своем блоге, который так и называется – «PowerShell + PowerCLI, с чего начать». Ссылка: <http://link.vm4.ru/powercli>. Эта пополняемая статья может оказаться полезной для начинающих.

В ней я постарался дать подробную инструкцию, с чего начать, описание основных возможностей языка и ссылки на вспомогательные инструменты и ресурсы. Настоятельно рекомендую прочесть ее и начать использовать этот язык в работе – разумеется, если размер вашей инфраструктуры заслуживает автоматизации.

Но на всякий случай я приведу минималистскую инструкцию, достаточную для выполнения того PowerCLI-кода, что я привожу в книге.

Настройка PowerCLI

Вам потребуется:

- 1) установить PowerShell, если его еще нет;
- 2) установить PowerCLI;
- 3) подключиться к vSphere.

На Windows 7 и Windows 2008 скриптовый язык PowerShell поставляется предустановленным. Для Windows XP и Windows 2003 его требуется установить отдельно. Загрузить соответствующий дистрибутив под названием «Windows Management Framework» можно по ссылке <http://www.microsoft.com/powershell>.

Я считаю удобным использовать графическую среду разработки PowerShell ISE. Она автоматически установится на Windows XP/2003, присутствует по умолчанию в Windows 7, и ее требуется доустановить для Windows 2008 (это стандартный компонент, просто он не устанавливается по умолчанию). Предположим, вы следуете моему совету и в дальнейшем будете запускать сценарии из оболочки PowerShell ISE.

Перед установкой PowerCLI требуется понизить уровень безопасности PowerShell. Запустите PowerShell ISE и PowerCLI и в обеих консолях выполните команду

```
Set-ExecutionPolicy RemoteSigned
```

Теперь можно устанавливать PowerCLI.

Загрузить дистрибутив можно по ссылке <http://vmware.com/go/powercli>. Установка проста и вопросов не вызывает.

Теперь вы можете запустить следующие ярлыки:

- ☐ PowerShell;
- ☐ PowerShell ISE;
- ☐ PowerCLI.

Последний запускает ту же командную строку, что и первый, но с другой цветовой схемой и подгружая командлеты от VMware. А вот запуская оригинальный PowerShell или PowerShell ISE, мы командлетов для работы с vSphere не обнаружим. Допустим, как мы условились ранее, вы хотите использовать PowerShell ISE.

Запустите эту оболочку. Выполните команду

```
add-psnapin VMware*
```

Теперь командлеты VMware подгружены, и их можно использовать. Но подгрузка дополнительных модулей происходит только на текущий сеанс, при следующем открытии PowerShell или PowerShell ISE вам придется повторить эту команду. Или добавить ее в автозагрузку, см. информацию о профиле PowerShell.

Самый первый командлет, с которого следует начинать всегда, – это командлет для подключения к vCenter (или к отдельному ESXi):

```
Connect-VIServer «имя или IP сервера vCenter»
```

Если учетная запись, под которой вы работаете, имеет права на подключение к vCenter – то подключение будет установлено. Иначе вы увидите стандартный запрос учетной записи, и подключение будет установлено после ее ввода.

Теперь команды PowerCLI будут работать с серверами и виртуальными машинами того vCenter, к которому вы подключились. Для проверки выполните команду, например

```
Get-VM
```

Она выведет на экран список всех виртуальных машин.

Точно так же после этих манипуляций будет работать код, приведенный в этой книге.

Суперкороткая инструкция на этом заканчивается, за дополнительной информацией приглашаю вас сюда – <http://link.vm4.ru/powercli>.

1.6.3. vSphere CLI, работа с vMA

vSphere CLI – это инструмент, который позволяет централизованно управлять из командной строки серверами ESXi. Более того, некоторые команды можно и удобно направлять на vCenter Server.

vSphere CLI представляют собой набор сценариев, которые выполняются на том компьютере, где vSphere CLI установлен. При выполнении сценариев обращается к API на указанном сервере ESXi или сервере vCenter и выполняет свою работу на этом сервере.

Однако на ESXi с бесплатной лицензией vSphere CLI работают в режиме только чтения (read-only). Это означает, что вы можете использовать ее для просмотра каких-то свойств и значений, но не сможете их изменять.

vSphere CLI поставляются в трех вариантах:

- ☐ дистрибутив под Windows;
- ☐ дистрибутив под Linux;
- ☐ в составе vSphere Management Assistant, vMA.

Здесь я подробнее остановлюсь на последнем варианте.

vSphere Management Assistant (vMA) – это виртуальная машина с предустановленной ОС и набором продуктов. В ней установлены vSphere CLI, которые позволяют централизованно выполнять команды командной строки на нескольких серверах ESXi.

Начать пользоваться vMA очень просто:

1. Загружаем сам продукт с сайта VMware (<http://vmware.com/go/vma>). Распаковываем архив.



2. Запускаем клиент vSphere и импортируем виртуальную машину vMA в нашу виртуальную инфраструктуру. Это делается из меню **File ⇒ Deploy OVF Template**.
3. На последнем шаге мастера импорта мы можем указать IP-адрес для этой VM.
4. Включаем импортированную VM. Открываем консоль к ней. При первом включении от нас спросят настройки IP и пароль пользователя vi-admin.

Теперь выполним начальную настройку в локальной консоли vMA, или подключившись к ней по SSH. Авторизуйтесь пользователем vi-admin. С помощью команды

```
vifp addserver <servername>
```

добавьте свои сервера ESXi и vCenter. От вас попросят указать пароль пользователя (root для ESXi или учетную запись Windows, имеющую административные права для vCenter). В дальнейшем вводить учетные данные при запуске сценариев не придется.

Проверить список зарегистрированных серверов можно командой

```
vifp listservers
```

Затем самым удобным, на мой взгляд, будет следующее: укажите целевой сервер командой

```
vifptarget -s <servername>
```

Теперь любая команда vSphere CLI будет выполнена в отношении указанного сервера.

Проверьте работоспособность сделанных настроек:

```
vicfg-nics --list
```

Эта команда должна отобразить список физических сетевых контроллеров сервера ESXi.

Выполняя команду

```
vifptarget -s <servername>
```

вы можете менять целевые сервера. Чтобы обнулить указание целевого сервера и опять работать только в командной строке vMa, выполните команду

```
bash
```

Это не единственный вариант настройки аутентификации на серверах ESXi при выполнении команд vSphere CLI, в первую очередь обратите внимание на воз-

возможность ввести vMA в домен Active Directory и использовать его возможности для аутентификации в vSphere. Но остальные кажутся мне более специфичными и/или менее удобными в повседневной работе, так что приводить здесь их не буду.

Обратите внимание. Если вам потребуется изменить сетевые настройки для vMA, то проще всего это осуществить, выполнив в локальной консоли vMA команду `sudo/opt/vmware/vma/bin/vmware-vma-netconf.pl`.

За дополнительной информацией обращайтесь в документ «vSphere Management Assistant Guide», доступный на <http://vmware.com/go/vma>.

1.6.4. Полезные команды

В тексте книги я иногда буду приводить какие-то команды для выполнения тех или иных действий. Эти команды или относятся к PowerCLI, или их можно запустить в локальной командной строке/SSH/vSphere CLI.

Еще раз напомним: в пятой версии ESXi VMware делает акцент на команде `esxcli` – и предлагает выполнять все или большинство настроек ESXi из командной строки с ее помощью. А упоминаемые здесь команды можно назвать «классическими», унаследованными из предыдущих версий. Здесь я привожу их для справки.

В табл. 1.2 я перечислил многие из полезных «классических» команд.

Таблица 1.2. Список полезных команд vSphere CLI и локальной командной строки

Команда vSphere CLI	Аналог в ESXi	Описание
<code>resxtop</code>	<code>esxtop</code>	Мониторинг системных ресурсов
<code>svmotion</code>		Запуск Storage VMotion
<code>vicfg-advcfg</code>	<code>esxcfg-advcfg</code>	Изменение расширенных настроек
<code>vicfg-cfgbackup</code>		Резервная копия настроек ESXi
<code>vicfg-dns</code>		Настройка DNS
<code>vicfg-dumppart</code>	<code>esxcfg-dumppart</code>	Доступ к диагностическим данным
<code>vicfg-iscsi</code>	<code>esxcfg-hwiscsi</code> и <code>esxcfg-swiscsi</code>	Настройка iSCSI (программного и аппаратного)
<code>vicfg-module</code>	<code>esxcfg-module</code>	Управление модулями VMkernel
<code>vicfg-mpath</code>	<code>esxcfg-mpath</code>	Вывод информации о путях к LUN
<code>vicfg-nas</code>	<code>esxcfg-nas</code>	Настройка доступа к NAS
<code>vicfg-nics</code>	<code>esxcfg-nics</code>	Настройка физических NIC
<code>vicfg-ntp</code>		Настройки сервера NTP
<code>vicfg-rescan</code>	<code>esxcfg-rescan</code>	Сканирование СХД, обнаружение новых LUN и разделов VMFS
<code>vicfg-route</code>	<code>esxcfg-route</code>	Настройка маршрутизации
<code>vicfg-scsidevs</code>	<code>esxcfg-scsidevs</code>	Информация об устройствах хранения
<code>vicfg-snmp</code>		Управление агентом SNMP
<code>vicfg-vmknic</code>	<code>esxcfg-vmknic</code>	Управление интерфейсами VMkernel
<code>vicfg-volume</code>	<code>esxcfg-volume</code>	Перемонтирование разделов VMFS

Таблица 1.2. Список полезных команд vSphere CLI и локальной командной строки (окончание)

Команда vSphere CLI	Аналог в ESXi	Описание
vicfg-vswitch	esxcfg-vswitch	Управление виртуальными коммутаторами
vihostupdate	esxupdate	Установка обновлений
vmkfstools	vmkfstools	Управление разделами. VMFS Управление файлами vmdk

Несложно заметить, что большинство локальных команд начинаются на `esxcfg-`, аналогичные им команды vSphere CLI – на `vicfg-`. Однако команды `esxcfg-` в vSphere CLI также доступны, и запускают они соответствующую команду `vicfg-`. Так сделано для упрощения совместимости ранее созданных сценариев.

Синтаксис команд в локальной командной строке и в vSphere CLI практически идентичен. Многие команды отличаются только тем, что при запуске их из vSphere CLI необходимо указать целевой сервер (ESXi или сервер vCenter). Имя сервера указывается после ключа `-server` (если выполнено указание целевого сервера на сеанс командой `vihosttarget -s <servername>`, то явно указывать сервер в самой команде необходимости нет).

Для получения справки большинство команд достаточно запустить без параметров. Правда, как правило, объем справочной информации значительно превышает один экран. Так что для ее просмотра вам пригодится команда «more» из табл. 1.1. В случае же если вы предпочитаете читать красиво форматированный текст – найдите подробную справку по синтаксису команд в документации VMware.

1.6.5. Полезные сторонние утилиты

При работе с ESXi нам могут потребоваться некоторые действия, невозможные при помощи клиента vSphere. Это такие действия, как:

- ☐ командная строка, в том числе для изменения конфигурационных файлов и просмотра журналов;
- ☐ графический файловый менеджер для работы с файлами ESXi, в первую очередь изменения конфигурационных файлов и просмотра журналов;
- ☐ графический файловый менеджер для работы с файлами виртуальных машин. Копирование, перемещение, изменение, загрузка на ESXi, выгрузка с ESXi.

Перечислю основные бесплатные утилиты, которые могут вам помочь в этих операциях.

Командная строка, SSH

Для работы в командной строке могу посоветовать две утилиты – клиент SSH под названием PuTTY и менеджер сессий mRemote.

PuTTY можно загрузить с веб-сайта по адресу <http://www.putty.org>. Запустите программу, укажите адрес вашего сервера ESXi (если вы включили сервер SSH на нем) или vMA (рис. 1.14).

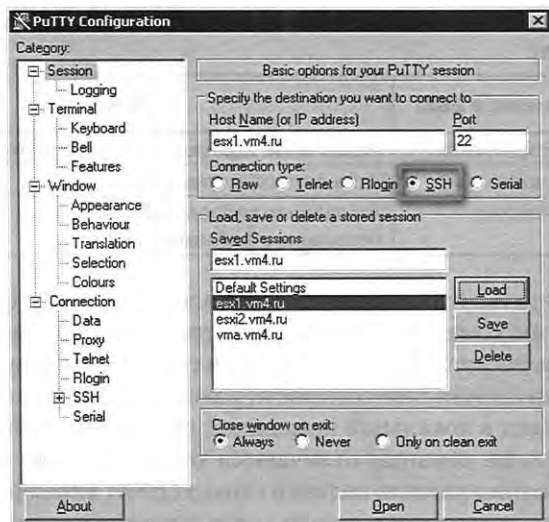


Рис. 1.14. Подключение с помощью PuTTY

Добавить удобства в работе с несколькими серверами вам поможет утилита mRemote (рис. 1.15).

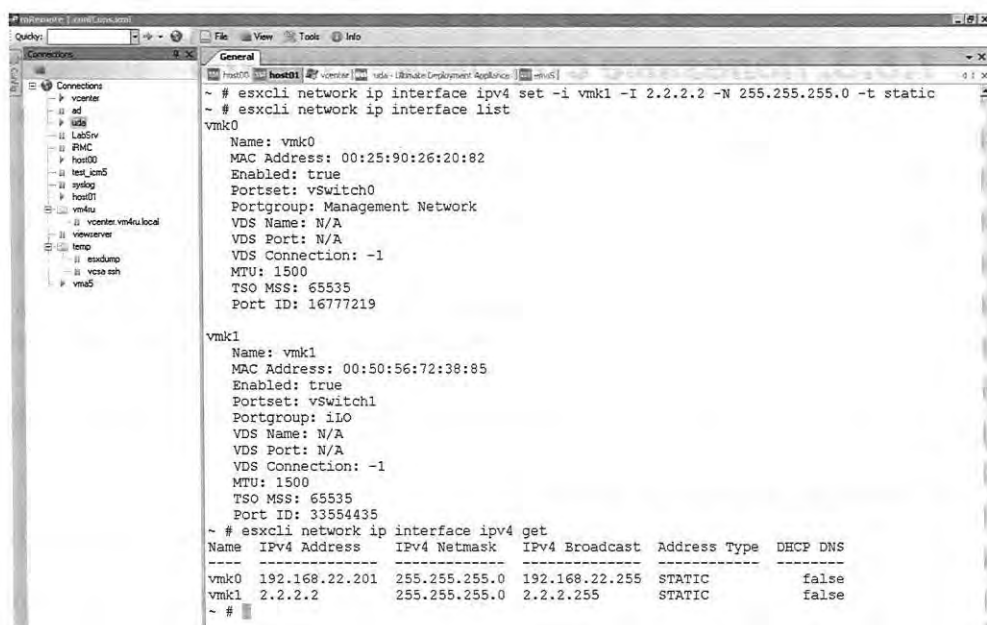


Рис. 1.15. Окно PuTTY в менеджере сессий mRemote

Она позволяет сохранять параметры ssh-подключений, притом запоминая учетную запись, а также поддерживает другие полезные протоколы, кроме ssh, в первую очередь rdp, http. Сайт этой утилиты – <http://www.mremote.org>.

Файловый менеджер

На случай, когда вам необходимы манипуляции с файлами виртуальных машин, самым удобным средством я считаю Veeam FastSCP (<http://www.veeam.com/ru/product.html>). Это специализированный файловый менеджер именно для vSphere. С его помощью вы легко и удобно получите доступ к файлам виртуальных машин. Вернее, к любым файлам на хранилищах VMFS и NFS-хранилищах ваших серверов (рис. 1.16).

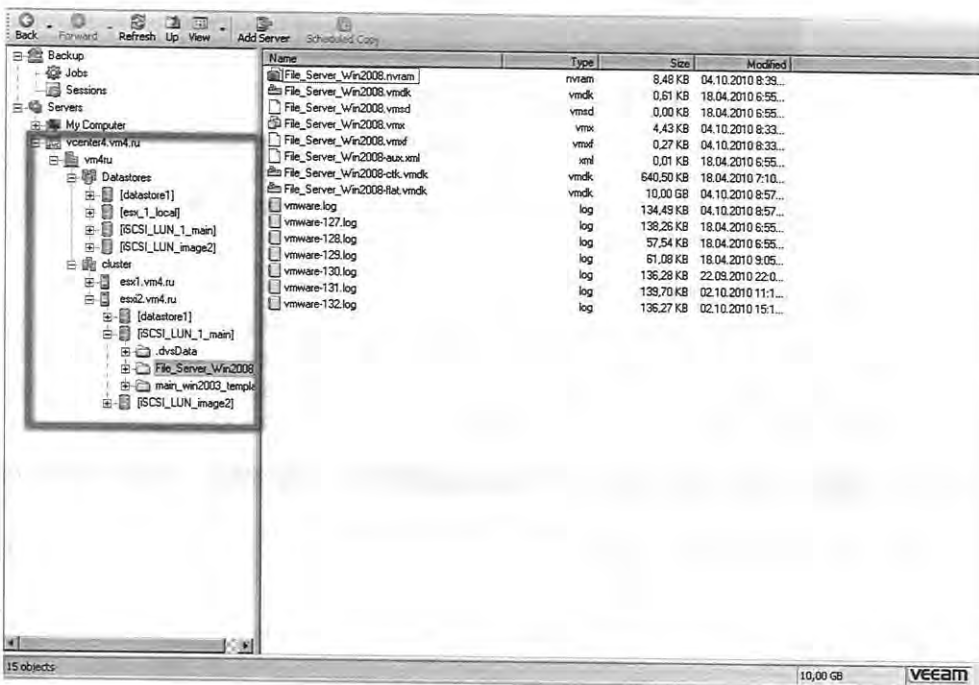


Рис. 1.16. Окно FastSCP

С помощью этой утилиты легко организовать копирование файлов между хранилищами разных серверов или между серверами ESXi и Windows (в первую очередь имеется в виду резервное копирование).

Еще одна утилита, о которой хочу здесь упомянуть, – файловый менеджер WinSCP (<http://winscp.net>). С его помощью можно обратиться к ESXi – правда, лишь при условии включения сервера SSH (рис. 1.17).

После подключения этой утилитой вы увидите двухпанельный файловый менеджер. В левом окне всегда система Windows, с которой вы подключились, а в правом – ESXi, к которому вы подключились. Примечательно, что с помощью

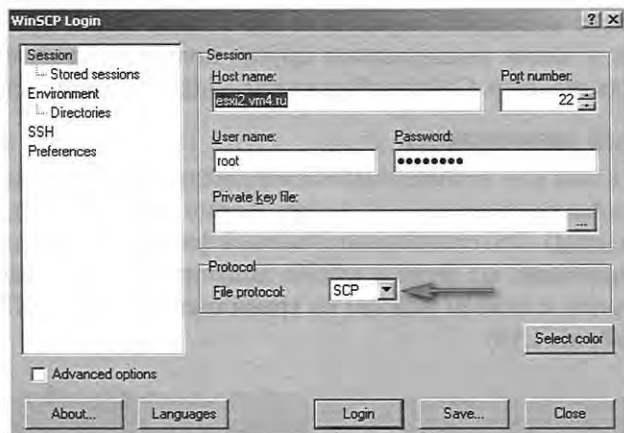


Рис. 1.17. Настройки подключения WinSCP

этой утилиты вы получаете доступ к файловой системе Linux из состава ESXi. Это означает, что, зайдя в каталог `/etc`, вы найдете большинство конфигурационных файлов. В каталоге `/var/logs` вам доступны файлы журналов. В каталог `/vmsfs/` volumes подмонтируются хранилища.

Вспомогательные утилиты

Еще одна утилита, о которой хотелось бы тут упомянуть, – утилита RVtools. Подключившись с ее помощью к серверу vCenter, вы сможете в удобном виде получить полезные данные об инфраструктуре. Особо интересной может оказаться вкладка **vHealth** с информацией о потенциальных проблемах (рис. 1.18).

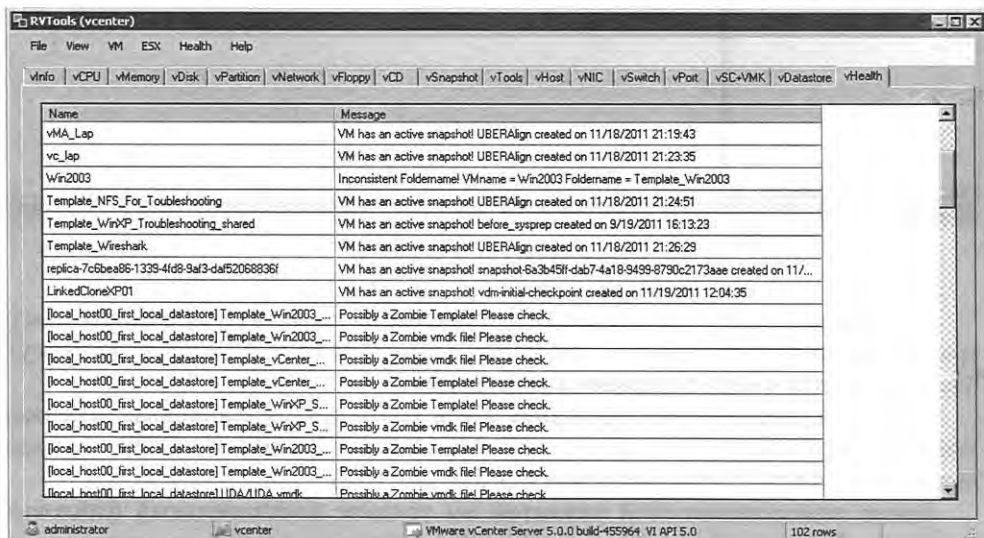


Рис. 1.18. Утилита RVTools

Утилита доступна по адресу <http://www.robware.net>.

Именно на этой вкладке вы сможете обнаружить такие проблемы, которые затруднительно обнаружить штатными средствами.

1.7. Сайзинг и планирование

В этом разделе мне бы хотелось обозначить некоторые вопросы выбора конфигурации (сайзинга) аппаратного обеспечения под vSphere. Вопрос вида «какой сервер выбрать под любимый гипервизор» – один из популярных и самых раздражающих, потому что такая формулировка вопроса некорректна.

То, что написано далее, ни в коей мере не «Полное и абсолютное руководство по подбору оборудования всегда», нет. Это немного теории про то, что такое производительность, – как показывает мой шестилетний опыт ведения ИТ-курсов, даже у самых опытных людей бывают пробелы в обыденных для кого-то другого вещах. Это небольшие нюансы выбора оборудования, связанные именно с виртуализацией. Это вопросы, на которые нельзя давать ответ абстрактно, но о которых стоит задуматься, имея на руках данные конкретного случая.

Итак, нам нужно оборудование под ESXi. Что дальше?

Самое главное для понимания, пусть и очевидное до невозможности, – это следующее соображение:

Количество ресурсов, которыми должен обладать сервер под ESXi, зависит от суммы требований для выполнения всех задач, которые мы собираемся выполнять в VM на этом сервере. И еще от того, будем ли мы использовать некоторые функции vSphere (в первую очередь HA/FT), так как эти функции требуют ресурсов сверх необходимых для работы VM.

Поговорим про все подсистемы – процессор, память, диски, сеть. Самая большая и сложная тема – это дисковая подсистема. Должна ли она быть представлена системой хранения данных (СХД) или обойдемся локальными дисками сервера – это вопрос относительно простой. Есть бюджет на СХД – она нам нужна ☺. На нет и суда нет. Какую именно СХД использовать – будет ли это система с подключением по оптике, Fibre Channel SAN? Или достаточно будет 1 Гбит подключения Ethernet и это будет iSCSI SAN? Или это будет iSCSI, но с 10 Гбит подключением? Или NFS? Вариантов масса. Какие диски там будут стоять? Или стоит задуматься не о дисках, а о SSD-накопителях?

В любом случае, вам необходимо опираться на цифры. Цифры по нагрузке существующих систем, которые вы планируете переносить в виртуальную среду. Или цифры по планируемой нагрузке приложений, которые вы собираетесь добавлять в виртуальную среду.

Для уже существующей инфраструктуры статистику использования той или иной подсистемы имеет смысл получать с помощью соответствующих средств мониторинга. Есть вероятность, что у вас в компании уже может быть развернуто средство мониторинга, например решение от Майкрософт – System Center Operations Manager (OpsMgr, SCOM). И тогда вы можете обратиться к собранной с его помощью информации о нагрузке на ваши сервера – чтобы понять,

сколько ресурсов будут потреблять виртуальные машины, в которые вы превратите ваши сервера.

Ну а наиболее правильно и удобно использовать специализированное средство анализа инфраструктуры для ее виртуализации – VMware Capacity Planner. Это средство позволит собрать данные по потребляемым ресурсам и проанализировать их, в значительной степени автоматически, именно в контексте виртуализации vSphere. Услуги по обследованию с его помощью оказывают компании-партнеры VMware.

Для планируемых систем цифры по предполагаемой нагрузке берутся из тех же источников, что и при сайзинге физических серверов, – из соответствующей документации, нагрузочных тестов и опыта.

1.7.1. Процессор

Мне видятся два момента, связанных с выбором процессоров в сервер под ESXi.

Первый – какой производительностью они должны обладать, и второй – накладываются ли на процессоры какие-то условия с точки зрения некоторых функций vSphere.

Про производительность поговорим чуть позже, сейчас про функционал.

Выбор процессоров с точки зрения функционала

Процессор – это единственный компонент сервера, который не виртуализируется. Это означает, что гостевая ОС видит тактовую частоту и, самое главное, наборы инструкций (типа SSE4 и т. п.) физического процессора. Если ВМ просто и без затей работает на каком-то ESX-сервере, то от такого положения вещей нам ни холодно, ни жарко. Проблемы начинаются, если мы хотим осуществить живую миграцию (ту, что у VMware называется vMotion) этой ВМ на другой ESXi.

Вывод: если мы предполагаем использование vMotion, то ЦП на этих серверах должны быть одинаковыми. Раскроем нюансы.

«Одинаковость» здесь должна быть по функциям этих процессоров. Например, набор инструкций SSE 4.2 – если их поддерживает ЦП сервера, то наличие этих инструкций увидит и гостевая ОС. Если на ЦП другого сервера этих инструкций не будет, а мы мигрируем включенную ВМ на этот сервер, то гостевая ОС чрезвычайно удивится невозможности использовать то, что только что было доступно. И может умереть, сама ОС или приложение. Чтобы такого не было, vCenter в принципе не даст нам мигрировать ВМ между серверами с разными ЦП.

Резюме: не важны тактовая частота, размер кеш-памяти и количество ядер. Важны поддерживаемые функции, то есть поколение ЦП, версия прошивки и (важно!) настройки BIOS. Некоторые функции могут включаться/выключаться из BIOS, поэтому вроде бы одинаковые ЦП могут оказаться несовместимыми с точки зрения vMotion. Кстати говоря, не всегда легко найти отличия в настройке. В моей практике в таких ситуациях помогал сброс настроек BIOS на значения по умолчанию.

Далее. Забудьте предыдущий абзац ☺.

Еще в ESXi 3.5 Update 2 и тем более в версии 5 VMware реализовала и предлагает нам функцию EVC – Enhanced vMotion Compatibility. Суть этой функции – в том, что мы можем «привести к единому знаменателю» разные ЦП на группе серверов. Например, у нас есть сервера с ЦП поновее – поддерживающие наборы инструкций до SSE 4.2. И есть сервера с ЦП постарше, поддерживающие набор инструкций SSE 4. Если мы объединим две эти группы серверов и включим для них EVC, то более новые ЦП выключат у себя SSE 4.2. И все эти сервера (их ЦП) станут совместимы для vMotion.

Однако для работы EVC требуется, чтобы все процессоры поддерживали технологию «AMD-V Extended Migration» или «Intel VT FlexMigration». Это:

- ☐ для AMD: процессоры Opteron™ начиная с моделей Rev. E/F;
- ☐ для Intel: начиная с Core™ 2 (Merom).

Подробную информацию о моделях можно почерпнуть или в списке совместимости, или в статье базы знаний <http://kb.vmware.com/kb/1003212>.

Резюме: если мы приобретаем процессоры с поддержкой этой технологии, то мы можем не волноваться – совместимость для vMotion будет обеспечена. Ценой этому будет недоступность части новых функций в новых ЦП. Как правило, это более чем допустимая жертва.

Надо понимать, что vMotion между серверами с процессорами разных производителей (с AMD на Intel и обратно) невозможен ни при каких условиях.

Попытаюсь резюмировать.

1. Если вы выбираете процессоры в сервер под ESX, предполагаете использование живой миграции aka vMotion и
 - это будут первые ESXi сервера вашей инфраструктуры – то выбирать можно достаточно свободно (см. еще соображения по производительности). Если в среднесрочной перспективе планируется докупать еще сервера, то лучше сейчас выбирать процессоры поновее (из последней группы совместимости EVC), чтобы в будущих серверах еще были доступны к заказу такие же (или представители той же группы);
 - у вас уже есть несколько ESXi-серверов, к ним надо докупить еще. Процессоры с поддержкой EVC можно использовать в одном «кластере EVC» с процессорами, не поддерживающими Extended Migration/FlexMigration. В таком случае процессоры без поддержки EVC должны быть одинаковы между собой.
2. Если vMotion не предполагается к использованию, то обращать внимание нужно лишь на соображения производительности. Однако в силу того, что на момент написания у VMware остались только две редакции vSphere, не включающие vMotion, – Free и Essentials, лучше исходить из возможного включения vMotion в будущем.

Если процессоры у нас не поддерживают EVC, то совместимость ЦП можно обеспечить на уровне настроек ВМ. По эффекту это похоже на EVC, но выполняется вручную и на уровне отдельной ВМ, а не группы серверов. Правда, эта возможность является неподдерживаемой. Вас интересует настройка **Options** ⇒ **CPUID Mask** ⇒ **Advanced**. Наконец, можно просто отключить проверку совместимости.

тимости ЦП на уровне настроек vCenter. Подробности см. в соответствующем разделе – про vMotion.

Вторая после vMotion функция, которая обладает собственными требованиями к процессорам, – это VMware Fault Tolerance, FT.

Для серверов, на которых работает VMware Fault Tolerance, должны выполняться все условия для vMotion. Плюс к тому тактовые частоты процессоров на разных серверах не должны отличаться более чем на 300 МГц. Наконец, эта функция работает только на процессорах из списка:

- ☐ Intel 31xx, 33xx, 34xx, 35xx, 36xx, 52xx, 54xx, 55xx, 56xx, 65xx, 74xx, 75xx;
- ☐ AMD 13xx, 23xx, 41xx, 61xx, 83xx.

Актуальную информацию о моделях процессоров для FT ищите в статье базы знаний VMware № 1008027 (<http://kb.vmware.com/kb/1008027>).

Выбор процессоров с точки зрения производительности

На производительность процессорной подсистемы оказывают влияние:

- 1) количество ядер (ну и самих процессоров, само собой);
- 2) их тактовая частота;
- 3) размер кеш-памяти;
- 4) поддерживаемые инструкции.

В идеале мы покупаем самый новый процессор (с максимумом поддерживаемых инструкций), с максимальным размером кеша, максимальными тактовой частотой и количеством ядер. В реальности же это не всегда возможно, да и не всегда оправдано.

Самое эффективное в общем случае – использовать меньше процессорных сокетов и больше ядер из тех соображений, что ESXi лицензируется на процессоры. Получим более эффективное соотношение цены и производительности.

Еще один нюанс: один виртуальный процессор равен одному ядру физического сервера как максимум. Нельзя взять два физических ядра и объединить в одно виртуальное, можно только «поделить» одно физическое на несколько виртуальных. Поэтому чем большей частотой будут обладать выбранные вами процессоры, тем большей максимальной производительностью будет обладать один виртуальный процессор.

Остальное, по моему мнению, менее критично.

Что такое и зачем надо Intel-VT/AMD-V. Аппаратная поддержка виртуализации

Процессор исполняет инструкции. Эти инструкции выполняются с тем или иным приоритетом, который традиционно называется «кольцом» (ring). Кольцо 0 имеет наибольший приоритет, и именно в этом кольце работает ядро ОС, управляющее всеми ресурсами.

Но. В случае использования виртуализации ОС несколько, и ресурсами сервера должен управлять гипервизор, распределяя эти ресурсы между ОС в ВМ. Выходов несколько:



- в ESXi реализована технология под названием «Binary translation» – программный механизм «вытеснения» ядра гостевой ОС из нулевого кольца. ESXi на лету подменяет инструкции процессора таким образом, что работают они на уровне выше нулевого кольца, а ядру гостевой операционной системы кажется, что оно на самом деле в нулевом кольце. Таким образом, ESXi может работать на процессорах без аппаратной поддержки виртуализации. Гипервизор работает в кольце 0. Это самый старый механизм, который в наше время считается унаследованным. Он обладает самыми высокими накладными расходами и вообще не работает для 64-битных гостевых ОС;
- паравиртуализация ОС. Ядро гостевой ОС может быть изменено для корректной работы в ВМ – тогда гипервизору не требуется принудительно «выгонять» ядро из нулевого кольца. В силу необходимости изменения кода ОС на уровне ядра, притом изменения разного для разных гипервизоров, паравиртуализация не стала массовой. Тем не менее использование паравиртуализованных ОС в ВМ на современных гипервизорах позволяет снизить накладные расходы на виртуализацию. Известные мне источники говорят о примерно 10%-ной разнице, по сравнению с непаравиртуализованными ОС. Для запуска паравиртуализованных гостевых ОС нет необходимости в аппаратной поддержке виртуализации. Паравиртуализация де-факто неактуальна для ESXi, поэтому здесь упоминается для полноты картины;
- та самая аппаратная поддержка виртуализации. Она реализуется с помощью технологий Intel VT (Virtualization Technology), или AMD-V. Если процессор обладает этой возможностью, то это означает, что он предоставляет для гипервизора специфический приоритет. Этот приоритет обычно называют кольцо –1 (минус один). Специфичность проявляется в том, что исполняемые в этом кольце инструкции, с одной стороны, еще приоритетнее, чем инструкции кольца нулевого; а с другой – в минус первом кольце могут исполняться не все инструкции. Впрочем, все, необходимые гипервизору, в наличии. Таким образом, если процессоры сервера обладают аппаратной поддержкой виртуализации, то гипервизор на таком сервере может запускать любые гостевые операционные системы без необходимости их модификации. Этот режим является штатным для ESXi.

Уже несколько лет все выпускаемые серверные (да и не только серверные) процессоры аппаратно поддерживают виртуализацию. Если вы работаете с не очень современным сервером, обратите внимание на этот аспект отдельно. Дело в том, что аппаратная поддержка виртуализации должна быть реализована не только в процессоре, но и в материнской плате (возможно, потребуется обновление BIOS).

Однако возможность реализовать виртуальную инфраструктуру даже на серверах без аппаратной поддержки виртуализации является архиполезной, если подобная задача перед вами все-таки встала.

1.7.2. Память

К ОЗУ требований особо нет. Ее просто должно быть достаточно. Грубая оценка достаточного объема:

1. Берем информацию о том, сколько памяти необходимо каждой ВМ. Суммируем. Округляем до удобного для установки в сервер объема.
2. Хотя при сайзинге это учитывать не следует, но часть памяти окажется свободной. В этом помогут механизмы ESXi по работе с памятью. Учитывать же при планировании эти механизмы не рекомендуется по причине того, что в общем случае эффект экономии не гарантируется. Впрочем, вы можете вывести и использовать свой коэффициент в зависимости от собственного опыта, понимания специфики планируемой нагрузки и готовности идти на риск не полностью сбывшихся прогнозов.
3. Кроме того, разные ВМ могут действительно использовать свою память полностью в разное время. Следовательно, суммарный объем занятой памяти на сервере в среднем окажется меньше суммарного объема памяти, выделенной для всех виртуальных машин. На этот эффект тоже не рекомендуют полагаться при сайзинге.
4. По причинам из пп. 2 и 3 делать большой запас по памяти не следует. Впрочем, большинство инфраструктур имеют обыкновение расти и в сторону увеличения количества ВМ, и в сторону увеличения нагрузки на существующие ВМ, так что при такой перспективе памяти лучше взять больше.

В vSphere версии 5 VMware изменила правила лицензирования таким образом, что лицензия теперь ограничивает максимальный объем памяти, выдаваемый виртуальным машинам. Например, бесплатная лицензия ESXi 5 позволит выдать не больше 32 Гб памяти всем ВМ одного сервера. А самая дорогая лицензия, Enterprise Plus, – до 96 Гб (до 192 для двухпроцессорного сервера, так как лицензий на сервер нужно по числу сокетов как минимум).

Эти правила имеет смысл учесть при планировании будущей инфраструктуры.

Сами правила лицензирования я здесь приводить не хочу (хотя не упомянутые мною нюансы имеют место быть) – так как правила имеют обыкновение меняться со временем. Свою задачу я вижу в упоминании того, что такой вопрос есть – и если он для вас актуален, то стоит обратиться к актуальному первоисточнику о правилах лицензирования (<http://www.vmware.com/products/vsphere/upgrade-center/licensing.html>).

1.7.3. Дисковая подсистема

Когда мы говорим про ресурсы дисковой подсистемы, то назвать их можно три:

- ❑ объем места;
- ❑ скорость чтения записи в количестве операций ввода-вывода в секунду (Input/Output per second, IOPS, или просто I/O);
- ❑ скорость чтения и записи в Мб/сек.

При планировании дисковой подсистемы для задачи виртуализации следует брать в расчет объем и производительность в IOps – для характера нагрузки виртуальных машин число операций ввода/вывода более важно, чем количество прочитанных/записанных мегабайт.

В идеале к планированию следует привлечь специалиста в области дисковой подсистемы/систем хранения данных. Однако базовые соображения я постараюсь привести. Сразу оговорюсь, что они применимы к хранилищам начального (локальные диски и СХД уровня HP P2000/MSA, Xyratex) и среднего уровней (NetApp FAS, HP P6000/EVA и P4000, EMC VNX и Clariion, HDS AMS и т. д.). Системы уровня «High end» (EMC Symmetrix DMX и V-MAX, HDS USP-V) сами устроены весьма сложно, и в двух словах даже самые простые рекомендации к ним не дашь.

Оговорка – у нас могут быть разные задачи (ВМ). Для каких-то приоритетом будет скорость, для других – объем. Возможно, на системе хранения будут созданы рейд-массивы (RAID-группы) с разными характеристиками – под разные задачи. Так вот, нижеприведенные соображения применимы к отдельно взятой RAID-группе.

Расчет требуемого места на системе хранения

Поговорим сначала про объем. Я приведу соображения, на которые следует ориентироваться, и пример расчета.

Соображения следующие:

- ❑ место на диске занимают сами файлы-диски виртуальных машин. Следовательно, нужно понять, сколько места нужно им;
- ❑ если для всех или части ВМ мы планируем использовать тонкие (thin) диски, то следует спланировать их первоначальный объем и последующий рост (здесь и далее под thin-дисками понимается соответствующий тип vmdk-файлов, то есть функция thin provisioning в реализации ESXi. Дело в том, что функционал thin provisioning может быть реализован на системе хранения независимо от ESXi, и я имею в виду не функционал систем хранения). Забегая вперед – я считаю оправданным пользоваться тонкими дисками для тестовых ВМ. Для производственных виртуальных машин было бы лучше их избегать;
- ❑ по умолчанию для каждой ВМ гипервизор создает файл подкачки, по размерам равный объему ее оперативной памяти. Этот файл подкачки располагается в папке ВМ (по умолчанию) или на отдельном LUN;
- ❑ если планируются к использованию снимки состояния, то под них тоже следует запланировать место. За точку отсчета можно взять следующие соображения:
 - если снимки состояния будут существовать короткий период после создания, например только на время резервного копирования, то под них запасаем процентов десять от размера диска ВМ;
 - если снимки состояния будут использоваться со средней или непрогнозируемой интенсивностью, то для них имеет смысл заложить порядка 30% от размера диска ВМ;
 - если снимки состояния для ВМ будут использоваться активно (что актуально в сценариях, когда ВМ используются для тестирования и разработки), то занимаемый ими объем может в разы превышать номинальный

размер виртуальных дисков. В этом случае точные рекомендации дать сложно, но за точку отсчета можно взять удвоение размера каждой ВМ. (Здесь и далее под снимками состояния понимается соответствующий функционал ESXi. Дело в том, что снимки состояния (snapshot) могут быть реализованы на системе хранения независимо от ESXi, и я имею в виду не функционал систем хранения.)

Примерная формула выглядит следующим образом:

Объем места для группы одинаковых/похожих ВМ = количество ВМ $\times$ $\times$ (размер диска $\times T$ + размер диска $\times S$ + объем памяти – объем памяти $\times R$).

Здесь:

- ☐ T – коэффициент тонких (thin) дисков. Если такие диски не используются, равен 1. Если используются, то абстрактно оценку дать сложно, зависит от характера приложения в ВМ. По сути, thin-диски занимают места на системе хранения меньше, чем номинальный размер диска. Так вот, этот коэффициент показывает, какую долю от номинального размера занимают диски виртуальных машин;
- ☐ S – размер снимков состояния. 10/30/200 процентов в зависимости от продолжительности непрерывного использования;
- ☐ R – процент зарезервированной памяти. Зарезервированная память в файл подкачки не помещается, файл подкачки создается меньшего размера. Размер его равен: объем памяти ВМ минус объем зарезервированной памяти.

Оценочные входные данные, для примера, см. в табл. 1.3.

Таблица 1.3. Данные для планирования объема дисковой подсистемы

Группа ВМ	Размер дисков одной ВМ, Гб	Используются тонкие диски?	Примерный размер размер снимков	Средний размер ОЗУ ВМ, Гб	Резервирование ОЗУ, %	Количество ВМ в группе
Инфраструктура	20	нет	10%	2	0	15
Сервера приложений	20 + 20	нет	10%	2	0	20
Критичные сервера приложений	20 + 80	нет	10%	6	50	10
Тестовые и временные	20	да	200%	2	0	20

Получаем оценку требуемого объема:

- ☐ инфраструктурная группа – $15 \times (20 + 20 \times 10\% + 2 - 2 \times 0) = 360$ Гб;
- ☐ сервера приложений – $20 \times (40 + 40 \times 10\% + 2 - 2 \times 0) = 920$ Гб;
- ☐ критичные сервера – $10 \times (100 + 100 \times 10\% + 6 - 6 \times 0,5) = 1130$ Гб;
- ☐ тестовые и временные – $20 \times (20 \times 30\% + (20 \times 30\%) \times 200\% + 2 - 2 \times 0) = 400$ Гб.

Всего получаем 2810 Гб.

Конкретные цифры в данных расчетах приведены лишь для иллюстрации подхода. Делайте поправки в зависимости от особенностей вашей инфраструктуры.

Хочу привести еще немного дополнительных соображений для планирования дисковой подсистемы.

Я склоняюсь к мысли, что использования тонких дисков для производственных ВМ следует стараться избегать. Причин две:

1. Для приложений, интенсивно записывающих данные на диск, «тонкость» дисков быстро сойдет на нет – в силу особенностей реализации.
2. Для производственных ВМ обычно нет задачи экономить место, зато есть задача обеспечить как можно большую доступность. Если не использовать тонкие диски – инфраструктура будет чуть более устойчива к проблемам из-за закончившегося места на хранилище.

Производительность дисковой подсистемы

Еще одним ресурсом дисковой подсистемы является производительность. В случае виртуальных машин скорость в Мб/сек не является надежным критерием, потому что при обращении большого количества ВМ на одни и те же диски обращения идут непоследовательно. Для виртуальной инфраструктуры более важной характеристикой является количество операций ввода-вывода (IOPS, Input/Output per second). Дисковая подсистема нашей инфраструктуры должна позволять больше этих операций, чем их запрашивают виртуальные машины.

Какой путь проходят обращения гостевой ОС к физическим дискам в общем случае:

1. Гостевая ОС передает запрос драйверу контроллера SAS/SCSI (который для нее эмулирует гипервизор).
2. Драйвер передает его на сам виртуальный контроллер SAS/SCSI.
3. Гипервизор перехватывает его, объединяет с запросами от других ВМ и передает общую очередь драйверу физического контроллера (HBA в случае FC и аппаратного iSCSI или Ethernet-контроллер в случае NFS и программного iSCSI).
4. Драйвер передает запрос на контроллер.
5. Контроллер передает его на систему хранения по сети передачи данных.
6. Контроллер системы хранения принимает запрос. Запрос этот – операция чтения или записи с какого-то LUN или тома NFS.
7. LUN – это «виртуальный раздел» на массиве RAID, состоящем из физических дисков. То есть запрос передается контроллером СХД на диски этого массива RAID (в случае NFS-системы хранения термин LUN не применим, но данные рассуждения абсолютно применимы к томам NFS).

Где может быть узкое место дисковой подсистемы:

- скорее всего, на уровне физических дисков. Важно количество физических дисков в массиве RAID. Чем их больше, тем лучше операции чтения-записи могут быть распараллелены. Также чем быстрее (в терминах I/O) сами диски, тем лучше;

- ❑ разные уровни массивов RAID обладают разной производительностью. Законченные рекомендации дать сложно, потому что, кроме скорости, типы RAID отличаются еще стоимостью и надежностью. Однако базовые соображения звучат так:
 - RAID-10 – самый быстрый, но наименее эффективно использует пространство дисков, отнимая 50% на поддержку отказоустойчивости;
 - RAID-6 – самый надежный, но страдает низкой производительностью на записи (30–40% от показателей RAID-10 при 100% записи), хотя чтение с него такое же быстрое, как с RAID-10;
 - RAID-5 компромиссен. Производительность на запись лучше RAID-6 (но хуже RAID-10), выше эффективность хранения (на отказоустойчивость забирается емкость всего одного диска). Но RAID-5 страдает от серьезных проблем, связанных с долгим восстановлением данных после выхода из строя диска в случае использования современных дисков большой емкости и больших RAID-групп, во время которого остается незащищенным от другого сбоя (превращаясь в RAID-0) и резко теряет в производительности;
 - RAID-0, или «RAID с нулевой отказоустойчивостью», для хранения значимых данных использовать нельзя;
- ❑ настройки системы хранения, в частности кеша контроллеров системы хранения. Изучение документации СХД важно для правильной ее настройки и эксплуатации;
- ❑ сеть передачи данных. Особенно если планируется к использованию IP СХД, iSCSI или NFS. Я ни в коем случае не хочу сказать, что не надо их использовать, – такие системы давно и многими эксплуатируются. Я хочу сказать, что надо постараться убедиться, что переносимой в виртуальную среду нагрузке хватит пропускной способности сети с планируемой пропускной способностью.

Результирующая скорость дисковой подсистемы следует из скорости дисков и алгоритма распараллеливания контроллером обращений к дискам (имеются в виду тип RAID и аналогичные функции). Также имеет значение отношение числа операций чтения к числу операций записи – это отношение мы берем из статистики или из документации к приложениям в наших ВМ.

Разберем пример. Примем, что наши ВМ будут создавать нагрузку до 1000 IOPS, 67% из которых будет составлять чтение, а 33% – запись. Сколько и каких дисков нам потребуется в случае использования RAID-10 и RAID-5? (Оговорюсь, что эти типы RAID упомянуты лишь для примера, как самые известные.)

В массиве RAID-10 в операциях чтения участвуют сразу все диски, а в операции записи – лишь половина (потому что каждый блок данных записывается сразу на два диска). В массиве RAID-5 в чтении участвуют все диски, но при записи каждого блока возникают накладные расходы, связанные с подсчетом и изменением контрольной суммы. Можно считать, что одна операция записи на массив RAID-5 вызывает четыре операции записи непосредственно на диски.

Это означает, что суть массивов RAID предполагает возникновение дополнительных операций чтения/записи. Если на систему хранения пришло A дисковых операций от ВМ, то непосредственно на диски записей и чтения произойдет больше, чем A .

Проиллюстрирую это расчетами. Напомню, что ВМ в моем примере генерируют тысячу операций чтения-записи в секунду, 1000 I/Ops.

Для RAID-10:

- ☐ чтение – $1000 \times 0,67\% = 670$ I/Ops;
- ☐ запись – $1000 \times 0,33\% = 330 \times 2$ (так как в записи участвует лишь половина дисков) = 660 I/Ops.

Всего от дисков нам надо 1330 I/Ops. Если поделить 1330 на количество I/Ops, заявленное в характеристиках производительности одного диска, получим требуемое количество дисков в массиве RAID-10 под указанную нагрузку.

Для RAID-5:

- ☐ чтение – $1000 \times 0,67\% = 670$ I/Ops;
- ☐ запись – $1000 \times 0,33\% = 330 \times 4 = 1320$ I/Ops.

Всего нам от дисков надо 1990 I/Ops.

По документации производителей один жесткий диск SAS 15k обрабатывает 150–180 I/Ops. Один диск SATA 7.2k – 70–100 I/Ops. Однако есть мнение, что лучше ориентироваться на несколько другие цифры: 50–60 для SATA и 100–120 для SAS.

Закончим пример.

При использовании RAID-10 и SATA нам потребуется 22–26 дисков.

При использовании RAID-5 и SAS нам потребуется 16–19 дисков.

Очевидно, что приведенные мною расчеты достаточно приблизительны. В системах хранения используются разного рода механизмы, в первую очередь кеширование – для оптимизации работы системы хранения. Но как отправная точка для понимания процесса сайзинга дисковой подсистемы эта информация пригодна.

Выбор количества LUN

Следующий вопрос: а на сколько LUN следует полученный объем распределить?

Ранее нас ограничивал объем в 2 Тб – ESXi предыдущих версий не мог использовать LUN большего размера. Однако сегодня можно создать LUN объемом до 64 Тб, и ESXi 5 замечательно сможет его использовать.

Получается, если рассматривать отдельно взятую raid-группу, то вариантов у нас два:

- ☐ мы можем создать один-единственный LUN/том NFS и вообще все ВМ разместить на нем одном;
- ☐ или все же сделать несколько LUN/томов и как-то между ними поделить виртуальные машины.

Для того чтобы принять решение, важными я вижу следующие соображения:

- ☐ иногда оправданно под разные группы ВМ или разные задачи делать отдельные LUN/тома NFS. Это может быть полезно:

- если для разных LUN/томов NFS есть возможность обеспечить разные скоростные характеристики;
 - если за их наполнение/обслуживание несут ответственность разные люди;
 - создав отдельный LUN и указав размещать на нем файлы подкачки виртуальных машин, мы сокращаем объем занятого места на основном хранилище для виртуальных машин;
 - если для этих виртуальных машин используются некоторые специфические функции системы хранения, в первую очередь репликация данных, то мы исключили эти файлы подкачки из процесса репликации. Файлы подкачки – это характерный пример оптимизации, но не единственно возможный;
- чем меньше LUN/томов, тем проще их администрировать. Поэтому создания дополнительных LUN/томов NFS лучше избегать;
- если в нашей системе хранения у каждого ее контроллера есть несколько сетевых интерфейсов (плюс обычно и самих контроллеров несколько) – то, создав несколько LUN/томов, мы можем нагрузку на каждый из них пустить по отдельному сетевому интерфейсу, а то и распределить между контроллерами. Из этих соображений лучше сделать LUN/томов не меньше, чем по числу сетевых интерфейсов системы хранения. А на одном LUN размещать виртуальных машин столько, чтобы для обработки их дискового трафика было достаточно одного пути от сервера к СХД.

За кадром остаются методики получения требуемого для ВМ количества IOPS и отношение чтения к записи. Для уже существующей инфраструктуры (при переносе ее в виртуальные машины) эти данные можно получить с помощью специальных средств сбора информации, например VMware Capacity Planner. Для инфраструктуры планируемой – из документации к приложениям и собственного опыта.

1.7.4. Сетевая подсистема

По поводу сети. Здесь мы говорим с точки зрения выбора конфигурации сервера. Я рискну предположить, что отправной точкой у нас является сервер с парой гигабитных сетевых интерфейсов, и нам надо определиться, будет ли этого достаточно. Или следует добавить еще пару, а то и четыре сетевых контроллера? Или даже выбрать модель с парой десятигигабитных интерфейсов?

Соображений для выбора я вижу три:

1. Необходимая пропускная способность для наших задач. Когда-то пары гигабитных контроллеров достаточно для двух десятков ВМ на одном сервере, в других случаях для какой-то ВМ со специфической задачей может захотеться выделить десятигигабитный контроллер в приватное пользование. Обычно сеть не является узким местом, но если хочется быть в этом уверенным – следует собрать статистику с переносимых в ВМ серверов. Для увеличения пропускной способности используем группировку сетевых контроллеров (nic teaming).
2. Требуемая доступность. Здесь нам поможет дублирование – опять же за счет группировки сетевых контроллеров (Teaming).

3. Безопасность. Здесь я имею в виду тот факт, что в редких случаях для повышения безопасности нам предписано использовать под какие-то задачи физически изолированную сеть, что автоматически требует выделенных сетевых контроллеров под эту задачу и отдельных контроллеров – под остальные.

Для реализации всех соображений нам может потребоваться несколько (много ☺) сетевых контроллеров. Абсолютный минимум – 1. Идеальный минимум – 4 или даже 6, в зависимости от планируемых к использованию функций vSphere. Такие функции, как vMotion, Fault Tolerance, работа с системами хранения поверх протокола IP, требуют ресурсов сети, и под них рекомендуется выделять отдельные физические контроллеры.

Как посчитать, сколько требуется в конкретном случае? Нам требуется понять:

1. Сколько разных типов трафика у нас будет (список вариантов абзацем ниже).
2. Требуются ли для трафика какой-то задачи физически выделенные сетевые контроллеры – из соображений производительности и/или безопасности.
3. Есть ли задачи, для трафика которых мы можем не дублировать сетевые контроллеры? Здесь акцент на том, что резервирование важно для трафика любого типа (кроме разве что трафика тестовых ВМ), но, может быть, где-то придется пойти на компромисс, чтобы снизить требуемое количество сетевых контроллеров.

Сеть нам нужна для:

1. Управления. Интерфейс VMkernel, настроенный для Management traffic. Задача управления у нас будет наверняка.
2. vMotion. Скорее всего, вы будете использовать живую миграцию – значит, трафик этой задачи у вас будет.
3. Fault Tolerance. Если вы хотите защищать свои ВМ при помощи этой функции, то ее трафик стоит отдельно учесть при планировании сети.
4. iSCSI и/или NFS, если они будут использоваться.
5. ВМ. Среди ВМ может быть несколько групп ВМ, каждой из которых может потребоваться выделенный сетевой интерфейс (или не один).

Для управления нам минимально необходим один сетевой контроллер. С технической точки зрения, он не обязан быть выделенным – но этого может требовать политика вашей компании. Конечно, лучше бы нам управляющий интерфейс разместить на парочке контроллеров – для надежности.

Для vMotion нам необходим выделенный сетевой интерфейс. С технической точки зрения, он может быть невыделенным – живая миграция будет происходить, и если через тот же физический контроллер идет еще какой-то трафик. Но поддерживаемая конфигурация сервера для vMotion предполагает выделение контроллера под трафик vMotion. Нужно ли его дублировать, зависит от того, насколько для вас критична невозможность какое-то время мигрировать ВМ с сервера в случае проблем в сети, используемой для vMotion.

Для Fault Tolerance соображения примерно те же, что и для vMotion. Но, скорее всего, необходимость наличия дублирующего контроллера для FT более вероятна. Однако нагрузка на сетевой контроллер со стороны других задач может помешать нормальной работе Fault Tolerance. Верно и в обратную сторону: если трафик Fault Tolerance будет достаточно велик, он может мешать другим задачам, если они разделяют один и тот же физический сетевой интерфейс.

Если активно будет использоваться система хранения iSCSI или NFS, то с точки зрения производительности ее трафик лучше выделить в отдельную сеть. И в любом случае рекомендуется это сделать с точки зрения безопасности. Для надежности, очевидно, следует выделить под данный трафик хотя бы пару сетевых контроллеров.

Для ВМ – зависит от ваших нужд. Какой предполагается организация виртуальной сети? Смогут ли все ВМ использовать один и тот же контроллер (группу контроллеров)? Необходимо ли дублирование?

По результатам ответов на эти вопросы определяемся с необходимым числом сетевых контроллеров на типовом сервере ESXi.

Обратите внимание на то, что есть возможность для всех вероятных задач использовать всего один физический контроллер. Или одну пару. Это возможно технически, но может быть неприменимо из-за конфигурации физической сети или из-за недостаточной производительности. Так что после того как мы ответим на вопрос номер 1 – трафик каких задач будет на нашей vSphere, нам следует подумать: а какие задачи можно объединить на одних и тех же контроллерах.

Плохие кандидаты на объединение – трафик iSCSI и NFS. Под IP-системы хранения настоятельно рекомендуется выделить пару физических сетевых контроллеров.

Популярные кандидаты на объединение – трафик управления и vMotion. Часто используется конфигурация, когда эти две задачи используют один и тот же виртуальный коммутатор с парой физических интерфейсов.

Необходимо ли выделять отдельные сетевые контроллеры под трафик виртуальных машин, сильно зависит от прочих факторов, в первую очередь от организации физической сети, поэтому в общем сказать сложно.

Если речь идет про использование сетевых контроллеров 10 Гбит, то обычно используются сервера с одной парой таких контроллеров. Для решения потенциальных проблем с производительностью следует использовать механизм NIOC, Network IO Control, доступный в самой дорогой лицензии vSphere, при использовании распределенных виртуальных коммутаторов.

Более подробная информация по непосредственной настройке и планированию виртуальных коммутаторов будет приведена в соответствующем разделе, посвященном сетям vSphere.

1.7.5. Масштабируемость: мало мощных серверов или много небольших?

Заключительный вопрос. Допустим, вы определились с конфигурацией СХД. Посчитали, сколько процессорной мощности и оперативной памяти необходимо вам для размещения всех ВМ. Но как распределить их по серверам? Например, 16 про-

цессоров и 256 Гб ОЗУ можно разместить в 4 четырехпроцессорных серверах по 64 Гб в каждом или 8 двухпроцессорных по 32 Гб. Что выбрать при прочих равных?

Здесь (как и везде в этом разделе) сложно дать однозначные рекомендации, поэтому привожу свои соображения.

1. Модельный ряд того поставщика серверного оборудования, с которым вы работаете. Если все ваши используемые сервера (пусть других моделей) произведены фирмой XYZ и вы ими довольны, настоятельно рекомендую не «разводить зоопарк» и не пускаться в эксперименты с другими поставщиками. Потом сами будете не рады.
2. Конечно, предыдущий совет не включает ситуации, когда вас не устраивает качество или уровень поддержки имеющихся серверов, или ваш поставщик не предлагает серверов нужной конфигурации.
3. При прочих равных обратите внимание на дополнительные функции. Например, крупные сервера уровня предприятия обычно обладают полезными возможностями вроде горячей замены или развитыми функциями удаленного управления.
4. Крупные сервера создают меньшие накладные расходы, а также позволяют вместить суммарно больше ВМ. В нашем примере это можно проиллюстрировать следующим образом. В каждый из четырех серверов по 64 Гб вы сможете с достаточным комфортом разместить по 21 ВМ по 3 Гб ОЗУ каждая (всего получается $4 \times 21 = 84$ ВМ). В каждый из восьми серверов по 32 Гб – по десять таких же ВМ, всего получается $8 \times 10 = 80$. Итого получается, что в более крупных серверах плотность размещения ВМ выше и, соответственно, в то же суммарное количество ОЗУ можно вместить больше ВМ. Если же учесть экономию от Transparent Page Sharing, то разница станет еще более заметна.
5. С другой стороны, нельзя забывать все сервера под завязку. Надо подумать и о доступности своих служб. Разумно запланировать, что один из серверов может отказать, а еще один в это время может находиться на обслуживании (например, обновлении ПО или оборудования). Итого мы хотим, чтобы наше решение сохраняло доступность при одновременном выходе из строя двух узлов. В первом случае мы таким образом лишаемся половины ($4 - 2 = 2$) серверов. И получается, что мы можем задействовать с пользой лишь $256 - (2 \times 64) = 128$ Гб ОЗУ и разместить в них только $(4 - 2) \times 21 = 42$ ВМ. Во втором случае нам остается уже $8 - 2 = 6$ серверов с $256 - (2 \times 32) = 192$ Гб, в которых поместится $(8 - 2) \times 10 = 60$ ВМ! Как видим, с учетом планирования доступности преимущество в нашем примере оказывается уже на стороне более модульной архитектуры.
6. С третьей стороны, вы ни под каким видом не сможете разделить нагрузку одной ВМ между несколькими серверами. Иными словами, если вам вдруг однажды потребуется создать ВМ с объемом ОЗУ 40 Гб, то второй сценарий (сервера по 32 Гб) вам просто не подойдет (без учета возможностей по оптимизации использования памяти, которые мы здесь не станем учитывать для наглядности). А вот первому варианту (сервера по 64 Гб) такой сценарий окажется вполне под силу.

7. Все вышесказанное в равной степени применимо и к другим ресурсам сервера (процессор, подсистемы ввода-вывода). Просто с памятью это выглядит наиболее наглядно.
8. Сервера-лезвия (Blade servers) обладают массой преимуществ с точки зрения стоимости владения (затраты на управление, обслуживание, питание и охлаждение). Они также отлично отвечают пятому соображению, приведенному выше, – ведь каждое лезвие является сравнительно небольшой составляющей инфраструктуры. В результате выход из строя сразу нескольких из них вряд ли сильно уменьшит суммарную мощность системы. Блейд-сервера также крайне эффективны с точки зрения уменьшения проблем и головной боли – куда девать все эти кабели? Предположим, что, следуя соображениям отказоустойчивости, мы будем использовать 6 одногигабитных сетевых интерфейсов (2 = управление + vMotion, 2 = iSCSI, 2 = виртуальные машины). Не забудем о сети IPMI/iLO – в итоге 7 интерфейсов (и кабелей) на сервер. 8 серверов виртуализации в результате дают нам 56 кабелей, которые к тому же надо куда-то подключать, то есть потребуется еще и 56 сетевых портов. В случае с блейд-серверами количество кабелей (и портов) сокращается до 10–12.
9. С другой стороны, стоит задуматься о сбоях не только на уровне каждого лезвия в отдельности, но и каждой корзины (шасси). А вот выход из строя целой корзины окажется все-таки более существен, чем даже пары обычных стоечных серверов, хотя, конечно, куда менее вероятен.
10. С третьей стороны, увеличивая нашу инфраструктуру и поднимая требования к доступности, может потребоваться вести счет уже не на потерю отдельных серверов или корзин, а целых стоек. И здесь нет особой разницы, потеряли вы стойку с лезвиями или стойку с обычными серверами. То есть преимущества лезвий могут снова выйти на первый план.

В общем, надеюсь, что общую идею и способы оценки привлекательности разных подходов к выбору типа серверов вы поняли, и это оказалось для вас полезным. Далее вам придется решать самим, руководствуясь подобными или какими-то совершенно другими соображениями.

Но, как это ни странно, аналогичные соображения могут действовать не только на оборудование и доступность служб, но и на лицензирование ПО, которое вы планируете запускать в виртуальных машинах. Рекомендую заранее ознакомиться с тонкостями лицензирования этого ПО и нюансами, связанными с виртуализацией.

Например, некоторые версии серверного ПО требуют приобретения лицензий по количеству всех процессоров в физическом сервере – даже в том случае, если для ВМ с этим ПО вы выделили всего один виртуальный процессор. В этом случае более маленькие сервера с небольшим количеством процессоров очевидно оказываются выгоднее. Правда, в последнее время с развитием популярности виртуализации в промышленных инфраструктурах поставщики ПО стали постепенно отказываться от этой крайне невыгодной схемы лицензирования.

Другие производители могут накладывать ограничения на перепривязку лицензий между серверами и считают таким переносом операции миграции ВМ (например, vMotion). И здесь, уже наоборот, крупные сервера, которые снижают частоту (или необходимость) переноса ВМ, могут оказаться выгоднее.



Глава 2. Настройка сети виртуальной инфраструктуры

В этой главе будет рассказано про то, как работают с сетью сервера ESXi, про организацию сети в виртуальной инфраструктуре vSphere. Ключевой элемент сети в vSphere – это виртуальный коммутатор, virtual switch или vSwitch. Виртуальные коммутаторы делятся на несколько типов: стандартный виртуальный коммутатор VMware, распределенный виртуальный коммутатор VMware (dvSwitch, distributed vSwitch) и виртуальный коммутатор стороннего производителя (на данный момент такой предлагает Cisco, модель Nexus 1000V; и IBM, модель IBM System Networking Distributed Virtual Switch 5000V). Стандартный виртуальный коммутатор VMware доступен во всех вариантах лицензирования vSphere, включая бесплатный ESXi. Распределенный и третьесторонний виртуальные коммутаторы – лишь в дорогих лицензиях. Поговорим про них последовательно.

2.1. Основы сети ESXi, объекты виртуальной сети

Основное соображение, которое необходимо уяснить: физические сетевые контроллеры сервера ESXi не являются «активными сетевыми устройствами». Это означает, что у физического сетевого контроллера нет своего IP-адреса, его MAC-адрес фигурирует лишь в техническом трафике. А являются активными сетевыми устройствами сетевые контроллеры виртуальные.

Очевидно, что виртуальные сетевые контроллеры гипервизор создает для виртуальных машин. Но и для себя самого гипервизор тоже использует виртуальные сетевые контроллеры (рис. 2.1).

На данном рисунке фигурируют и объекты внешней сети – физические коммутаторы.

Если вы используете сервер без виртуализации, устанавливаете на него какую-то ОС и настраиваете подключение к сети, то настраиваете вы физические сетевые контроллеры, IP-адреса, группировку контроллеров, VLAN и прочее, что может понадобиться для сети этого сервера.

Если же мы настраиваем сеть на ESXi, то физические сетевые контроллеры являются лишь каналами во внешнюю сеть. Через один физический сетевой контроллер в сеть могут выходить и управляющий интерфейс, и интерфейсы для подключения NFS/iSCSI/vMotion/Fault Tolerance (все это – виртуальные сетевые карты VMkernel, гипервизора), и разные виртуальные машины. (Здесь имеется

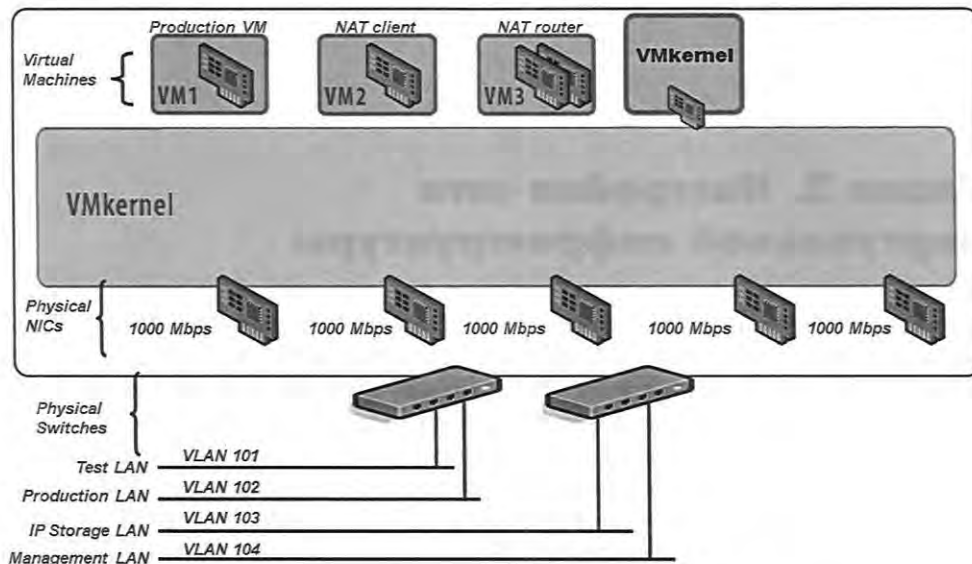


Рис. 2.1. Основные объекты сети «внутри» ESXi

Источник: VMware

в виду принципиальная возможность. Трафик разных назначений рекомендуется разделять по разным физическим сетевым контроллерам, см. раздел, посвященный сайзингу.)

Связующим звеном между источниками трафика (виртуальными сетевыми контроллерами виртуальных машин и гипервизора) и каналами во внешнюю сеть (физическими сетевыми контроллерами) являются виртуальные коммутаторы (рис. 2.2).

На данном рисунке фигурируют и объекты внешней сети – физические коммутаторы.

Перечислим объекты виртуальной сети:

- ❑ физические сетевые контроллеры (network interface card, NIC) – те, что установлены в сервере. ESXi дает им имена вида `vmnic#`. Таким образом, когда вам по тексту попадется термин «`vmnic`», знайте: речь идет о физическом сетевом контроллере сервера. Они же имеются в виду под словосочетанием «канал во внешнюю сеть», или, для краткости, «аплинк»;
- ❑ виртуальные коммутаторы (vSwitch или вКоммутаторы) – основные объекты сети на ESXi. Их задача – связать между собой виртуальные сетевые контроллеры и дать им выход во внешнюю сеть через физические сетевые контроллеры;
- ❑ группы портов (Port groups) – логические объекты, создаваемые на вКоммутаторах. Виртуальные сетевые контроллеры подключаются именно к группам портов;

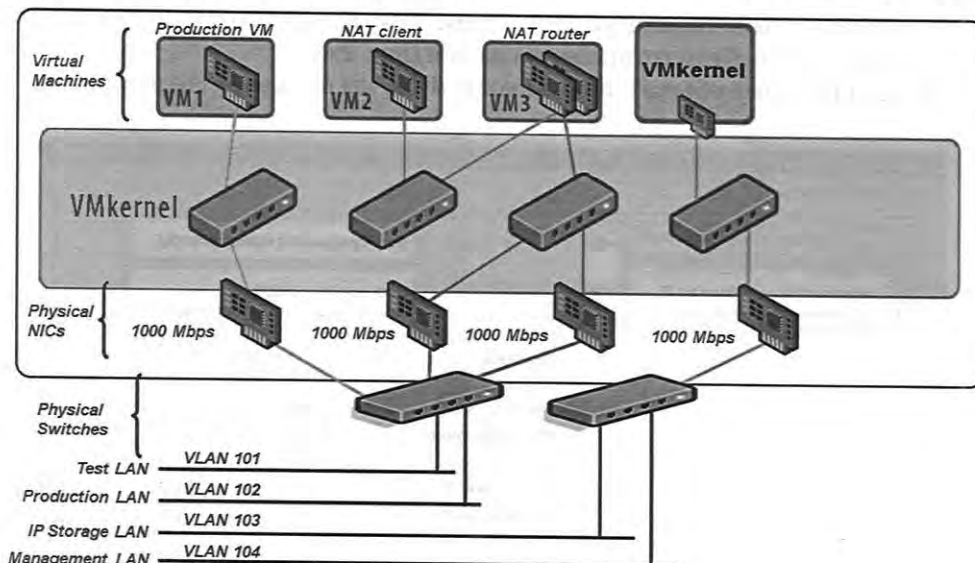


Рис. 2.2. Связь между объектами сети «внутри» ESXi
Источник: VMware

- ❑ виртуальные сетевые контроллеры. Они могут принадлежать виртуальным машинам или VMkernel.

Обратите внимание на то, что интерфейс vSphere весьма наглядно показывает вам связь между объектами виртуальной сети (рис. 2.3). Сравните со схемой на рис. 2.2.

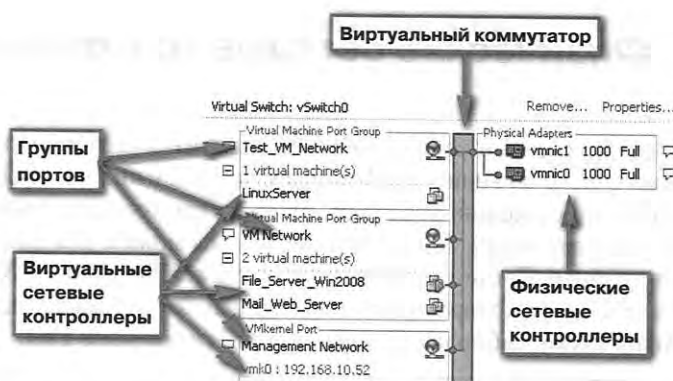


Рис. 2.3. Объекты виртуальной сети на ESXi
в интерфейсе клиента vSphere

Если зайти в свойства виртуального коммутатора, то мы получим доступ к его настройкам и настройкам групп портов на нем (рис. 2.4).

Выделив нужный объект и нажав кнопку **Edit**, мы попадем в его настройки.

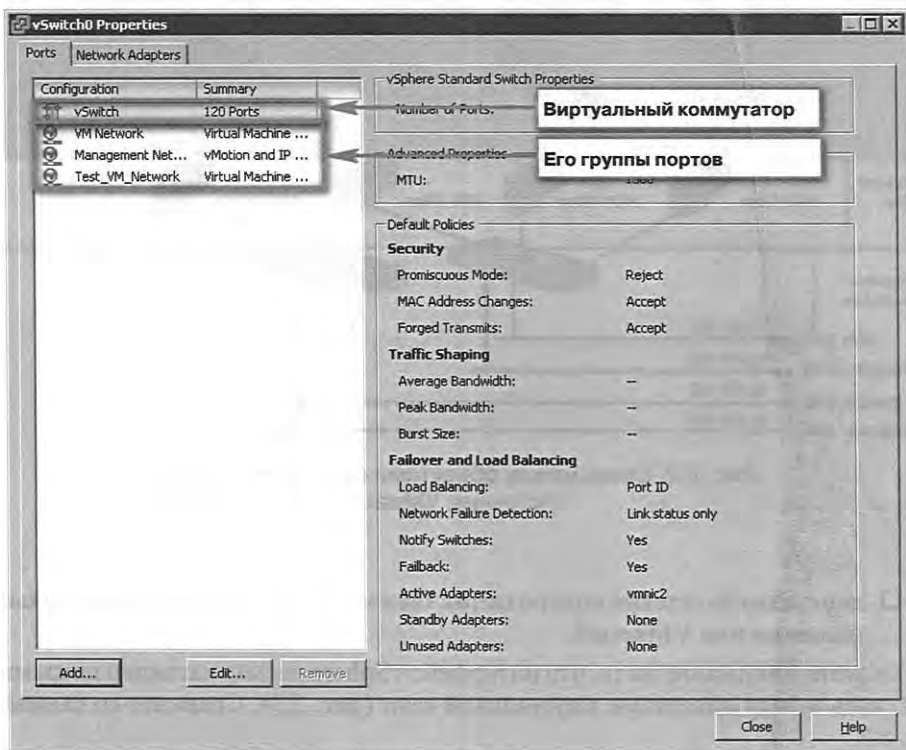


Рис. 2.4. Свойства виртуального коммутатора

2.1.1. Физические сетевые контроллеры, vmnic

Про физические сетевые контроллеры сказать особо нечего – они используются просто как каналы во внешнюю физическую сеть, у них нет собственного IP-адреса, и их MAC-адрес можно отследить лишь в техническом, вспомогательном трафике ESXi (см. beaconing). Единственное, что мы можем для них настроить, – это скорость и дуплекс. Делается это из GUI так: **Configuration** ⇒ **Networking** ⇒ **Properties** для vSwitch (к которому подключен физический контроллер) ⇒ вкладка **Network Adapters** ⇒ выбираем нужный vmnic и нажимаем **Edit**.

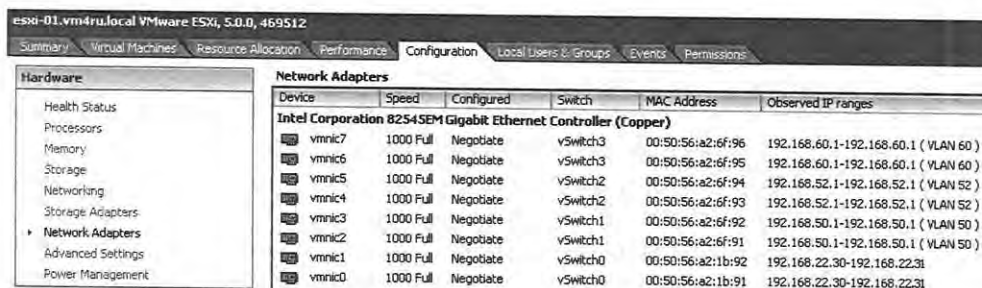
Обратите внимание. Иногда при нестабильной работе сети может помочь смена явного указания скорости и дуплекса (обычно 1 Гбит/с полный дуплекс) на автосоглашение (auto negotiate). Или наоборот.

Каждый физический сетевой контроллер нам надо привязать к вКоммутатору. Можно, конечно, и не привязывать – но это имеет смысл, только лишь если мы хотим отдать этот vnic напрямую какой-то ВМ. Подробности о подобном варианте см. в разделе, посвященном виртуальным машинам, вас интересует функция VMDirectPath.

Итак, если мы не планируем использовать VMDirectPath, то не привязанный к вКоммутатору сетевой контроллер сервера у нас не используется никак. Это бессмысленно, поэтому, скорее всего, все vnics будут привязаны к тем или иным виртуальным коммутаторам.

Есть логичное правило: один физический сетевой контроллер может быть привязан к одному и только одному вКоммутатору. Но к одному вКоммутатору могут быть привязаны несколько vnics. В последнем случае мы можем получить отказоустойчивую и более производительную конфигурацию сети.

Получить информацию обо всех vnics одного сервера можно в пункте **Configuration ⇒ Network Adapters** (рис. 2.5).



Device	Speed	Configured	Switch	MAC Address	Observed IP ranges
Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)					
vnic7	1000 Full	Negotiate	vSwitch3	00:50:56:a2:6f:96	192.168.60.1-192.168.60.1 (VLAN 60)
vnic6	1000 Full	Negotiate	vSwitch3	00:50:56:a2:6f:95	192.168.60.1-192.168.60.1 (VLAN 60)
vnic5	1000 Full	Negotiate	vSwitch2	00:50:56:a2:6f:94	192.168.52.1-192.168.52.1 (VLAN 52)
vnic4	1000 Full	Negotiate	vSwitch2	00:50:56:a2:6f:93	192.168.52.1-192.168.52.1 (VLAN 52)
vnic3	1000 Full	Negotiate	vSwitch1	00:50:56:a2:6f:92	192.168.50.1-192.168.50.1 (VLAN 50)
vnic2	1000 Full	Negotiate	vSwitch1	00:50:56:a2:6f:91	192.168.50.1-192.168.50.1 (VLAN 50)
vnic1	1000 Full	Negotiate	vSwitch0	00:50:56:a2:1b:92	192.168.22.30-192.168.22.31
vnic0	1000 Full	Negotiate	vSwitch0	00:50:56:a2:1b:91	192.168.22.30-192.168.22.31

Рис. 2.5. Информация обо всех физических сетевых контроллерах сервера ESXi

В столбце **vSwitch** мы видим, к какому виртуальному коммутатору они привязаны. В столбце **Observed IP ranges** – пакеты из каких подсетей и VLAN получает ESXi на этом интерфейсе. Это информационное поле, его значения не всегда точны и актуальны, но часто оно помогает нам сориентироваться. Например, из рис. 2.5 ясно, что сетевые контроллеры vnic0 и vnic1 перехватывают трафик из одной подсети, а vnic2 и vnic3 – из другой, vnic4 и vnic5 – из третьей и т. д. Скорее всего, это означает, что каждая пара контроллеров подключена к другому физическому коммутатору или принадлежат другому/другим VLAN на физическом коммутаторе.

Обратите внимание. Физическим сетевым контроллерам ESXi дает имя вида vnic. Командой `esxcfg-nics -l` вы выведете на экран список всех физических контроллеров сервера ESXi и информацию о них. Фактически эта команда покажет на ту же информацию, что и соответствующее окно графического интерфейса (рис. 2.5). Новая версия команды для тех же целей – `esxcli network nic list`.

Теперь поговорим о прочих компонентах виртуальной сети – виртуальных сетевых контроллерах и виртуальных коммутаторах с группами портов.

2.1.2. Виртуальные контроллеры VMkernel

Первое, чем отличаются виртуальные сетевые контроллеры, – это принадлежность. Принадлежат они VMkernel (самому гипервизору) или виртуальным машинам.

Если виртуальный контроллер принадлежит ВМ, то он может быть разных типов – Flexible, vmxnet2, vmxnet3, E1000. Но про них поговорим в разделе, посвященном свойствам и оборудованию виртуальных машин, а здесь сконцентрируем внимание на виртуальных сетевых контроллерах для VMkernel.

Гипервизору (VMkernel) требуются свои собственные сетевые интерфейсы (виртуальные, напомню) для:

- ❑ управления сервером. В том смысле что нам для управления сервером ESXi по сети следует обращаться на IP-адрес его виртуального сетевого контроллера;
- ❑ vMotion – по этим интерфейсам будет передаваться содержимое оперативной памяти переносимой ВМ;
- ❑ подключения дисковых ресурсов по iSCSI в случае использования программного инициатора iSCSI;
- ❑ подключения дисковых ресурсов по NFS;
- ❑ Fault Tolerance – по этим интерфейсам будут передаваться процессорные инструкции на резервную ВМ в Fault Tolerance-паре.

См. подробности и требования к сети для соответствующих функций в разделах, им посвященных.

Таким образом, работа гипервизора с сетью немного двойственна, потому что происходит на двух уровнях. С одной стороны, гипервизор имеет доступ и управляет физическими контроллерами; с другой – сам для себя, для своего доступа в сеть создает контроллеры виртуальные. Тем не менее такая схема весьма удобна для понимания: физические контроллеры – это всегда «ресурс», всегда только канал во внешнюю сеть. А как источник трафика, как активные объекты сети всегда выступают контроллеры виртуальные, в том числе для трафика гипервизора.

В качестве интерфейсов управления для ESXi используются виртуальные сетевые контроллеры VMkernel, те из них, в свойствах которых установлен флажок **Management traffic** (рис. 2.6).

Еще раз – если где-либо в тексте вам встретится упоминание об «управляющем интерфейсе ESXi», «управляющем интерфейсе VMkernel», «интерфейсе управления» и т. п., то речь идет о виртуальном сетевом интерфейсе VMkernel с флажком **Management traffic** в свойствах. Таких может быть несколько на одном сервере.

Один виртуальный сетевой контроллер для VMkernel создается установщиком, он используется для управления сервером ESXi. Именно ему принадлежит тот IP-адрес, который вы указывали при установке. Именно на IP-адрес этого первого виртуального интерфейса гипервизора вы подключаетесь клиентом vSphere, через него с сервером работает vCenter, на этот IP подключаются утилиты для работы с ESXi. Также через интерфейс VMkernel с флажком **Management traffic**

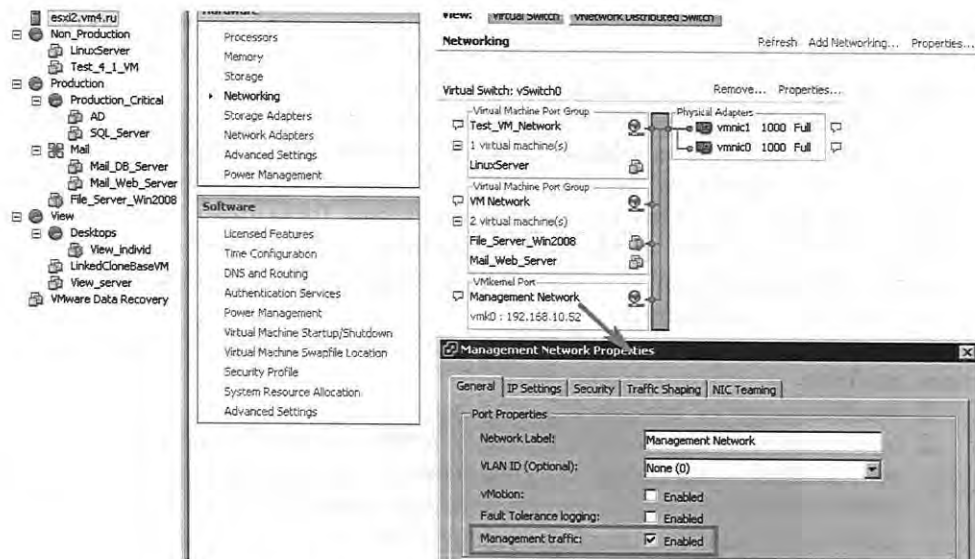


Рис. 2.6. Управляющий интерфейс ESXi

идут сигналы пульса (heartbeat) при работе кластера VMware HA. Наконец, некоторые варианты резервного копирования виртуальных машин осуществляются через управляющие интерфейсы.

Вам может потребоваться создать дополнительный управляющий интерфейс. Основная причина для этого – необходимость обеспечить дублирование (рис. 2.7).

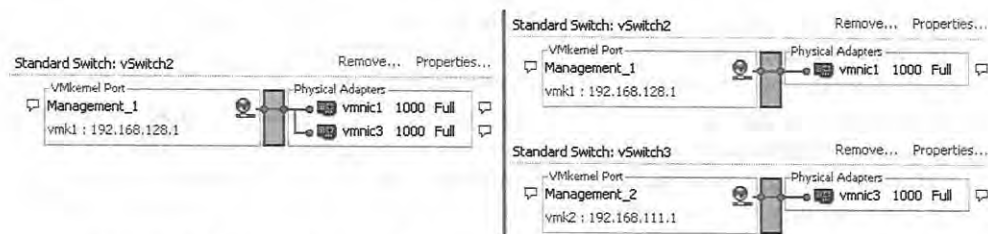


Рис. 2.7. Два варианта дублирования управляющего интерфейса ESXi

В левой части рисунка вы видите конфигурацию с резервированием единственного управляющего интерфейса. Это за счет того, что к vКоммутатору подключены два сетевых контроллера. Таким образом, выход из строя одной физической сетевой карточки (а если они подключены к разным физическим коммутаторам – то и одного из них) не приведет к недоступности управления сервером ESXi по сети.

В правой части рисунка вы тоже видите конфигурацию с резервированием, но резервирование происходит по-другому – созданы два интерфейса VMkernel на разных vКоммутаторах, следовательно, на разных vmnic. Если выйдет из строя

один из каналов во внешнюю сеть (сам vmpic или порт в физическом коммутаторе, к которому он подключен, или сам физический коммутатор), то один из управляющих интерфейсов станет недоступен, но останется другой.

Первый вариант удобнее в использовании, но если мы не можем себе позволить выделить два vmpic на вКоммутатор с управляющим интерфейсом, то он будет невозможен. Выделить может не получиться, если количество сетевых контроллеров в сервере недостаточно под все наши задачи. Тогда имеет смысл пойти по второму пути. Само собой, на вКоммутаторах интерфейсы VMkernel могут соседствовать с любыми другими виртуальными интерфейсами в любых количествах — на рис. 2.6 их нет для простоты.

Однако первый вариант резервирования не защитит от двух дополнительных типов проблем:

- ❑ ошибки настроек интерфейса управления;
- ❑ проблемы во внешней сети. Во втором примере двум разным управляющим интерфейсам даны адреса из разных сетей, и это защитит в том случае, если одна из этих сетей начнет испытывать проблемы (например, перегрузка паразитным трафиком или сбой маршрутизатора).

Создать еще один интерфейс очень просто: **Configuration** ⇒ **Networking** ⇒ **add Networking** (для создания нового вКоммутатора — то есть резервирование по правому варианту с рис. 2.6). Нас спросят, группу портов для чего мы хотим создать на этом вКоммутаторе. Несложно догадаться, что сейчас нас интересует VMkernel. Выбираем, жмем **Next**. Выберем, какие vmpic привязать к создаваемому сейчас вКоммутатору. **Next**. Указываем имя (**Network Label**) — это название группы портов.

Это название — для человека, так что рекомендую давать говорящие названия. «Management_port_2» вполне подойдет.

Однако при чем здесь группа портов, мы ведь создаем виртуальный сетевой интерфейс VMkernel? Дело в том, что любой виртуальный сетевой контроллер числится подключенным именно к группе портов, поэтому при создании интерфейса VMkernel из GUI мы создаем и интерфейс VMkernel (его имя — vmk#, то есть первый будет назван vmk0, второй — vmk1 и т. д.) и группу портов для него.

На стандартном виртуальном коммутаторе интерфейс VMkernel всегда занимает свою группу портов целиком (или, что то же самое, группа портов для VMkernel всегда состоит из одного порта). Этот факт не является ограничением — на одном вКоммутаторе могут сосуществовать любые виртуальные интерфейсы в любых комбинациях, просто под каждый интерфейс VMkernel создается отдельная группа портов.

Обратите внимание. Я не рекомендую использовать пробелы и спецсимволы в каких бы то ни было названиях. Это не смертельно, но может создать дополнительные проблемы при работе в командной строке и сценариях.

Затем для виртуального контроллера указываем настройки IP. В принципе, все. Единственный, наверное, нюанс — если хотим создать интерфейс VMkernel не на новом, а на уже существующем вКоммутаторе, тогда делаем так: **Configuration**

⇒ **Networking** ⇒ для нужного в Коммутатора нажимаем **Properties** ⇒ и на вкладке **Ports** нажимаем **Add**. Дальше все так же.

Наконец, в случае распределенных виртуальных коммутаторов пройдите **Configuration** ⇒ **Networking** ⇒ кнопка **Distributed Virtual switch** ⇒ ссылка **Manage Virtual Adapters** ⇒ **Add**.

Будьте аккуратны в случае изменения настроек интерфейса управления, когда он один, – в случае ошибки (например, опечатки в IP-адресе) доступ к управлению ESXi по сети станет невозможен. Решается такая проблема из командной строки или из BIOS-подобного локального интерфейса. В любом случае понадобится доступ к локальной консоли сервера – физически или по iLO и аналогам.

Обратите внимание на то, что для интерфейса VMkernel у нас есть возможность установить четыре флажка (рис. 2.8):

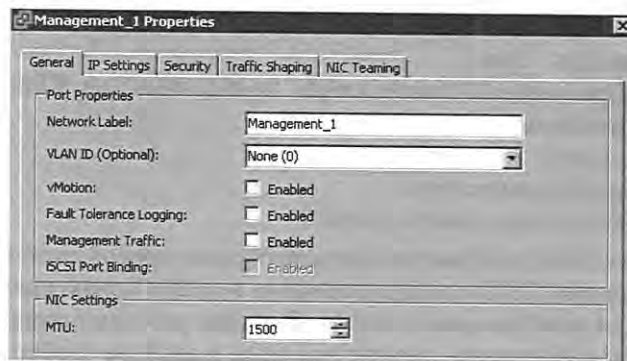


Рис. 2.8. Настройки интерфейса VMkernel

Каждый из флажков (кроме **iSCSI Port Binding**) разрешает передачу соответствующего трафика через данный интерфейс. Технически допускается любая комбинация флажков на одном виртуальном интерфейсе. Однако настоятельно рекомендуется трафик разных типов разносить по разным физическим сетевым контроллерам, из чего следует создание отдельного интерфейса VMkernel под разные типы трафика гипервизора. Устанавливать флажки следует только по необходимости.

Если вы планируете использовать один интерфейс для управления и обращения к IP-СХД, то лучше всего создать несколько интерфейсов, пусть даже из одной подсети и на одном и том же виртуальном коммутаторе.

Флажок **iSCSI Port Binding** неактивен по той причине, что устанавливается он из настроек инициатора iSCSI, а в этом месте интерфейса имеет информационную функцию. Этот флажок следует устанавливать при настройке iSCSI multipathing – см. соответствующий раздел.

Обратите внимание. Виртуальные сетевые контроллеры этого типа называются vmk – при создании их из GUI им даются имена вида vmk#, в командной строке мы управляем этими объектами с помощью команды `esxcfg-vmknic`. Новая команда для этой же цели – `esxcli network ip interface list`.

2.2. Стандартные виртуальные коммутаторы VMware – vNetwork Switch

Если вы перейдете на вкладку **Configuration** $\Rightarrow$ **Networking**, то увидите вашу виртуальную сеть.

Кстати, из командной строки можно увидеть все то же самое командой `esxcfg-vswitch -l`. Новая команда для этой цели:

`esxcli network vswitch standard list` для стандартных вКоммутаторов

или

`esxcli network vswitch dvs vmware list` для распределенных вКоммутаторов VMware.

Виртуальная сеть – это:

- ☐ виртуальные сетевые контроллеры, принадлежащие VMkernel или виртуальным машинам;
- ☐ группы портов;
- ☐ виртуальные коммутаторы;
- ☐ физические сетевые контроллеры.

Про первые и последние мы говорили чуть ранее, теперь коснемся остального.

Начну с аналогии. ESXi с работающими на нем VM можно представить в виде серверной стойки или серверного шкафа. В стойке работают сервера, эти сервера подключены к коммутаторам, смонтированным в этой же стойке, и в некоторые порты коммутаторов подключены выходящие за пределы стойки сетевые кабели – Uplinks.

Это прямая аналогия. Вся стойка и есть ESXi, сервера внутри – VM, коммутаторы – виртуальные коммутаторы, подключения ко внешней сети (Uplinks) – это физические сетевые контроллеры. Что же тогда «ports group»? В общем-то, группами портов они и являются.

Смотрите – допустим, у вас есть несколько физических серверов, подключенных к одному коммутатору. Согласитесь, что у вас может возникнуть желание сделать на коммутаторе разные настройки для этих серверов. Указать им разные VLAN, указать разные настройки ограничения пропускной способности сети (traffic shaping), что-нибудь еще... На коммутаторе физическом большинство таких настроек выполняются для порта, некоторые – для группы портов. То есть в настройках вы указываете – хочу создать группу портов, в нее включить порты с 1-го по 12-й и им задать значение VLAN = 34.

В коммутаторе виртуальном стандартном все так же. Единственное, наверное, различие – в том, что вы не указываете настроек на уровне порта – только на уровне групп портов, и не указываете номеров портов, входящих в группу, и даже их количества.

Таким образом, для настройки тех же самых VLAN или ограничения пропускной способности сети (traffic shaping) вам достаточно создать на вКоммутаторе

объект «группа портов» с каким-то говорящим (для вашего удобства) именем. Например: «Production», «Test_Project_SCOM» и т. п. Затем задать для нее необходимые настройки и, наконец, в свойствах ВМ указать – сетевой контроллер подключен к группе портов «Production». Все. Нам не надо выбирать, к порту с каким номером мы хотим подключить ВМ. Не надо добавлять еще портов в группу. Группа портов – логическое образование, количество портов в группе не ограничено – она будет расти по мере добавления в нее ВМ. Ограничено только количество портов на весь вКоммутатор, то есть на количество виртуальных сетевых контроллеров (не ВМ, так как в одной ВМ может быть несколько виртуальных адаптеров), подключенных ко всем группам портов на этом вКоммутаторе.

Отсюда выводы:

Если вы хотите вывести во внешнюю сеть ВМ через определенные физические сетевые контроллеры, то вам надо выполнить следующее:

1. Создать вКоммутатор, к нему привязать эти vmnics. **Configuration ⇒ Networking ⇒ add Networking.**
2. В запущившемся мастере первым вопросом будет тип первой группы портов на создаваемом вКоммутаторе. Следует указать тип «Virtual Machine».
3. Выбрать vmnics. Это дело потенциально не простое – о нем пара слов чуть позже.

Кстати, вы можете создать вКоммутатор без единого привязанного физического контроллера. Это пригодится для помещения ВМ в изолированную сеть. Например, для использования NAT или для создания изолированной тестовой сети.

4. Укажите название группы портов. Еще раз повторяюсь, вам будет удобнее сделать его максимально понятным для человека. Также здесь вы можете указать VLAN ID – о VLAN я расскажу чуть позже.

По поводу п. 3 – выбора физического сетевого контроллера, который будет каналом во внешнюю сеть для создаваемого вКоммутатора, – поговорим немного подробнее. Представьте ситуацию, что у вас в сервере несколько vmnics, и они скоммутированы в разные сети (подключены к разным физическим коммутаторам и/или в разные VLAN на физических коммутаторах). И вот вам из, например, шести свободных необходимо выбрать два правильных (рис. 2.9 и 2.10).

Здесь «pSwitch» означает Physical Switch, физический коммутатор.

На рис. 2.9 вы видите сервер с шестью сетевыми картами. Они попарно подключены к разным физическим коммутаторам и работают в разных сетях. Они уже подключены к каким-то коммутаторам виртуальным – или вам как раз сейчас надо это сделать. Предположим, вам надо подключить ВМ в сеть к шлюзу с указанным IP-адресом. Как вы поймете, к какому виртуальному коммутатору и через какие vmnics ее надо подключить?

Нам может помочь **Configuration ⇒ Network Adapters** (рис. 2.10).

Что нам поможет?

Обратите внимание на столбец **Observed IP ranges** в этом окне. Диапазоны IP-адресов в нем и VLAN ID – это подсети и номера VLAN, пакеты из которых получает ESXi на соответствующем интерфейсе. Это не настройка, не рекоменда-

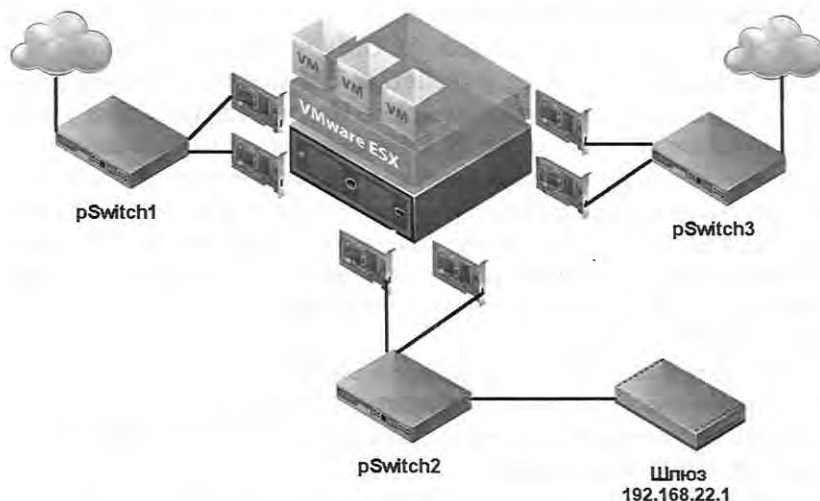


Рис. 2.9. Пример организации сети

esxi-01.vm4ru.local VMware ESXi, 5.0.0, 469512

SummaryVirtual MachinesResource AllocationPerformanceConfigurationLocal Users & GroupsEventsPermissions

Hardware

Health StatusProcessorsMemoryStorageNetworkingStorage Adapters**Network Adapters**Advanced SettingsPower Management

Network Adapters

Device	Speed	Configured	Switch	MAC Address	Observed IP ranges
Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)					
 vmnic7	1000 Full	Negotiate	vSwitch3	00:50:56:a2:6f:96	192.168.60.1-192.168.60.1 (VLAN 60)
 vmnic6	1000 Full	Negotiate	vSwitch3	00:50:56:a2:6f:95	192.168.60.1-192.168.60.1 (VLAN 60)
 vmnic5	1000 Full	Negotiate	vSwitch2	00:50:56:a2:6f:94	192.168.52.1-192.168.52.1 (VLAN 52)
 vmnic4	1000 Full	Negotiate	vSwitch2	00:50:56:a2:6f:93	192.168.52.1-192.168.52.1 (VLAN 52)
 vmnic3	1000 Full	Negotiate	vSwitch1	00:50:56:a2:6f:92	192.168.50.1-192.168.50.1 (VLAN 50)
 vmnic2	1000 Full	Negotiate	vSwitch1	00:50:56:a2:6f:91	192.168.50.1-192.168.50.1 (VLAN 50)
 vmnic1	1000 Full	Negotiate	vSwitch0	00:50:56:a2:1b:92	192.168.22.30-192.168.22.31
 vmnic0	1000 Full	Negotiate	vSwitch0	00:50:56:a2:1b:91	192.168.22.30-192.168.22.31

Рис. 2.10. Информация о физических сетевых контроллерах ESXi

ция – это информационное поле. То есть если вы знаете, что vmnic, который вам сейчас надо подключить, коммутирован в сеть с известной вам подсетью/vlan, то вы сможете сориентироваться по представленной здесь информации.

Итак, глядя на рис. 2.9 и 2.10, к каким vmnic надо подключить эту ВМ? Правильный ответ – vmnic0 и vmnic1. В моем примере они подключены к вКоммутатору с именем vSwitch0. Значит, надо посмотреть, какая группа портов для ВМ существует на этом виртуальном коммутаторе, и именно к ней подключать ВМ.

Второй вариант – если вам известен MAC-адрес искомого физического контроллера, то вы можете сопоставить MAC-адреса и названия физических контроллеров (вида vmnic#) и – как следствие – название vSwitch и групп портов.

Если вы хотите добавить группу портов для ВМ на существующий вКоммутатор, то делается это очень просто:

1. **Configuration** ⇒ **Networking** ⇒ **Properties**, но НЕ в верхней правой части окна, а справа от названия нужного вКоммутатора.

2. На вкладке **Ports** кнопка **Add**. Далее укажите, что вновь добавляемая группа портов – для ВМ, и ее название.

Зачем вам может понадобиться на один вКоммутатор добавлять несколько групп портов для ВМ? Ответ прост – чтобы разные ВМ выходили в сеть через одни и те же физические контроллеры, но при этом с разными настройками. Здесь под настройками понимаются следующие из них:

- ☐ настройки VLAN (самая частая причина создания множества групп портов);
- ☐ настройки Security (безопасности);
- ☐ настройки Traffic Shaping (управления пропускной способностью);
- ☐ настройки NIC Teaming (группировки контроллеров).

Эти настройки могут задаваться на уровне вКоммутатора – тогда они наследуются всеми группами портов. Однако на уровне группы портов любую из настроек можно переопределить. Еще один нюанс – настройка **Number of Ports** (количества портов) существует только для вКоммутатора, группы портов у нас «безразмерные». А настройка VLAN существует только для групп портов, не для коммутатора целиком.

Поговорим про эти настройки более подробно чуть далее.

2.3. Распределенные коммутаторы – vNetwork Distributed Switch, dvSwitch. Настройки

Виртуальные коммутаторы VMware – штука хорошая. Однако нет предела совершенству, и в больших инфраструктурах вам могут быть интересны распределенные виртуальные коммутаторы – vNetwork Distributed Switch, или dvSwitch.

Обратите внимание на то, что настройки распределенных виртуальных коммутаторов описываются в разных разделах:

- ☐ в разделах 2.3.4 и 2.3.5 описаны уникальные настройки dvSwitch;
- ☐ в разделе 2.4 описаны настройки, доступные и для стандартных, и для распределенных виртуальных коммутаторов, это группы настроек «Security», «VLAN», «Traffic shaping» и «NIC Teaming».

Таким образом, если вас интересуют настройки только стандартных виртуальных коммутаторов – см. раздел 2.4, а если распределенных – то разделы 2.3 и 2.4.

2.3.1. Основа понятия «распределенный виртуальный коммутатор VMware»

Идея их заключается в следующем: в случае использования стандартных вКоммутаторов у вас разные (пусть очень часто и идентично настроенные) виртуальные коммутаторы на каждом сервере. Акцент я хочу сделать вот на чем: создать и поддерживать эту, пусть даже идентичную, конфигурацию придется вам.

При необходимости, например, создать еще одну группу портов для еще одного VLAN вам потребуется повторить это простое действие на всех серверах ESXi.

А в случае создания распределенного коммутатора вы получаете один-единственный (логически) коммутатор, существующий сразу на всех серверах ESXi.

Вы настраиваете только этот, логически единый коммутатор. Новую группу портов из примера выше вы создадите один раз, и она появится для всех серверов ESXi, входящих в распределенный коммутатор.

ВМ, даже при миграции с сервера на сервер, остаются не только на том же коммутаторе, но и даже на том же порту распределенного виртуального коммутатора (это называют Network vMotion). Это позволяет проще настраивать политики безопасности, осуществлять мониторинг трафика ВМ, привязываясь даже к отдельному порту в Коммутатора.

Сравнение стандартных и распределенных виртуальных коммутаторов

По сравнению со стандартными виртуальными коммутаторами VMware, distributed vSwitch интересны двумя вещами. Во-первых, распределенные коммутаторы хороши распределенностью – то есть централизованным управлением, упомянутым абзацем выше.

Второе, чем интересны распределенные вКоммутаторы, – они поддерживают больше сетевых функций:

- ❑ Private VLAN – расширение стандарта VLAN, позволяющее дополнительно ограничить видимость (здесь) виртуальных машин внутри одного VLAN;
- ❑ двухсторонний traffic shaping;
- ❑ больше возможностей при работе с конфигурацией VLAN trunk – когда необходимо оставлять тэги vlan в трафике от и к виртуальной машине. На стандартных виртуальных коммутаторах мы можем сделать такую конфигурацию, но подобный порт обязательно участвует во всех vlan вообще. А здесь мы сможем явно указать только те vlan, для которых необходимо настроить vlan trunk;
- ❑ Network IO Control – возможность гибко настраивать минимально и максимально доступные ресурсы пропускной способности для трафика разных типов. Подробности о NIOC см. в главе 6, посвященной распределению ресурсов;
- ❑ Load Based Teaming – балансировка нагрузки между физическими контроллерами одного вКоммутатора в зависимости от нагрузки на каждый контроллер;
- ❑ Link Layer Discovery Protocol (LLDP) – протокол сбора данных о сетевых устройствах. Аналог Cisco Discovery Protocol, но, в отличие от CDP, является независимым от производителя;
- ❑ NetFlow – протокол для сбора статистики трафика;
- ❑ QoS (802.1p) – поддержка тэгов приоритизации трафика.

Еще один плюс – распределенный вКоммутатор доступен в реализациях от VMware и от третьих фирм. Сегодня это Cisco и IBM.

Варианты от третьих фирм приобретаются независимо от vSphere (но подойдет не любая лицензия vSphere, на момент написания – только Enterprise Plus).

На примере распределенного виртуального коммутатора от Cisco – он обладает всеми возможностями, присущими сетевому оборудованию от Cisco. Позволяет добиться того, что все используемые коммутаторы в сети компании созданы одним производителем (здесь имеется в виду Cisco), и сетевые администраторы могут управлять ими точно так же, как коммутаторами физическими, снимая, таким образом, эти задачи с администратора vSphere. Плюс к тому Nexus 1000V обладает большим функционалом. Однако рассмотрение настроек и возможностей этого решения в данной книге описано не будет.

Здесь будет описан распределенный виртуальный коммутатор от VMware. Возможность им пользоваться появляется после активации лицензии, дающей на него право, – сегодня это vSphere Enterprise Plus (и ознакомительный 60-дневный период после установки vSphere, Evaluation-лицензия).

Как я уже сказал, dvSwitch – объект скорее логический, как таковой существующий только для vCenter. Мы создали один распределенный вКоммутатор, но на каждом сервере, для которых он существует, автоматически создаются стандартные вКоммутаторы, скрытые от нас (рис. 2.11). Таким образом, де-факто распределенный коммутатор является шаблоном настроек. Мы создаем его и указываем настройки в vCenter – это control plane по документации VMware, то есть уровень управления. А после включения в созданный распределенный коммутатор серверов ESXi vCenter создает на них стандартные коммутаторы, скрытые от нас. По документации VMware это IO Plane, прикладной уровень. За всю работу отвечают

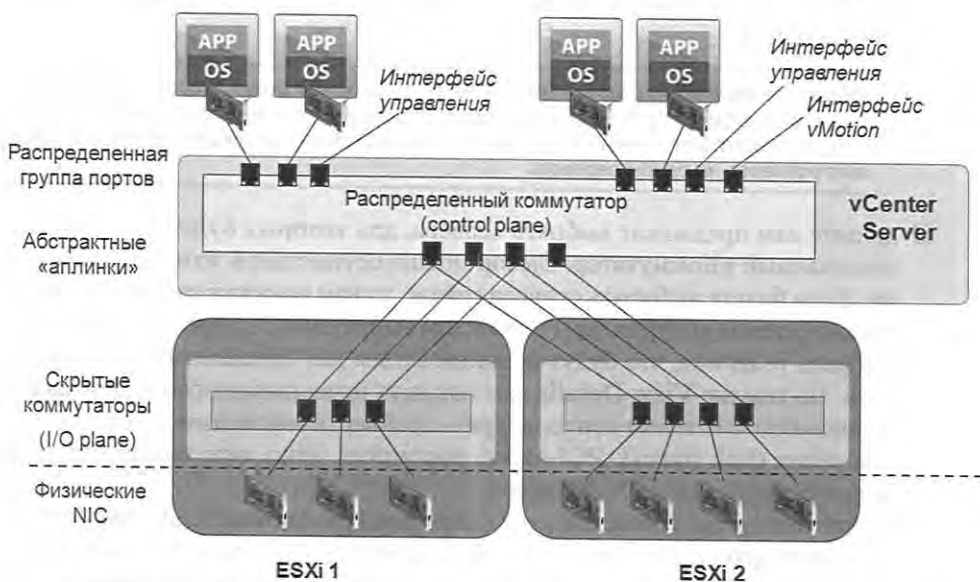


Рис. 2.11. Иллюстрация сути распределенного виртуального коммутатора
Источник: VMware

скрытые коммутаторы на серверах ESXi, все управление осуществляется только через vCenter.

Минусом такой схемы распределенного виртуального коммутатора VMware является возросшая сложность, в частности зависимость от vCenter. vCenter является управляющим элементом для dvSwitch, и при недоступности vCenter у администратора нет возможности менять настройки распределенного коммутатора и даже переключать виртуальные машины на другие группы портов. Однако даже при недоступности vCenter сеть будет продолжать работать – ведь за техническую сторону вопроса отвечают скрытые от нас коммутаторы на каждом ESXi, теряется только возможность изменения настроек.

Подключитесь клиентом vSphere к vCenter. Предполагается, что в иерархии vCenter уже создан объект Datacenter, уже добавлены сервера. Распределенный виртуальный коммутатор создается для объекта Datacenter, и существовать для серверов из разных Datacenter один dvSwitch не может.

Пройдите **Home** ⇒ **Networking**. В контекстном меню объекта Datacenter выберите пункт **New vNetwork Distributed Switch**.

В запущившемся мастере нас спросят:

- ☐ **Version** – версию распределенного коммутатора. Выбор старой версии, очевидно, имеет смысл тогда, когда в этом vКоммутаторе должны участвовать сервера ESX/ESXi версии 4.
- ☐ **Name** – имя создаваемого vКоммутатора. Влияет только на удобство;
- ☐ **Number of dvUplink ports** – максимальное количество подключений ко внешней сети – привязанных vmnics – на каждом сервере. Число портов для аплинков в шаблоне настроек, которым по сути является распределенный vКоммутатор;

Обратите внимание: один такой vКоммутатор может существовать для серверов с разной конфигурацией, с разным количеством доступных сетевых контроллеров – а здесь мы ограничиваем максимальное число используемых для данного dvSwitch контроллеров на одном сервере.

- ☐ далее вам предложат выбрать сервера, для которых будет существовать создаваемый vКоммутатор. Выбор можно осуществить или изменить позже. Если будете выбирать сервера сейчас, то вам покажут свободные физические сетевые контроллеры на каждом выбранном сервере – и вы сможете выбрать те из них, что будут использоваться для создаваемого vКоммутатора. По ссылке **View Details** вам покажут исчерпывающую информацию о физическом сетевом контроллере – драйвер, производитель, статус подключения (link status), PCI-адрес, доступные через него подсети IP, – не покажут здесь только MAC-адрес;
- ☐ на последнем шаге можно будет снять флажок **Automatically create a default port group**. Если он стоит, то автоматически будет создана группа портов со 128 портами и именем по умолчанию. Имя по умолчанию мало информативно, поэтому я рекомендую этот флажок снимать и создавать группу портов самостоятельно, после создания vКоммутатора.

Обратите внимание. У стандартных вКоммутаторов число портов настраивается для них самих, группы портов «безразмерные». У распределенных вКоммутаторов наоборот.

Вы можете создать до 248 распределенных виртуальных коммутаторов для сервера. На одном сервере может быть до 4096 портов стандартных и распределенных вКоммутаторов. Это означает, что если не задавать заведомо огромных значений числа портов, то с ограничениями с этой стороны вы не столкнетесь.

Когда вы создали dvSwitch, то на странице **Home** ⇒ **Networking** вы видите картинку примерно как на рис. 2.12:

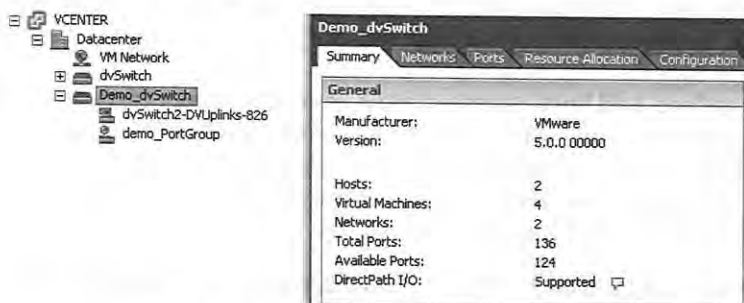


Рис. 2.12. Свежесозданный dvSwitch

Здесь «Demo_dvSwitch» – это сам объект «распределенный виртуальный коммутатор». Объект «dvSwitch2-DVUplinks-826» – это группа портов, в которую объединены его подключения ко внешним сетям. То есть те порты вКоммутатора, к которым подключены физические сетевые контроллеры серверов, на которых этот dvSwitch существует. Нужен этот объект для получения информации о внешних подключениях – обратите внимание на столбцы на вкладке **Ports** для данной группы портов. Число 826 в названии – это случайным образом сгенерированное число, для уникальности имени этого объекта. Такая группа портов создается всегда, создается автоматически. Скорее всего, вы никогда не будете изменять какие-либо параметры этой созданной в технических целях группы портов.

Ну и «Demo_PortGroup» – созданная вручную группа портов, туда можно подключать ВМ, а также виртуальные сетевые контроллеры VMkernel.

Обратите внимание. При использовании стандартных виртуальных коммутаторов невозможно поместить в одну группу портов виртуальные машины и интерфейс VMkernel. На распределенных виртуальных коммутаторах VMware это возможно, на dvSwitch в одной группе портов могут сосуществовать и ВМ, и интерфейсы VMkernel в любых сочетаниях. Однако такого объединения следует избегать из организационных соображений. То есть следует создавать отдельные группы портов для ВМ и отдельные группы портов для интерфейсов VMkernel. Более того, если вы на одном и том же dvSwitch создаете с каждого сервера по несколько vmk (к примеру, для трафика управления и трафика vMotion), то для vmk разных задач следует создавать свои группы портов.

Кстати говоря, объект «VM Network» на рис. 2.12 – это группа портов на стандартных виртуальных коммутаторах серверов этого vCenter.

2.3.2. Добавление сервера в dvSwitch, настройки подключения vnic

У вас есть сервера, на них создан распределенный коммутатор. Появился еще один сервер, необходимо задействовать и его. Идем **Home** ⇒ **Networking** ⇒ вызываем контекстное меню нужного dvSwitch ⇒ **Add Host**. В запустившемся мастере вам покажут список серверов этого Datacenter, для которых выбранный dvSwitch не существует, и их vnic. Выберите нужный сервер и те из его vnic, что хотите задействовать под этот dvSwitch (последнее можно сделать и потом). Если вы выберете vnic, которая уже используется, вам покажут предупреждение, что выбранный контроллер будет отсоединен от старого vКоммутатора и подключен к данному dvSwitch.

После выбора **Add host** в контекстном меню вам может быть показано не окно выбора сервера, а мастер добавления сервера в vCenter (поле для ввода адреса сервера и учетных данных). Такое происходит в том случае, если все существующие сервера в Datacenter уже подключены к этому dvSwitch – вот вам и предлагают добавить в Datacenter какой-то новый сервер.

Нюансы задействования внешних подключений (Uplinks) dvSwitch

У dvSwitch есть несколько портов для внешних подключений (dvUplink) – их число мы указываем при создании. Например, в настройках с рис. 2.13 dvSwitch указано, что у него максимум три внешних подключения. Еще раз напомним, что

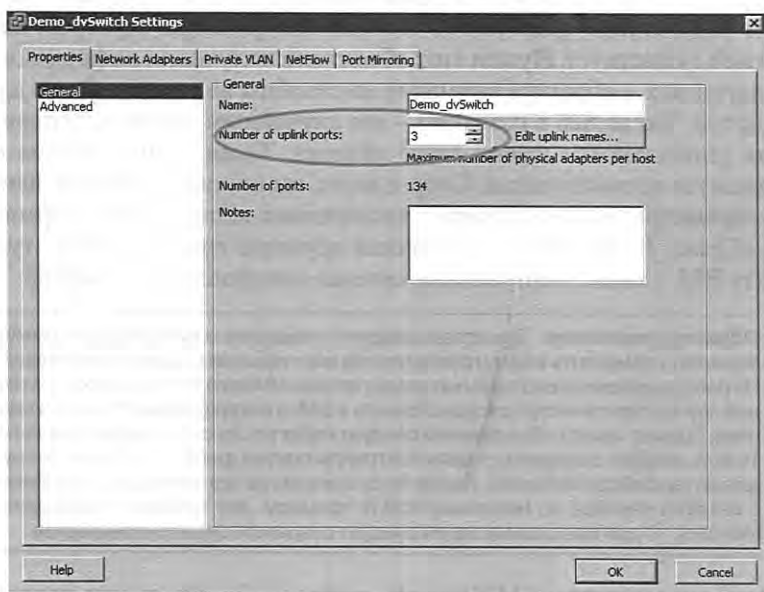


Рис. 2.13. Настройки dvSwitch

это означает – «до трех подключенных физических сетевых контроллеров на каждом сервере ESXi, для которого существует этот распределенный виртуальный коммутатор».

По умолчанию они называются «dvUplink1», «dvUplink2» и т. д. Увидеть их можно, пройдя **Configuration** ⇒ **Networking** ⇒ кнопка **Distributed Virtual Switch** (рис. 2.14). Если здесь развернуть список дочерних объектов для dvUplink (нажав +), мы увидим названия сетевых контроллеров серверов (вида vmnic#), подключенных к этому dvSwitch.

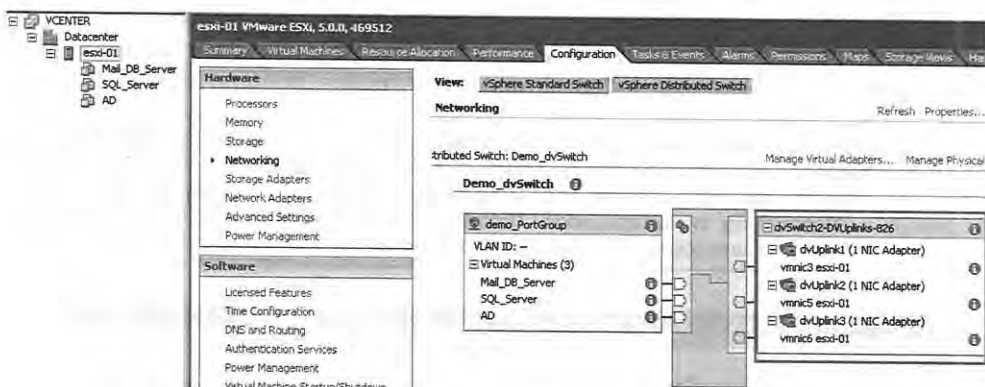


Рис. 2.14. Внешние подключения распределенного коммутатора на конкретном сервере

В данном окне мы видим ту часть и те настройки распределенного виртуального коммутатора, что существуют для данного сервера.

Увидеть же все физические сетевые контроллеры, подключенные к этому dvSwitch на каждом из серверов, мы можем, пройдя **Home** ⇒ **Networking** ⇒ dvSwitch ⇒ вкладка **Configuration**.

Эти объекты можно назвать «абстрактными подключениями», или «абстрактными аплинками», или «портами под аплинки». То есть три dvUplink на скриншоте выше – это три порта под аплинки в шаблоне настроек. Проиллюстрирую их суть на примере.

Есть распределенный vКоммутатор, на нем несколько групп портов и несколько внешних подключений. Для какой-то группы портов мы указали использовать только первое подключение (dvUplink1). Отлично – виртуальные машины, подключенные к этой группе портов, пользуются только первым внешним подключением этого коммутатора. Но на каждом сервере этому «абстрактному» подключению номер один соответствует какой-то конкретный физический сетевой интерфейс. Вот откуда берется нужда в абстракции. Получается, что одному абстрактному dvUplink# соответствует столько физических контроллеров, для скольких серверов существует распределенный коммутатор.

Поменять названия «dvUplink» на произвольные можно в свойствах dvSwitch, вкладка **Properties** ⇒ **General** ⇒ **Edit dvUplink port**, изменять эти названия име-

ет смысл лишь для нашего удобства. Например, в наших серверах установлено по два сетевых контроллера, один из которых 10 Гбит, а второй – гигабитный. Правильно будет 10 Гбит контроллеры с каждого сервера подключать в один и тот же «абстрактный порт» распределенного коммутатора. Если эти абстрактные порты будут называться не «dvUplink1» и «dvUplink1», а «dvUplink_10Gb» и «dvUplink_1Gb», то нам будет проще не ошибиться.

Если какой-то dvUplink# уже занят хотя бы одним физическим сетевым контроллером – вы увидите его имя (вида vmnic#) под именем порта (вида dvUplink#). Здесь имеются в виду настройки dvSwitch целиком: **Home** ⇒ **Networking** ⇒ вкладка **Configuration** – для распределенного виртуального коммутатора.

Далее на каждом сервере мы можем указать, какой же из физических сетевых контроллеров этого сервера является каким из внешних подключений dvSwitch.

Обратите внимание: эта настройка делается именно на уровне каждого сервера, а не распределенного коммутатора. Поэтому, для того чтобы увидеть то, о чем я говорю, надо пройти **Home** ⇒ **Hosts and Clusters** ⇒ настраиваемый сервер ⇒ **Configuration** ⇒ **Networking** ⇒ нажать кнопку **Distributed Virtual Switch** ⇒ для настраиваемого dvSwitch нажать ссылку **Manage Physical Adapter**. Откроется окно, как на рис. 2.15.

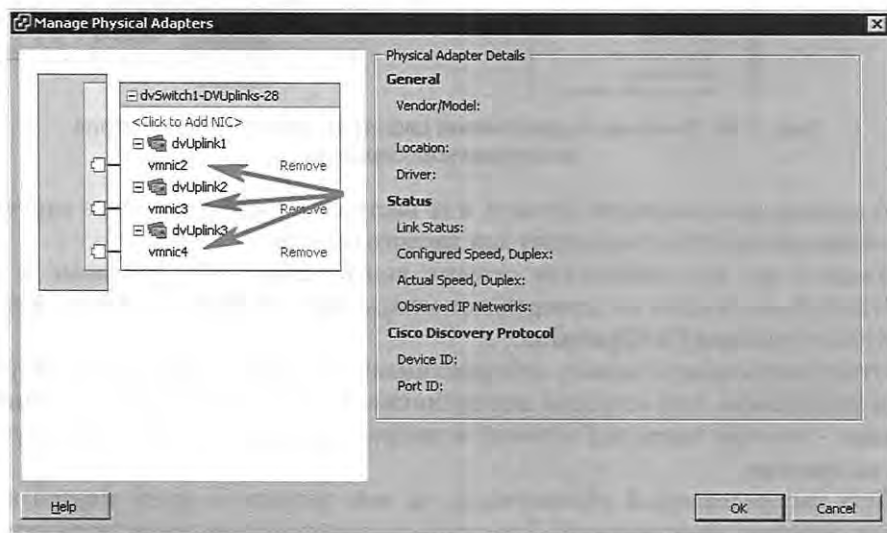


Рис. 2.15. Настройки привязки физических контроллеров сервера к dvSwitch

vmnic# на этом рисунке – это те vmnic, которые вы видите для сервера на странице **Configuration** ⇒ **Network Adapters**. Нажатие **Remove** удалит vmnic из этого распределенного виртуального коммутатора, нажатие **Add NIC** позволит выбрать, какой из свободных vmnic добавить и каким из абстрактных внешних подключений сделать.

Щелкнув на уже подключенный vmnic, вы увидите информацию о нем и сможете настроить скорость и дуплекс (рис. 2.16).

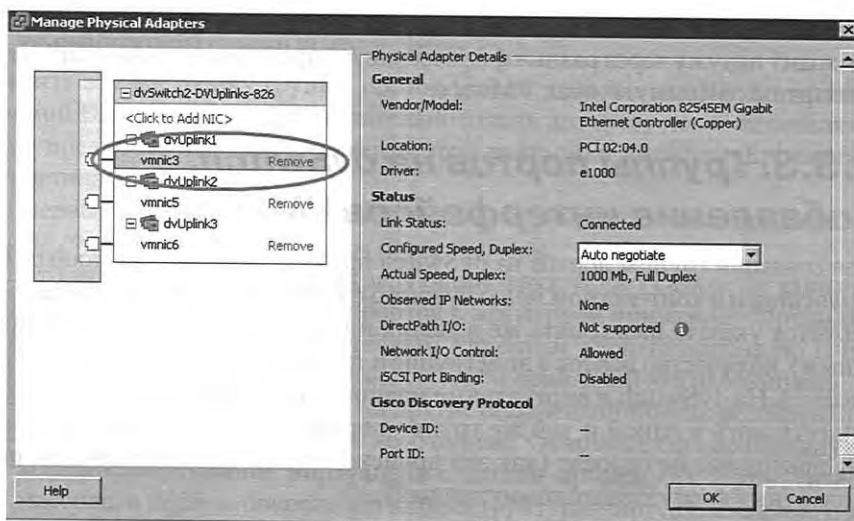


Рис. 2.16. Окно настроек vmnic

Какой vmnic к какому порту (dvUplink) подключен, может оказаться принципиально, потому что в свойствах групп портов на dvSwitch вы можете выбирать, какое из абстрактных внешних подключений dvSwitch является активным, запасным или не используемым для данной группы портов (рис. 2.17).

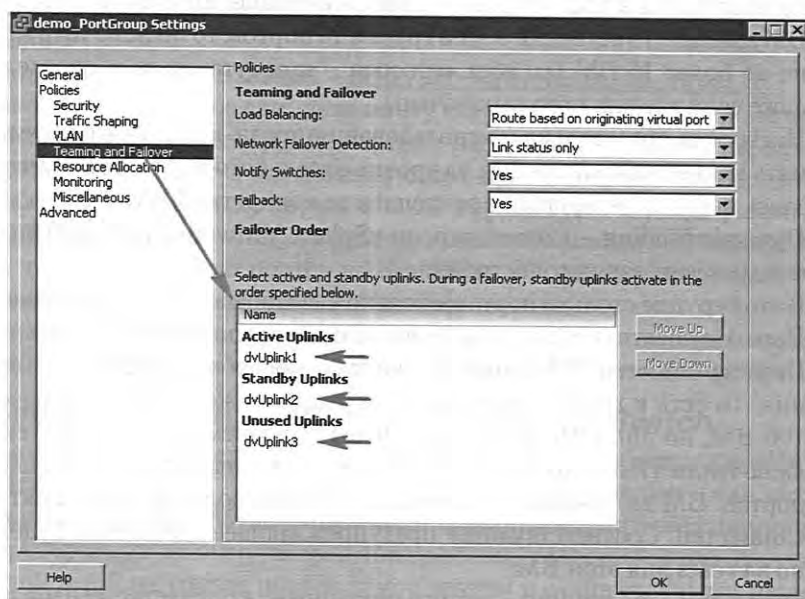


Рис. 2.17. Настройки использования внешних подключений для группы портов на dvSwitch

То есть на одном dvSwitch мы можем сделать несколько групп портов, чей трафик выходит наружу через разные физические контроллеры, – точно так же, как и на стандартных вКоммутаторах VMware.

2.3.3. Группы портов на dvSwitch, добавление интерфейсов VMkernel

Для создания группы портов на dvSwitch вручную необходимо пройти **Home** ⇒ **Networking** и в контекстном меню нужного dvSwitch выбрать **New Port Group**. Потребуется указать имя (опять же в ваших интересах сделать его максимально понятным), количество портов в этой группе и тип VLAN (про VLAN для dvSwitch чуть позже). На dvSwitch и виртуальные машины, и интерфейсы гипервизора могут существовать в одной и той же группе портов, поэтому о типе создаваемой группы портов нас не спросят (как это происходит при создании группы портов на стандартных виртуальных коммутаторах).

Обратите внимание на то, что если вы зайдете в настройки уже созданной группы портов, то количество настроек будет больше. В частности, мы можем настроить **Port binding**. Эта настройка связана с количеством портов в группе портов распределенного вКоммутатора. Варианты этой настройки:

- ❑ **Static binding** – в этой группе портов столько портов, сколько указано настройкой Number of ports. Порты существуют вне зависимости от того, подключены ли к ним ВМ. Порт закрепляется за ВМ в тот момент, когда она подключается к этой группе портов, – вне зависимости от того, включена ВМ или нет. То есть если в этой группе 16 портов, то подключить к ней можно не более 16 ВМ (со всех серверов – коммутатор-то распределенный), даже если все они пока выключены.

Настройка обязательна к использованию тогда, когда вам требуется изменять какие-либо настройки на уровне отдельного порта, потому что только здесь порт гарантированно закреплен за конкретной ВМ;

- ❑ **Dynamic binding** – в пятой версии vSphere VMware настоятельно не рекомендует этот вариант настройки.

В этой группе столько портов, сколько указано настройкой Number of ports. Порты существуют вне зависимости от того, подключены ли к ним ВМ. Порт занимает ВМ в момент запуска и освобождается после выключения. То есть к группе портов с 16 портами может быть подключено хоть 100 ВМ, но лишь 16 из них включены одновременно. К сожалению, при включении 17-ой ВМ вы не получите сообщения о том, что нет свободных портов. ВМ включится, но с ее сетевого контроллера будет снят флажок **Connected**. Соответствующее предупреждающее сообщение будет занесено в events для этой ВМ;

- ❑ **Ephemeral – no binding** – нет ограничения на количество портов для группы. Порт создается и закрепляется за ВМ в момент включения и существует, только пока ВМ включена. Но таких групп портов не может быть боль-

ше 256 на один ESXi (почти всегда это означает – на все распределенные вКоммутаторы одного vCenter Server).

Менять эту настройку группы портов можно, только если нет ни одной подключенной ВМ, то есть крайне желательно делать это сразу после ее создания.

Мне предпочтительным вариантом этой настройки видится «Ephemeral» по двум причинам.

Во-первых, эфемерная привязка портов – это гарантия того, что количество портов не станет для нас внезапным ограничением.

Во-вторых, имеет место быть следующий факт: если vCenter по каким-либо причинам недоступен, то к группе портов с эфемерной привязкой портов можно подключить ВМ (будучи подключенным клиентом vSphere напрямую к ESXi). В то время как со статичной или динамической привязкой это невозможно.

Однако сама VMware сегодня дает другие рекомендации: использовать Ephemeral binding только там, где это явно необходимо. Основные сценарии следующие:

- ❑ когда развертывание виртуальных машин происходит автоматически и массово, в первую очередь это инфраструктуры виртуальных рабочих столов и облачные решения. Попросту говоря, если поверх vSphere работают такие продукты, как VMware View и VMware vCloud Director;
- ❑ на случай подготовки к проблемам. Если вы опасаетесь ситуаций с недоступностью vCenter и стоит задача в этих ситуациях не терять возможность подключать к сети виртуальные машины, то без vCenter это возможно лишь при эфемерной привязке портов.

Главная, по моему мнению, причина такой рекомендации – операции вроде включения ВМ вызывают за собой создание новых портов, что замедляет исходные операции.

См. подробности в базе знаний – <http://kb.vmware.com/kb/1022312>.

Однако обратите особое внимание на эту статью базы знаний – кроме прочего, в ней описан способ сделать группу портов с настройкой static binding автоматически расширяемой. То есть мы можем указать правило вроде «всегда держи два порта свободными» – и число портов в такой группе будет автоматически увеличиваться. Таким образом, если мы планировали использовать ephemeral binding для того, чтобы количество портов не оказалось для нас ограничением, то этой проблемы можно избежать и для static binding.

Про прочие настройки, доступные для групп портов и отдельных портов распределенных виртуальных коммутаторов VMware, см. следующий раздел.

Добавление интерфейса VMkernel на dvSwitch

Разумеется, с распределенным коммутатором могут работать как ВМ, так и интерфейсы самого ESXi – интерфейсы VMkernel. Определите (или создайте) группу портов, в которую будет подключен созданный адаптер. Напоминаю, что для распределенных вКоммутаторов интерфейсы VMkernel могут сосуществовать в одной и той же группе портов друг с другом и с виртуальными машинами, в любых комбинациях. Но из организационных соображений я бы так не поступал и рекомендую всегда создавать для интерфейсов VMkernel под какую-то задачу

свою группу портов, отдельную от ВМ и от интерфейсов VMkernel других задач. Это упростит понимание конфигурации в Коммутатора и групп портов и упростит настройки (в первую очередь настройки VLAN).

Интерфейсы VMkernel – это объекты конкретного сервера. Поэтому для управления ими пройдите на вкладку **Configuration** для выбранного сервера ⇒ **Networking** ⇒ кнопка **Distributed Virtual Switch** ⇒ **Manage Virtual Adapters** для того dvSwitch, где вам требуется создать или изменить настройки интерфейса VMkernel.

Вам покажут список существующих интерфейсов VMkernel этого сервера на этом dvSwitch. В верхней левой части открывшегося окна активна ссылка **Add**. После ее нажатия запустится мастер с вопросами:

- ☐ создать новый интерфейс или мигрировать существующий с обычного в Коммутатора. Здесь мы говорим про создание нового;
- ☐ на втором шаге мы указываем, что хотим создать интерфейс для VMkernel (ранее для ESX-версии была альтернатива – кроме интерфейсов VMkernel, были так называемые интерфейсы Service Console – и данный шаг, бессмысленный сейчас, остался в наследство);
- ☐ **Select port group** – выберите ранее созданную группу портов, в которую подключится создаваемый интерфейс.

Select Standalone port – в какой отдельный порт подключить создаваемый интерфейс. Используется, лишь если вам необходимо подключить ВМ к строго определенному порту. Строго определенный порт может быть важен, если вы делаете какие-либо сетевые настройки на уровне отдельного порта, не группы портов. Обычно этого не требуется.

В случае добавления интерфейса VMkernel здесь вы можете указать, что этот интерфейс можно использовать для vMotion (флажок **Use this virtual adapter for vMotion**), для Fault Tolerance (**Use this virtual adapter for Fault Tolerance logging**) и для управления (**use this virtual adapter for management traffic**);

- ☐ **IP Settings** – укажите настройки IP для создаваемого интерфейса;
- ☐ все.

2.3.4. Уникальные настройки распределенных виртуальных коммутаторов

Если вы зайдете в **Home** ⇒ **Networking** ⇒ контекстное меню dvSwitch, выберете **Edit Settings**, то сможете изменить настройки распределенного виртуального коммутатора.

Настройки **General**:

- ☐ **Name** – имя dvSwitch;
- ☐ **Number of dvUplink ports** – максимальное количество физических сетевых контроллеров, которое может быть подключено к этому коммутатору *на одном сервере*;
- ☐ **Edit dvUplink port** – нажав эту кнопку, можно изменить название абстрактного внешнего подключения. Например, имя по умолчанию «dvUplink1»

заменим на более информативное «Primary_vMotion_uplink». Если все аплинки для вас равнозначны, то изменение этого имени смысла не имеет;

- ❑ **Notes** – произвольные примечания.

Настройки **Advanced**:

- ❑ **Maximum MTU** – Maximum Transmission Unit, размер поля с данными в пакете IP. Используется для включения/выключения Jumbo Frames. Эта функция позволяет снизить долю технического трафика в IP-сетях, то есть значительно снизить накладные расходы на передачу данных по сети. Может использоваться для iSCSI/NFS-трафика, трафика vMotion, для VM. Подробности про Jumbo Frames и их настройку см. далее.
- ❑ **Discovery Protocol** – здесь настраиваются протоколы CDP (Cisco Discovery Protocol) или LLDP (Link Layer Discovery Protocol). См. раздел 2.4.6.
- ❑ Вкладка **Network Adapters** чисто информационная, изменить настройки отсюда мы не можем.
- ❑ На вкладке **Private VLAN** мы можем настроить Private VLAN. В чем суть этой функции, см. далее.

NetFlow

Вкладка **NetFlow** позволяет настроить использование одноименного протокола. Поддержка протокола NetFlow (на момент написания поддерживается версия 5) позволяет распределенному коммутатору VMware пересылать на коллектор, «NetFlow collector», статистику по трафику.

Информация, полученная при помощи этого инструмента, позволяет ответить на такие вопросы, как:

- ❑ какой трафик идет через конкретный порт;
- ❑ по каким протоколам и портам;
- ❑ в каком объеме;
- ❑ какие тэги к нему применены;
- ❑ и некоторые другие (есть зависимость от версии протокола и конкретных средств по работе с ним).

Как правило, просмотр сразу статистики трафика требует меньше ресурсов, чем перехват всего трафика сетевым снифером и затем анализом этого трафика.

В рамках протокола NetFlow выделяются три типа устройств:

- ❑ зонд – тот, кто отчитывается. В нашем случае это распределенный коммутатор;
- ❑ коллектор, Netflow collector – система, на которую отчитываются зонды;
- ❑ анализатор – многие продукты для работы с Netflow предоставляют более или менее мощные средства анализа собранных данных.

В некоторых случаях анализатор поможет сделать определенные выводы для решения некоторых задач. Например, можно изменять настройки качества сервиса в сторону увеличения доли одного трафика и уменьшения другого, или можно поменять настройки протоколов маршрутизации для изменения маршрута прохождения трафика при перегрузках – и получать статистику, как меняется поведение сети/устройств/сервисов вследствие таких изменений.

Данный протокол был разработан Cisco, но в настоящий момент является общим стандартом де-факто.

Когда в вашей сети есть коллектор Netflow, для отправки на него информации с распределенного коммутатора следует выполнить следующие настройки в свойствах распределенного коммутатора на вкладке **NetFlow**:

- ☐ **Collector Settings** – укажите IP-адрес коллектора и сетевой порт;
- ☐ **VDS IP Address** – IP-адрес, от имени которого будет отправляться информация на коллектор. Если не указать, то отправителем будет отображаться каждый сервер ESXi независимо (информация будет отправляться с управляющего интерфейса каждого хоста).

Последний шаг – разрешить передачу статистики для конкретной группы портов (или даже конкретного порта) распределенного коммутатора. В свойствах группы портов на строке **Monitoring** единственная настройка **Netflow status**. При значении «Enabled» статистика этой группы портов/порта будет отдаваться на коллектор.

Если через распределенный коммутатор проходит большое количество трафика, то обработка статистики и отправка ее на NetFlow-коллектор может создать заметную нагрузку на процессоры хостов и сеть управления. Для минимизации этого эффекта мы можем увеличить значение настройки **Sampling rate**. Значение по умолчанию, «0», означает что обрабатывается каждый пакет. Указав значение «2», мы добьемся того, что обрабатываться будет каждый второй пакет, «5» – один из пяти и т. д.

Настройка **Active flow export timeout** указывает время, в течение которого должен просуществовать «поток», «сессия» трафика, чтобы статистика о нем была отправлена на коллектор. По умолчанию – 60 секунд. Уменьшаем это значение в случаях, когда нас интересует более быстрое получение информации о новых «потоках».

Флажок **Process internal flows only** определяет, будет ли пересылаться информация только о внутреннем трафике распределенного коммутатора или о трафике с/на аплинки тоже. Если для физических коммутаторов также настроен сбор данных на NetFlow-коллектор, то имеет смысл с dvSwitch снимать данные только о трафике, замыкающемся внутри него.

Port Mirroring

Эта функция пригодится для перехвата и анализа сетевого трафика, которым обмениваются подключенные к распределенному коммутатору виртуальные машины. Трафик будет пересылаться на указанный в настройках порт.

К сожалению, на момент написания актуально следующее неудобство: если получателем зеркалируемого трафика выступала виртуальная машина, то на нее поступал только тот трафик, что поступал на тот ESXi, где работала эта ВМ. То есть если есть некий трафик между ВМ1 и ВМ2 с ESXi1 и мы пытаемся зеркалировать этот трафик на ВМ3 – то ВМ3 не получит ничего, если работает на другом ESXi.

Мы можем настроить зеркалирование трафика во внешний порт (аплинк) – тогда каждый ESXi будет зеркалировать трафик в свой аплинк в указанном порту

dvUplink. Если физические коммутаторы будут настроены на дальнейшее зеркалирование этого трафика, то мы сможем получить его на какой-то внешней системе.

Еще раз выделю важный момент: для конкретной ВМ трафик будет зеркалироваться или в порт другой ВМ – но эта другая ВМ обязательно должна работать на том же сервере, или в аплинк этого ESXi – и предполагается, что физический коммутатор передаст этот трафик дальше, на систему анализа. Однако если ВМ мигрирует на другой сервер – ее трафик начнет зеркалироваться уже в аплинк нового ESXi. Таким образом, физический коммутатор должен передавать «куда надо» трафик с порта каждого ESXi, на котором могут оказаться интересующие нас ВМ.

Таким образом, если перед вами встала задача перехватить трафик всех или только некоторых ВМ на распределенном коммутаторе, то сделать это несложно (напомню, заработает это для ВМ на одном ESXi, если зеркалировать собираемся на какую-то из ВМ). Алгоритм настройки начинается с того, что надо определиться с тем, на какой сервер следует передавать зеркалируемый трафик. Если этим сервером будет виртуальная машина – подключите ее сетевой интерфейс к распределенному коммутатору и запишите номер порта, к которому она оказалась подключена. Номер порта см. на вкладке **Ports** группы портов.

Если вы хотите передавать трафик на физический сервер – следует понять, на какой физический интерфейс надо передавать трафик. Правильнее сказать – интерфейсы, так как каждый сервер ESXi будет передавать трафик своей части распределенного коммутатора на один из своих аплинков. Для распределенного коммутатора есть такое понятие, как «абстрактный аплинк», – и мы выбираем именно его как источник для зеркалирования трафика на физический интерфейс. Для иллюстрации: пройдите **Home** ⇒ **Networking** ⇒ распределенный коммутатор ⇒ вкладка **Configuration**. Справа от коммутатора вы увидите порты под аплинки с именами по умолчанию вида dvUplink#. Раскрыв + для выбранного dvUplink, вы увидите перечисление физических сетевых контроллеров для каждого сервера, которые соответствуют этому абстрактному аплинку. Каждый сервер ESXi будет зеркалировать трафик на свой аплинк.

Затем вам следует определиться с тем, чей трафик следует зеркалировать. Вам нужны номера портов интересующих вас виртуальных машин. Опять же вам поможет вкладка **Ports** для групп портов этого распределенного коммутатора.

Далее в свойствах распределенного виртуального коммутатора переходите на вкладку **Port Mirroring** и добавляйте правило зеркалирования кнопкой **Add**. Запустится мастер:

1. **General Properties**. Здесь вы должны указать имя правила и описание. Также есть возможность поставить несколько флажков:

- **Allow normal IO on destination ports** – если этот флажок не стоит, то порт, указанный как порт, куда трафик должен зеркалироваться, не будет обрабатывать другой трафик, кроме зеркалируемого. Проще говоря, флажок должен стоять, если этот интерфейс ВМ-получателя зеркалируемого трафика надо задействовать под обычный доступ к сети;
- **Encapsulation VLAN** – если установлен этот флажок, то весь зеркалируемый трафик тэгируется указанным vlan id. Если трафик уже с тэгами

vlan, то без флажка **Preserve original VLAN** исходные vlan id будут заменяться указанным здесь, если флажок стоит – то будет происходить двойное тэгирование;

- **Mirrored packet length** – эта настройка отвечает за возможность перед зеркаливанием обрезать кадр под указанную длину (если машине, принимающей зеркалированный трафик, не нужно поле данных, а нужны только заголовки, будет экономиться полоса пропускания).
2. **Specify Sources.** На этом шаге мастера вам следует указать номера и/или диапазоны портов распределенного коммутатора, чей трафик следует зеркалировать. В выпадающем меню **Traffic direction** можно указать, следует ли зеркалировать трафик как входящий, так и исходящий или только трафик какого-то одного направления.
 3. **Specify Destinations.** Здесь мы указываем, куда следует зеркалировать трафик. Получателем может выступать порт распределенного коммутатора (или несколько, если вдруг надо) или аплинк. Таким образом, распределенный коммутатор может зеркалировать свой трафик на порт коммутатора физического, это значит, что система, на которую мы хотим перенаправить трафик, может быть не только виртуальной машиной этого распределенного коммутатора.
Уточню, что зеркалировать трафик на физический аплинк будет каждый сервер независимо – и здесь мы укажем тот «порт под аплинки», dvUplink, то есть тот аплинк с каждого сервера, куда трафик будет зеркалироваться.
 4. **Ready to Complete.** На этом шаге следует поставить флажок **Enable this port mirroring session**, если созданное правило должно начать работать немедленно.

Когда нужда в правиле зеркалирования трафика отпадет, его можно удалить или отключить.

2.3.5. Уникальные настройки портов *dvSwitch: Miscellaneous и Advanced*

Здесь поговорим о том, какие бывают настройки для портов распределенных коммутаторов. Обратите внимание на то, что на dvSwitch настройки могут применяться как на уровне групп портов, так и индивидуально к порту. Чтобы настройка применилась к группе портов, вам следует выбрать пункт **Edit Settings** контекстного меню группы портов. Для настройки индивидуального порта следует выбрать пункт **Edit Settings** контекстного меню порта на вкладке **Ports**. Однако по умолчанию запрещено менять большинство настроек на уровне отдельного порта, разрешить это можно в настройках группы портов ⇒ **Advanced** ⇒ **Allow override of port policies** и **Edit override settings**.

Многие из настроек одинаковы для виртуальных коммутаторов обоих типов. Одинаковые настройки здесь я лишь упомяну, подробному их описанию посвящен следующий подраздел.

Security – эта настройка одинакова для стандартных и распределенных вКоммутаторов. Подробнее о ней далее.

Traffic shaping – эта настройка почти одинакова для стандартных и распределенных вКоммутаторов. Отличие в том, что на dvSwitch данный механизм работает и для исходящего (Egress), и для входящего (Ingress) трафиков. На стандартных вКоммутаторах – только для исходящего.

VLAN – на dvSwitch функционал работы с VLAN немного расширен по сравнению со стандартными вКоммутаторами. Основное отличие – поддерживаются Private VLAN. Немного отличаются сами окна настройки – все подробности далее.

Teaming and Failover – эта настройка одинакова для стандартных и распределенных вКоммутаторов. Но распределенные коммутаторы имеют на один алгоритм балансировки нагрузки больше.

Resource Allocation – уникальная настройка для dvSwitch. Текущая версия распределенного виртуального коммутатора позволяет настраивать распределение полосы пропускания между виртуальными машинами и другими типами трафика ESXi (например, vMotion). Это называется Network IO Control, NIOC. На вкладке **Resource allocation** для распределенного коммутатора можно создать **Network Resource Pool** – это логический объект с собственным набором настроек распределения ресурсов. Это **Shares** (Доля), **Limit** (Жесткое ограничение сверху). Кроме того, в распределенных коммутаторах пятой версии vSphere появилась поддержка тэгов приоритизации трафика (QoS, IEEE 802.1p). Тэг приоритета также указывается для пула ресурсов сети.

Так вот, на вкладке **Resource allocation** для dvSwitch мы определяем пулы сетевых ресурсов и их настройки, а в строке настроек **Resource allocation** для группы портов распределенного коммутатора мы выбираем, к какому из ранее созданных пулов сетевых ресурсов принадлежит данная группа портов.

Подробности про NIOC будут приведены в главе 6, посвященной распределению ресурсов.

Monitoring – уникальная настройка для dvSwitch. Если на уровне распределенного коммутатора настроен протокол NetFlow, то в данной настройке группы портов этого коммутатора мы включаем и выключаем отправку статистики трафика этой группы портов по протоколу NetFlow.

Miscellaneous – уникальная настройка для dvSwitch. В этом пункте мы можем заблокировать все или один порт в зависимости от того, делаем ли мы эту настройку в свойствах группы портов или порта.

Advanced – здесь можем настроить:

- ☐ **Override port policies** – можно ли переназначать вышеуказанные настройки на уровне портов. Если флажок стоит, то **Home** ⇒ **Inventory** ⇒ **Networking** ⇒ выделяем группу портов на dvSwitch и переходим на вкладку **Ports**. Видим порты с указанием номеров и того, чьи виртуальные контроллеры подключены к тому или иному порту. Из контекстного меню порта вызываем пункт **Edit Settings** и редактируем настройки, которые разрешено изменять соответственно настройке под кнопкой **Edit Override Settings**;

- ❑ **Configure reset at disconnect** – когда ВМ отключается от распределенного коммутатора или подключается на другой порт, то порт, к которому она была подключена, сбрасывает настройки на настройки по умолчанию. Данная функция нужна, только если настройка **Override port policies** разрешает изменение каких-то настроек на уровне отдельного порта и вы этим пользуетесь.

2.3.6. Миграция со стандартных виртуальных коммутаторов на распределенные

Если у вас уже есть сеть на стандартных виртуальных коммутаторах VMware и вы хотите переместить ее (частично или полностью) на распределенные виртуальные коммутаторы VMware, то это несложно. У вас есть два варианта проведения этой операции:

1. Используя стандартные механизмы по работе с сетями на ESXi, в первую очередь настройки в окне **Home** ⇒ **Inventory** ⇒ **Network**. Метод применим в любых случаях, но требует большого количества повторяющихся действий при большом количестве серверов.
2. Используя механизм **Host Profiles**, когда применение распределенных виртуальных коммутаторов на первом сервере мы настраиваем вручную и копируем эту настройку на остальные с помощью **Host Profiles**. Метод хорош автоматизацией.

Второй способ предполагает автоматизацию всех действий, что правильно. Но есть ограничения в его применимости: конфигурации серверов по сетевым контроллерам должны быть одинаковы, для применения профиля настроек сервера переводятся в режим обслуживания, что требует выключения или переноса ВМ на другие сервера. Здесь под «конфигурацией» понимаются количество сетевых адаптеров и порядок их подключения к разным физическим сетям (например, первый адаптер каждого сервера – Management VLAN, со второго по третий – Production и т. д.).

Зато функционал **Host Profiles** отлично подходит для отслеживания отхода настроек сети на серверах от заданных – если кто-то по ошибке, к примеру, меняет конфигурацию сети на каком-то сервере, то этот сервер перестает удовлетворять назначенному ему шаблону настроек, о чем мы получаем уведомление.

Также функционал **Host Profiles** интересен для настройки новодобавленных серверов ESXi.

Сначала разберем первый способ. Итак, вы имеете несколько серверов ESXi под управлением vCenter с уже существующей виртуальной сетью (рис. 2.18).

Для перевода всей или части виртуальной сети на распределенные коммутаторы выполните следующую процедуру.

Первое. Создайте распределенный коммутатор. Создайте необходимые группы портов на нем.

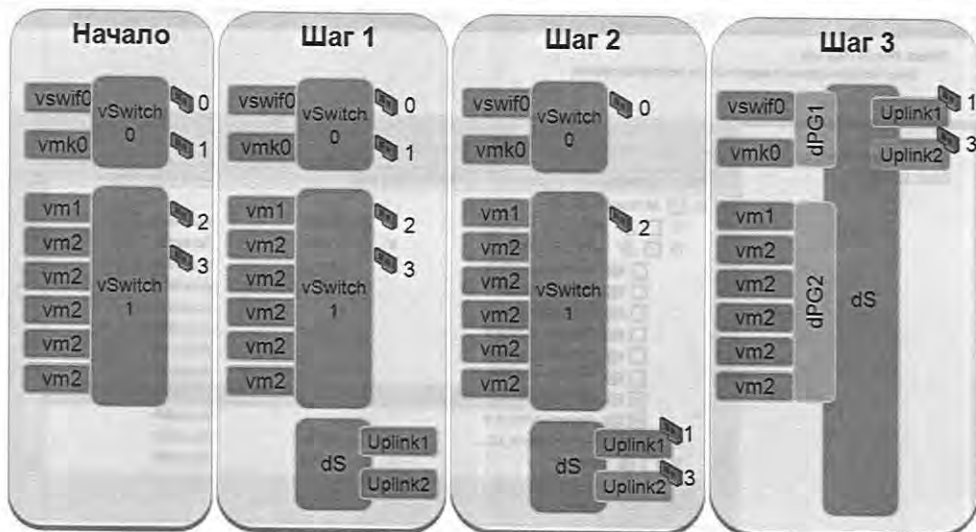


Рис. 2.18. Порядок миграции на dvSwitch
Источник: VMware

Второе. Добавьте сервер к этому распределенному вКоммутатору. Перенесите на dvSwitch часть физических сетевых интерфейсов сервера. Как вы понимаете, один физический контроллер не может принадлежать одновременно двум коммутаторам. В идеале на каждом вашем стандартном коммутаторе внешних подключений хотя бы два (для дублирования) – и один из них мы сейчас можем освободить. Освобождаем один из них и переносим на распределенный коммутатор. Этот шаг повторите для каждого сервера. Если внешнее подключение только одно, тогда придется сначала его отключить от обычного вКоммутатора, затем подключить к распределенному. Разница только в том, что во время переключения ВМ будут отрезаны от сети. Однако имейте в виду, что добавлять vmpic к распределенному вКоммутатору можно без предварительного отключения его от стандартного вКоммутатора – это отключение произойдет автоматически.

Третье. Перенесите ВМ на группы портов распределенного вКоммутатора. Проще всего это сделать, пройдя **Home** ⇒ **Inventory** ⇒ **Networking** ⇒ и в контекстном меню dvSwitch выбрать пункт **Migrate Virtual Machine Networking**. Будет запущен мастер:

❑ **Select Networks** – на этом шаге вам будет предложено выбрать, откуда и куда следует мигрировать ВМ (см. рис. 2.19).

Соответственно, в выпадающем меню верхнего раздела **Source Network** вам следует выбрать ту группу портов, виртуальные машины из которой вы хотите перенести на распределенный коммутатор.

Если вы выберете **Include all virtual machine network adapters that are not connected to any network**, то вам будут предложены ВМ с сетевыми контроллерами, не подключенными ни в одну группу портов.

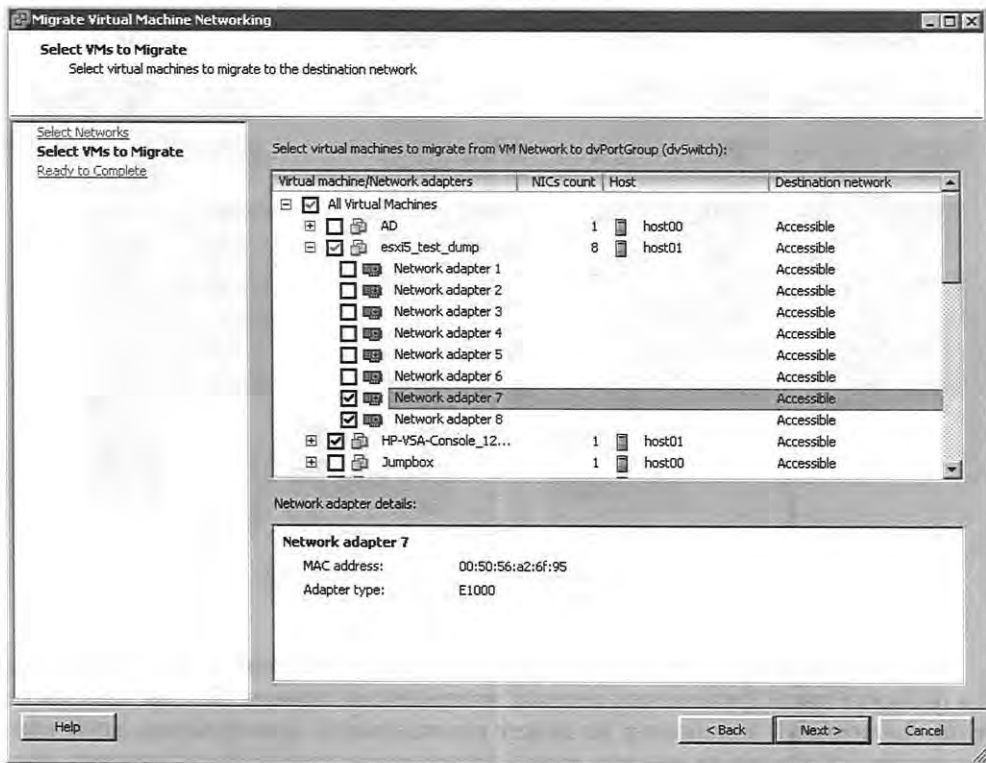


Рис. 2.19. Миграция VM между группами портов

В выпадающем меню **Destination Network** следует выбрать ту группу портов распределенного коммутатора, куда мы хотим перенести VM.

Ссылки **Filter by VDS** и **Filter by Network** (заменяющие друг друга после нажатия) позволяют выбрать нам или dvSwitch, затем его группу портов, или сразу группу портов из общего списка (включая группы портов на стандартном коммутаторе). Переключение делается из удобства – в последнем случае может быть неудобно выбирать из большого общего списка групп портов. Зато можно выбрать не только распределенные группы портов;

- ☐ **Select VMs to Migrate** – на этом шаге мы выбираем виртуальные машины для переноса (отображаются только VM, подключенные к группе портов, выбранной ранее в выпадающем меню **Source Network**). Обратите внимание, что для виртуальных машин с несколькими виртуальными сетевыми контроллерами мы можем независимо выбрать, какие из контроллеров хотим переподключить в рамках данной миграции (см. рис. 2.19).

После нажатия кнопки **Finish** на последнем шаге мастера выбранные контроллеры выбранных виртуальных машин будут последовательно перенесены в выбранную группу портов.

Другой вариант: пройти **Home** ⇒ **Inventory** ⇒ **Networking** ⇒ выделить группу портов стандартного коммутатора и перейти на вкладку **Virtual Machines**. Здесь выделите нужные ВМ (можно просто рамкой или используя **Shift** и **Ctrl**) и перетащите их в нужную группу портов распределенного виртуального коммутатора.

ВМ будут последовательно подключены к той группе портов, куда вы их перенесли.

Если у какой-либо из выбранных ВМ будет несколько сетевых контроллеров, то на экране появится сообщение с вопросом, как поступить. При нажатии **Yes** все сетевые контроллеры будут перенесены в новую группу портов. При нажатии **No** такие ВМ будут пропущены, и перенесутся только те из выбранных ВМ, у которых есть только один сетевой контроллер. Если вам требуется подключить к новой группе портов лишь некоторые сетевые контроллеры ВМ, то сделать это придется или первым способом – через мастер **Migrate Virtual Machine Networking**, или просто в свойствах виртуальной машины.

После переноса ВМ (и при необходимости интерфейсов VMkernel) на распределенные вКоммутаторы обычные вКоммутаторы вам следует удалить, а все физические сетевые контроллеры переназначить на dvSwitch.

Отдельно расскажу про перенос на dvSwitch интерфейсов VMkernel. Эта миграция выполняется для каждого сервера ESXi индивидуально. На распределенном коммутаторе должна быть группа портов, в которую вы планируете подключать переносимые интерфейсы. Для выполнения самой миграции пройдите **Home** ⇒ **Hosts and Clusters** ⇒ **Configuration** для сервера ⇒ **Networking** ⇒ кнопка **Distributed virtual Switch** ⇒ ссылка **Manage Virtual Adapters**. В открывшемся окне нажмите ссылку **Add**, в запустившемся мастере выберите **Migrate existing virtual adapters** (рис. 2.20).

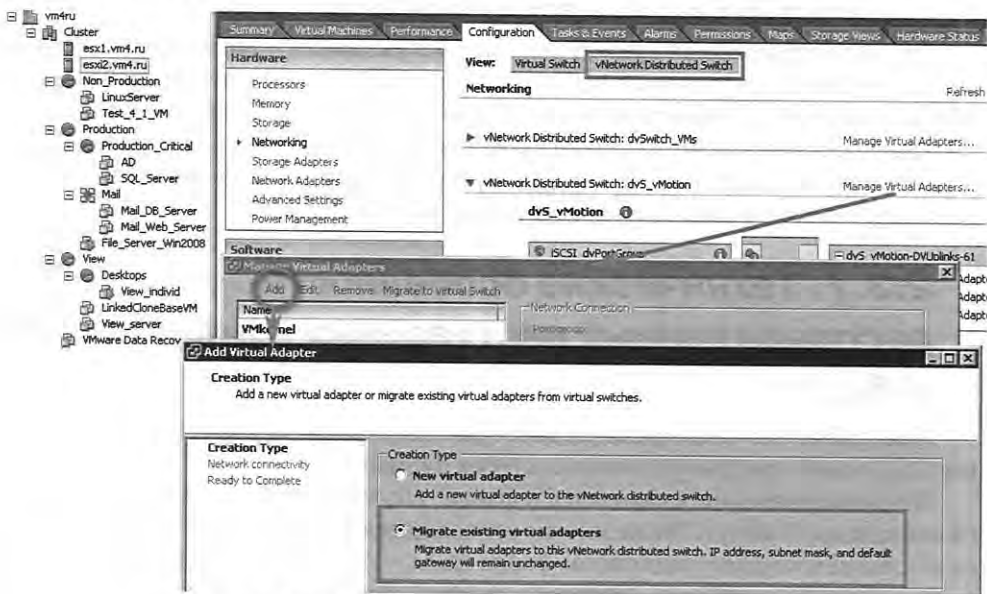


Рис. 2.20. Миграция интерфейсов ESXi на dvSwitch

С помощью этого мастера вы можете перенести существующие и создать новые интерфейсы VMkernel для сервера на dvSwitch. Это не только проще, чем создание нового интерфейса и удаление старого, – это еще и удобнее, потому что сохраняются старые MAC-адреса (а на них могут быть назначены резервации в DHCP, например).

Напомним, что сейчас мы говорим о миграции со стандартных виртуальных коммутаторов на распределенные. На данном этапе вы освободили стандартные коммутаторы, перенесли виртуальные машины и интерфейсы гипервизора dvSwitch. Теперь стандартные Коммутаторы можно удалить, физические контроллеры, которые они еще использовали, – перенести на распределенные коммутаторы.

Далее поговорим про решение той же задачи с помощью **Host Profiles**.

Использование **Host Profiles** поможет нам автоматизировать создание dvSwitch и назначение внешних подключений. Последовательность действий такова:

1. Мы создаем dvSwitch. Создаем группы портов. Выполняем необходимые настройки.
2. Добавляем к этому распределенному коммутатору один сервер. Переносим его физические сетевые контроллеры на dvSwitch. Удаляем ненужные теперь стандартные виртуальные коммутаторы.
3. Затем снимаем с этого сервера профиль настроек.
Для этого идем в **Home** ⇒ **Management** ⇒ **Host Profiles** и нажимаем **Create Profile**. В запустившемся мастере выбираем наш эталонный сервер.
4. Созданный профиль назначаем на следующий сервер или сервера. Для этого идем в **Home** ⇒ **Management** ⇒ **Host Profiles**, выбираем ранее созданный профиль и нажимаем **Attach Host/Cluster**.

Применение профиля требует режима обслуживания – это значит, на сервере не должно быть включенных ВМ. Таким образом, если мы не хотим выключать все ВМ, то придется применять профиль настроек к серверам последовательно.

Получается, что использование профиля настроек удобно для первоначально одновременной настройки множества серверов, когда ВМ еще нет. Или при добавлении в существующую инфраструктуру нового сервера.

2.3.7. Технические особенности распределенных виртуальных коммутаторов VMware

Настройки dvSwitch хранятся в базе данных сервера vCenter. Но каждый сервер ESXi имеет локальную копию настроек dvSwitch.

Эта локальная копия находится в файле `/etc/vmware/dvsdata.db`. Обновляется она каждые 5 минут. Также многие относящиеся к dvSwitch настройки хранятся в основном конфигурационном файле ESXi – `/etc/vmware/esx.conf`. Еще одно место хранения настроек – каталог с именем `.dvsData` на каждом из хранилищ сервера (рис. 2.21).

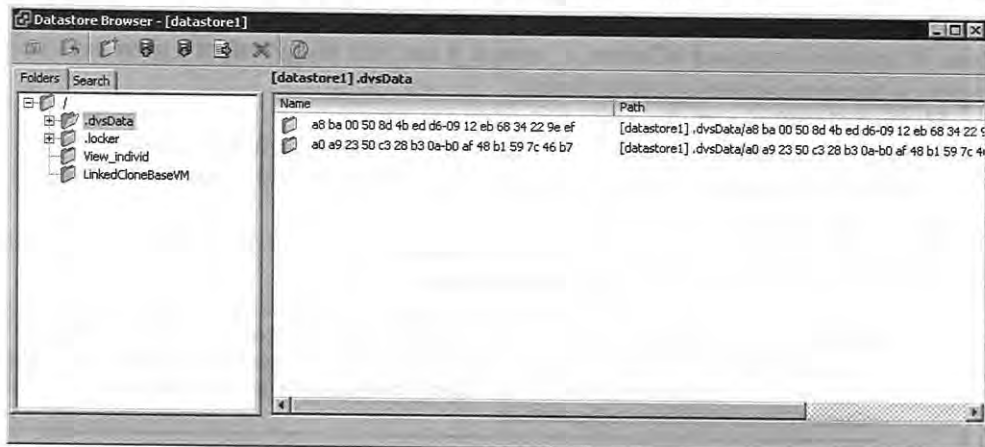


Рис. 2.21. Каталог с настройками dvSwitch

Каталог с настройками dvSwitch появляется на каждом хранилище, где расположена хотя бы одна из подключенных к этому распределенному виртуальному коммутатору виртуальных машин. В каталогах с UUID (уникальными внутренними ID распределенных коммутаторов) находятся файлы с именами в виде цифр. Эти цифры – номера портов распределенного коммутатора. Таким образом, в файле с именем 130 содержится информация о порте номер 130. Эта информация используется VMware HA – при перезапуске ВМ на другом сервере необходимо передать информацию о порте dvSwitch, к которому она подключена. Вся необходимая информация сохраняется на том же хранилище, где расположена ВМ на случай недоступности vCenter, который мог бы в этом помочь.

Данная информация приведена в основном для справки – у нас практически нет способов взаимодействовать с описанными объектами и, самое главное, вряд ли возникнет необходимость. Необходимость может возникнуть разве что при диагностике и решении проблем. Единственный инструмент, который нам в этом может помочь, – утилита net-dvs, дающая доступ к дампу настроек распределенного виртуального коммутатора на конкретном сервере ESXi.

Сценарий использования этой утилиты мне видится следующим: допустим, у нас есть проблема в виде неработающей сети для виртуальной машины. Мы исчерпали другие возможности обнаружить первопричину проблемы и решили перепроверить все возможные настройки. С помощью net-dvs мы можем изучить все-все доступные настройки виртуального коммутатора, и это информация «из первых рук».

2.3.8. Основы решения проблем dvSwitch

Распределенный виртуальный коммутатор интересен своими возможностями. Платой за них является возросшая сложность. Самая заметная прикладная иллюстрация этого – зависимость от vCenter.

Допустим, произошла проблема с распределенным виртуальным коммутатором. К примеру, отказал vCenter, и теперь у нас нет возможности изменять настройки распределенного коммутатора.

Наши действия:

Первое. Не паниковать и не совершать резких движений. Практически в любой подобной ситуации сеть будет продолжать работать, мы лишь потеряем возможность изменять ее настройки.

Второе. Спланировать решение проблемы. Если отказал vCenter – будем ли мы пытаться его реанимировать или установим новый.

Обратите внимание. Если резервное копирование БД vCenter Server осуществлялось на регулярной основе, то реанимировать vCenter Server мы сможем всегда. Так что при использовании dvSwitch регулярное резервное копирование базы данных vCenter Server особенно обязательно.

В последнем случае нам потребуется аккуратно создать новый распределенный коммутатор, перенести на него часть аплинков, затем виртуальные машины и так далее – см. раздел 2.3.6 про миграцию, потому что именно миграцией между коммутаторами мы и займемся.

Иногда какая-то проблема может затронуть не vCenter и весь dvSwitch, а только сеть одного сервера ESXi. В этом случае нам необходимо будет мигрировать с него все виртуальные машины (если сервер совершенно неуправляем – то выключив их и затем включив на других серверах) и затем пересоздавать сеть. Может оказаться полезным пункт Restore Standard Switch в локальном БИОС-подобном меню DCUI.

2.4. Настройки Security, VLAN, Traffic shaping и NIC Teaming

Здесь будет рассказано о настройках, которые применимы и к стандартным (за редким исключением), и к распределенным виртуальным коммутаторам VMware. Напомню, что часть настроек распределенных виртуальных коммутаторов уникальна и доступна только для них. Такие настройки описаны в разделах 2.3.4 и 2.3.5.

2.4.1. VLAN, виртуальные локальные сети. Настройка VLAN для стандартных виртуальных коммутаторов

Немного общей теории о виртуальных локальных сетях. VLAN поддерживаются как стандартными, так и распределенными виртуальными коммутаторами. Суть их в том, чтобы возложить на коммутатор работу по анализу и контролю трафика на втором уровне модели OSI с целью создавать не связанные между собой сети без физической их изоляции друг от друга.

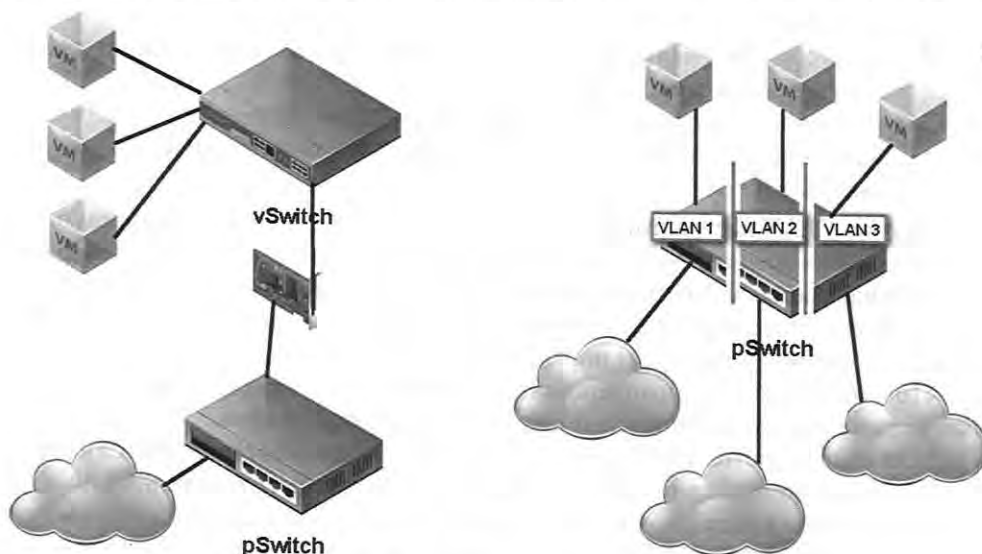


Рис. 2.22. Сервера (ВМ) подключены к одному коммутатору, но к разным VLAN

Что мы здесь видим?

Все сервера (в нашем случае это виртуальные сервера, виртуальные машины, но для VLAN это не принципиально) подключены к одному коммутатору (слева), но на этом коммутаторе настроены несколько VLAN, и все ВМ подключены к разным (справа). Это означает, что на коммутаторе (в случае виртуальных машин – на вКоммутаторе) мы сделали настройку – порт для ВМ1 принадлежит виртуальной сети 1 (с идентификатором VLAN ID = 1), порт для ВМ2 принадлежит VLAN 2 и т. д. Это означает, что эти ВМ друг с другом взаимодействовать не могут, им не даст такой возможности сам коммутатор. Изоляция между серверами (ВМ) получается практически такой же, как если бы они были подключены к разным коммутаторам.

Небольшое примечание: на коммутаторе физическом, к которому подключены коммутаторы виртуальные, также в обязательном порядке должны быть настроены VLAN. В общем случае VLAN – это разделение всей нашей сети на несколько как будто бы не связанных сегментов. Именно всей сети, а не отдельно взятого коммутатора.

Обратите внимание. Если две ВМ работают на одном сервере и подключены в одну группу портов (то есть в один VLAN), то при общении друг с другом их трафик остается «внутри» ESXi. А если ВМ подключены к одному вКоммутатору, но к разным VLAN (то есть к разным группам портов с разными VLAN), то в этом случае обмен трафиком между ними пойдет через физическую сеть, потому что виртуальный коммутатор не умеет маршрутизировать VLAN.

Зачем это надо:

- ☐ для того чтобы уменьшить домены широковещательной рассылки, следовательно, снизить нагрузку на сеть;
- ☐ для того чтобы повысить безопасность – хотя устройства подключены в одну сеть физически, находясь в разных `vlan`, они не смогут взаимодействовать по сети.

Обычно VLAN настраиваются на коммутаторах, и только коммутаторы о них знают – с точки зрения конечного устройства (такого, как физический сервер или виртуальная машина) в сети не меняется ничего. Что означает настроить `vlan` на коммутаторе? Это означает для всех или части портов указать `vlan id`, то есть `vlan` с каким номером принадлежит порт. Теперь если сервер подключен к порту с `vlan id` = «10», то коммутатор гарантированно не перешлет его трафик в порты с другим `vlan id`, даже если сервер посылает широковещательный трафик.

Если используются VLAN, то коммутаторы (обычно этим занимаются именно коммутаторы, но иногда и конечные устройства могут использовать VLAN) добавляют в каждый кадр поле, в которое записывают так называемый «`vlan id`» (тэг VLAN, идентификатор VLAN) – число в диапазоне от 1 до 4094. Эту операцию называют «тэгованием», добавлением в кадр тэга `vlan`.

Получается, для каждого порта коммутатора указано, кадры с каким `vlan id` могут пройти в порт, а в кадрах прописано, к какому `vlan` относится каждый кадр. За счет этого коммутатор и контролирует, какому трафику можно попасть в какой порт, а какому – нельзя.

Один VLAN может распространяться (и, как правило, распространяется) на несколько коммутаторов. То есть устройства, находящиеся в одном VLAN, могут физически быть подключены к разным коммутаторам.

Если за настройки сети отвечаете не вы, то формально, с точки зрения администрирования ESXi, нам достаточно знать: ESXi поддерживает VLAN, то есть протокол 802.1q. Мы можем настроить `vlan id` для групп портов на коммутаторе, и они будут тэговать и ограничивать проходящий через них трафик.

Если тема настройки VLAN вас касается, то несколько слов подробнее.

У вас есть три принципиально разных варианта настройки `vlan`:

- ☐ `external switch tagging, EST` – установка тэгов VLAN только на внешних, физических коммутаторах. За VLAN отвечают лишь физические коммутаторы, на коммутаторах трафик приходит без тэгов VLAN;
- ☐ `virtual switch tagging, VST` – установка тэгов VLAN на виртуальных коммутаторах. Коммутаторы физические настраиваются таким образом, чтобы тэги VLAN не вырезались из кадров, передаваемых на физические интерфейсы серверов ESXi, то есть виртуальным коммутаторам;
- ☐ `virtual guest tagging, VGT` – установка тэгов VLAN на гостевой ОС в виртуальной машине. В этом случае коммутаторы (и виртуальные, и физические) не вырезают тэг VLAN при пересылке кадра на клиентское устройство (в нашем случае на ВМ), а тэги VLAN вставляются в кадры самим клиентским устройством (в нашем случае виртуальной машиной).

EST, external switch tagging

Настройка тэгирования vlan *только* на физических коммутаторах – схема на рис. 2.23.

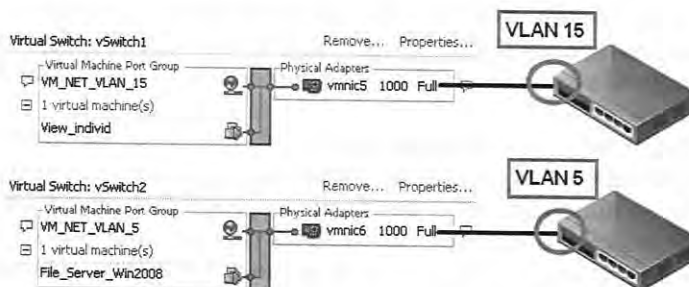


Рис. 2.23. Схема External switch tagging

Этот подход хорош тем, что все настройки VLAN задаются *только* на физических коммутаторах. Вашим сетевым администраторам не придется задействовать в этом коммутаторы ESXi – порты физических коммутаторов, куда подключены физические сетевые контроллеры ESXi, должны быть настроены обычным образом, чтобы коммутаторы вырезали тэг VLAN при покидании кадром порта.

Минус подхода EST – в том, что на каждый VLAN нам нужен выделенный физический сетевой контроллер в ESXi.

Таким образом, при реализации схемы EST уже физические коммутаторы пропускают в порты к ESXi пакеты только из нужных VLAN (5 и 15 в моем примере). Виртуальные машины и виртуальные коммутаторы про VLAN ничего не знают.

VST, virtual switch tagging

Настройка тэгирования vlan и на виртуальных коммутаторах – схема на рис. 2.24.

Этот подход предполагает настройку VLAN и на коммутаторах. Удобен тем, что на один коммутатор (и на одни и те же vmnic) может приходиться трафик множества VLAN.

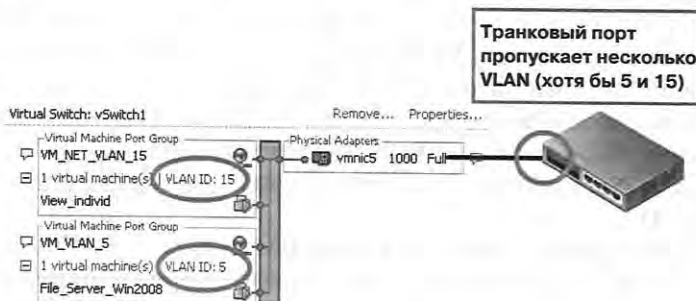


Рис. 2.24. Схема Virtual switch tagging

Из минусов – требует настройки на стороне и физических, и виртуальных коммутаторов. Те порты физических коммутаторов, к которым подключены контроллеры серверов ESXi, следует настроить как «транковые», то есть пропускающие пакеты из всех (или нескольких нужных) VLAN, и не вырезающие тэги VLAN при проходе кадра сквозь порт. А на вКоммутаторах надо сопоставить VLAN ID соответствующим группам портов. Впрочем, это несложно: **Configuration** ⇒ **Networking** ⇒ свойства vSwitch ⇒ свойства группы портов ⇒ в поле **VLAN ID** ставим нужную цифру.

Подавляющее большинство инфраструктур, которые в принципе используют vlan, используют именно эту схему. Единственная причина ее не использовать (и, как следствие, использовать EST, если vlan в принципе требуются) – это соображения безопасности. К примеру, если инфраструктура должна соответствовать формальным требованиям регуляторов, то это может означать требование использовать лишь сертифицированное сетевое оборудование. Если виртуальный коммутатор (или сам ESXi) не обладает нужной сертификацией перед регулятором, то это является формальным препятствием перед использованием виртуальных коммутаторов как устройств, обеспечивающих работу vlan.

VGT, virtual guest tagging

Настройка тэгирования vlan в виртуальной машине – схема на рис. 2.25.



Рис. 2.25. Схема Virtual guest tagging

Этот подход хорош в тех редких случаях, когда одна ВМ должна взаимодействовать с машинами из многих VLAN одновременно. Для этого вКоммутатор мы настроим не вырезать тэги VLAN из кадров к этой ВМ (фактически сделаем транковый порт на вКоммутаторе). Чтобы настроить транковый порт на стандартном виртуальном коммутаторе VMware, необходимо для группы портов, к которой подключена ВМ, в качестве VLAN ID прописать значение «4095». Пройдите **Configuration** ⇒ **Networking** ⇒ свойства vSwitch ⇒ свойства группы портов ⇒ в поле **VLAN ID**.

Минус конфигурации в том, что внутри ВМ должно быть ПО, обрабатывающее VLAN, – так как вКоммутатор тэги VLAN вырезать не будет и они будут доходить прямо до ВМ. На физических серверах очень часто этим ПО является драйвер сетевых контроллеров. Это актуально и для ВМ.

Для реализации схемы VGT виртуальные машины должны использовать виртуальные сетевые карты типа e1000 или vmxnet3.

Драйверы vmxnet3 для Windows из состава VMware Tools позволяют настраивать VLAN – см. рис. 2.26.

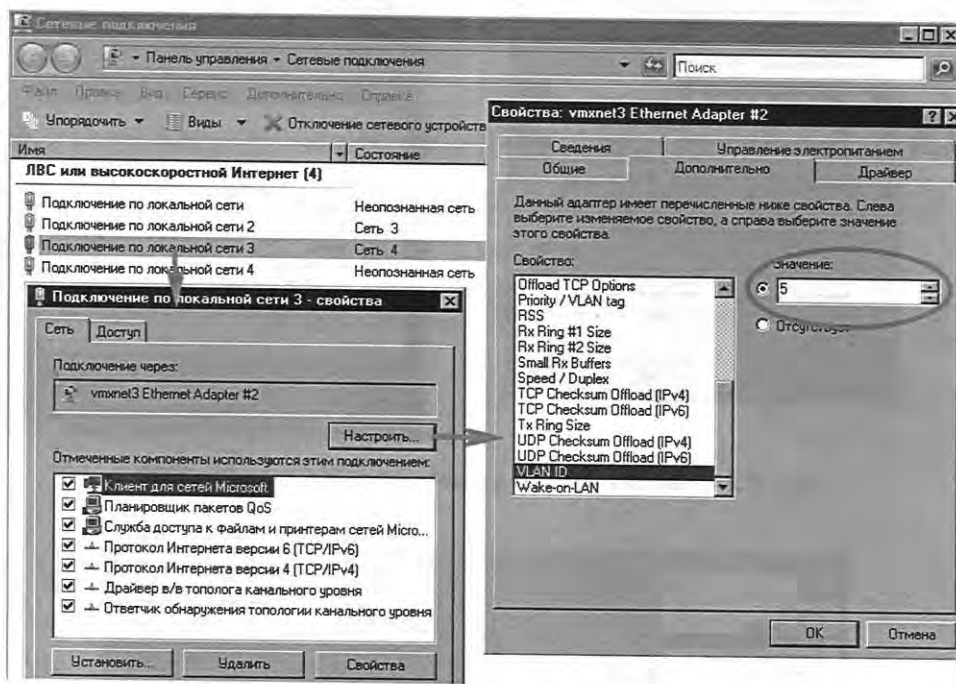


Рис. 2.26. Настройка VLAN в драйвере vmxnet3

Виртуальный контроллер E1000 эмулирует контроллер Intel Pro1000, и если установить соответствующий драйвер Intel (<http://www.intel.com/design/network/drivers/>), то получим все его возможности и, в частности, возможность настраивать VLAN – см. рис. 2.27.

Кроме стандартных виртуальных коммутаторов VMware, некоторые лицензии позволяют использовать распределенные виртуальные коммутаторы VMware. У них немного больше возможностей работы с VLAN (они позволяют использовать Private VLAN) и чуть-чуть по-другому выглядят окна настройки. Подробно см. в следующем разделе.

2.4.2. Настройка VLAN для dvSwitch. Private VLAN

Настройка VLAN для dvSwitch выглядит немного по-другому, нежели для обычного vSwitch (рис. 2.28).

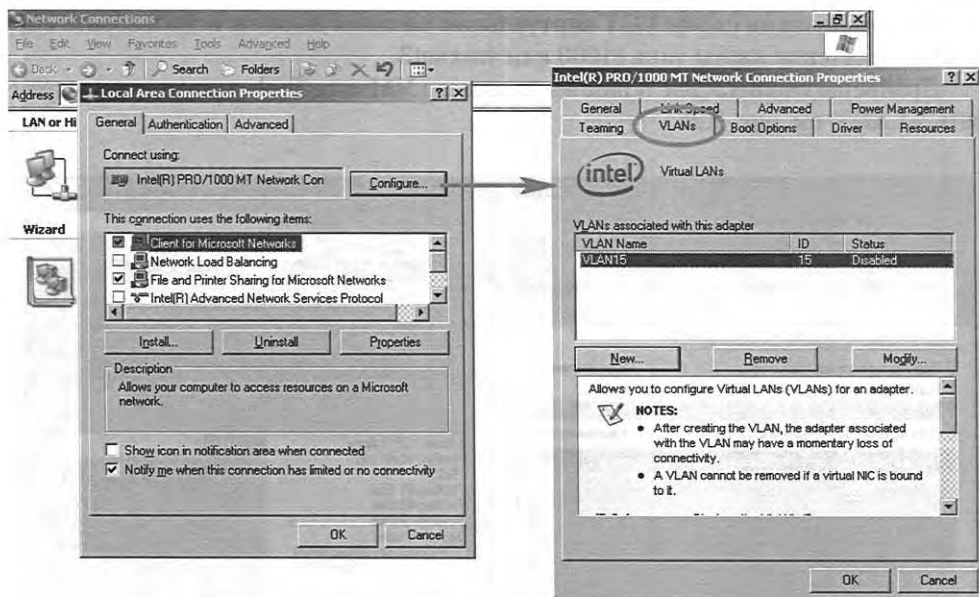


Рис. 2.27. Настройка VLAN в драйвере e1000/Intel Pro 1000

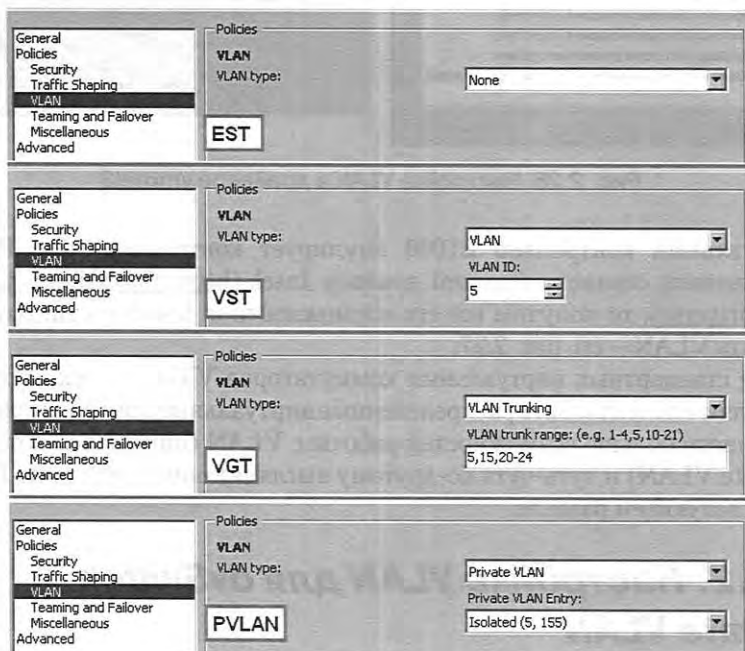


Рис. 2.28. Варианты настройки VLAN для групп портов на распределенном виртуальном коммутаторе VMware

Увидеть эту настройку можно, зайдя в свойство группы портов на dvSwitch. Как вы видите на рисунке, для настройки VGT, или транкового порта (группы портов), вы не указываете значение VLAN ID = 4095 (как для стандартных виртуальных коммутаторов), а выбираете из выпадающего меню **VLAN Trunking**. Кроме того, вы можете явно ограничить, пакеты с какими VLAN ID могут попадать в эту группу портов, – в то время как для обычных коммутаторов такой выбор невозможен.

Private VLAN, PVLAN

Для групп портов на dvSwitch появилась возможность настроить использование Private VLAN (PVLAN). Тайный смысл здесь в следующем: у нас есть какой-то VLAN (например, в моем примере это VLAN 10), какие-то наши VM в нем. Но мы хотим, чтобы часть из этих VM не могла взаимодействовать друг с другом. Можно этот один десятый VLAN разбить на несколько – но это усложнит конфигурирование VLAN в нашей сети, да и при большом количестве виртуальных машин такой вариант решения проблемы невозможен технически. Механизм Private VLAN позволяет сделать несколько «вторичных» (Secondary) VLAN «внутри» нашего основного, Primary VLAN 10. Обратите внимание на рис. 2.29.

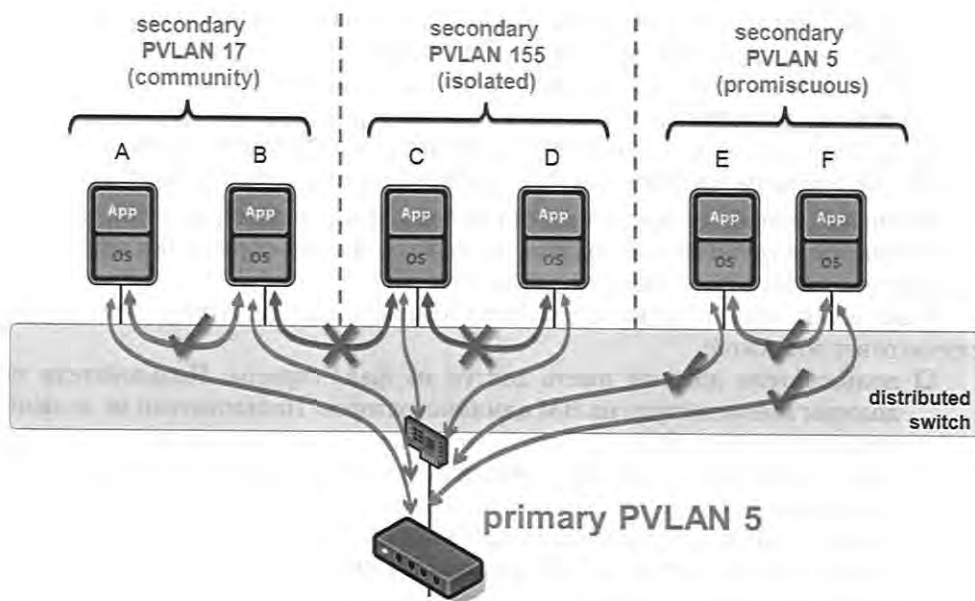


Рис. 2.29. Пример использования Private VLAN
Источник: VMware

Устройства внешней сети считают, что все (здесь – виртуальные) сервера принадлежат одному VLAN с номером 10. И лишь коммутаторы, непосредственно с которыми эти VM работают, знают о разделении основного VLAN 10 на не-

сколько внутренних, вторичных (Secondary) VLAN, с определенными правилами взаимодействия между ними.

Вторичные VLAN могут быть трех типов:

- ❑ Community – VM в этом Secondary VLAN могут взаимодействовать друг с другом и с виртуальными машинами в VLAN Promiscuous (VM E и F), но не могут с VM C и D;
- ❑ Isolated – VM в этом Secondary VLAN могут взаимодействовать только с машинами из Promiscuous VLAN (ну и, разумеется, с внешним миром). То есть виртуальные машины C и D могут работать с VM E и F, но не могут с VM A и B и друг с другом;
- ❑ Promiscuous – когда VM находятся в этом Secondary VLAN, они могут взаимодействовать со всеми VM на этом dvSwitch и всеми физическими и виртуальными компьютерами в данном Primary VLAN. То есть для виртуальных машин E и F доступны все VM. На распределенном vКоммутаторе вторичный VLAN этого типа создается автоматически. Его VLAN ID совпадает с VLAN ID Primary VLAN.

Приведу пример задачи, для решения которой pvlan могут пригодиться.

Допустим, у вас есть инфраструктура виртуальных рабочих столов. Это означает, что на ваших серверах ESXi работает большое количество VM с десктопными ОС, и эти VM выступают в роли рабочих мест пользователей. Кроме того, существует серверная инфраструктура, обслуживающая этих пользователей.

Перечислим вышеописанное, для определенности:

- ❑ VM для пользователей;
- ❑ VM для особых пользователей, например для администраторов;
- ❑ файл-сервера, на которых вышеописанные пользователи хранят данные.

Внимание, вопрос: если специалиста по безопасности спросить о том, какие из вышеописанных групп машин должны иметь возможность обращаться друг к другу по сети, а какие – не должны, то что он ответит?

Разумеется, в реальности ответы могут отличаться, но допустим, что в нашем случае ответ вот какой:

- ❑ пользователи должны иметь доступ на файл-сервера. Пользователи не должны иметь доступа на VM администраторов. Пользователи не должны иметь доступа на VM друг друга (обратите внимание на последнее условие – скорее, оно самое проблематичное в реализации, если речь идет не про pvlan);
- ❑ администраторы должны иметь доступ на файл-сервера. Администраторы должны иметь доступ на VM администраторов (допустим, этого требует специфика работы... или желание поиграть в Counter Strike). Администраторы не должны иметь доступа на VM пользователей (тут имеется в виду, что администраторы – это не helpdesk);
- ❑ файл-сервера должны быть доступны для всех, и друг для друга в том числе (допустим, для репликации данных друг с другом).

Так вот, Private VLAN – это стандарт, позволяющий реализовать именно такие правила видимости в сети. Все эти виртуальные машины будут в одном vlan

с точки зрения «внешней» сети, но для распределенного коммутатора этот «первичный» vlan будет разбит на несколько «вторичных», «внутренних», с нужными нам правилами взаимодействия между собой.

Настраиваются Private VLAN в свойствах распределенного коммутатора, на вкладке **Private VLAN**.

В левой части окна добавляем номер **Primary VLAN** (их может быть несколько на одном dvSwitch). Затем, выделив этот **Primary VLAN**, в правой части вводим номер **Secondary VLAN** с указанием их типа – **Isolated** или **Community** (рис. 2.30).

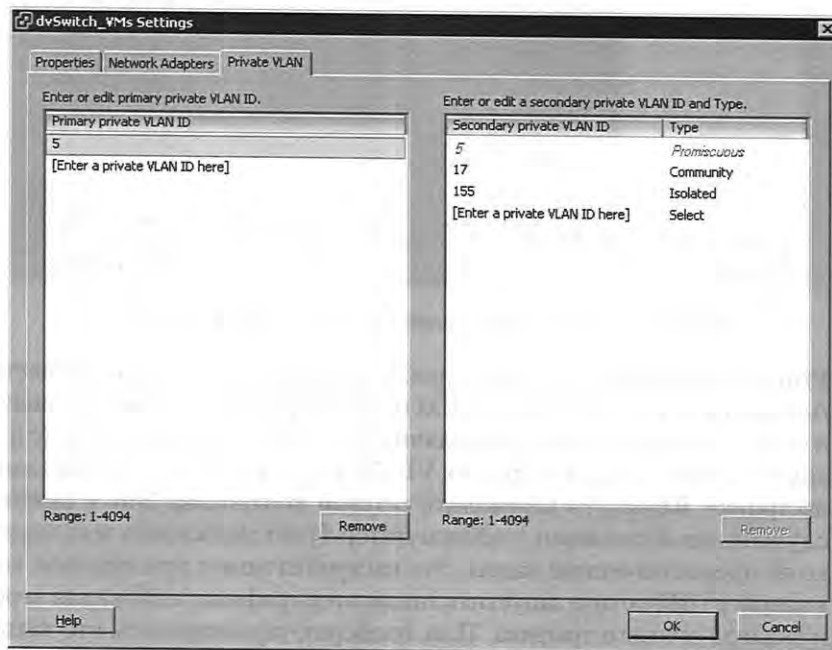


Рис. 2.30. Окно настроек Private VLAN для распределенного vКоммутатора

А потом, зайдя в свойства группы портов на dvSwitch, мы можем указать для нее **Secondary VLAN** – см. рис. 2.31.

Если VM в одном Private VLAN работают на разных серверах ESXi (такое будет всегда, исключая, возможно, тестовые стенды), то в обмене трафиком между ними участвуют физические коммутаторы. На них также должны быть настроены Private VLAN, если мы хотим, чтобы эта схема работала.

2.4.3. Security

Зайдя в свойства vКоммутатора или какой-то группы портов, мы увидим вкладку **Security** и там три настройки:

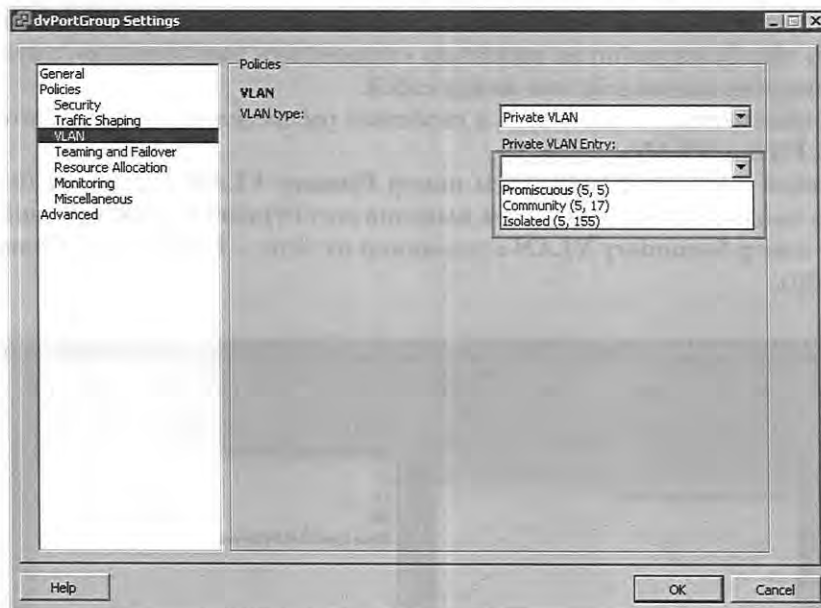


Рис. 2.31. Настройки Private VLAN для группы портов

- ❑ **Promiscuous Mode** – режим прослушивания. Если значение настройки **Accept**, то сетевой контроллер ВМ можно переводить в режим promiscuous, и он будет принимать все проходящие через вКоммутатор кадры (с поправкой на VLAN – кадры из других VLAN доступны не будут). Если значение настройки **Reject**, то переводить сетевой контроллер ВМ в режим прослушивания бесполезно – вКоммутатор будет пересылать в ее порт только ей предназначенные кадры. Эта настройка может пригодиться, если вы в какой-то ВМ хотите запустить анализатор трафика (sniffer) для перехвата и анализа сетевого трафика. Или, наоборот, гарантировать, что такой анализатор не заработает для вашей сети. Значение по умолчанию **Reject**;
- ❑ **MAC Address Changes** – изменение MAC-адреса. **Reject** – при этом значении настройки гипервизор проверяет совпадение MAC-адреса виртуальных сетевых контроллеров ВМ в конфигурационном файле vmx этой ВМ и в пакетах, приходящих по сети. Если пакет предназначен для MAC-адреса, не существующего в конфигурационном файле ни одной ВМ, он будет отклонен. Такое происходит в тех случаях, когда MAC-адрес переопределен средствами гостевой ОС. **Accept** – при таком значении настройки проверка не производится. Если ваша ВМ отключилась от сети – то проверьте, возможно, для вКоммутатора или группы портов этой ВМ **MAC Address Changes** = **Reject**, а кто-то зашел в диспетчер устройств гостевой ОС и менял MAC-адрес сетевого контроллера;
- ❑ **Forged Transmits** – если в исходящих кадрах MAC-адрес источника отличается от MAC-адреса ВМ (прописанного в файле vmx), то такие кадры

будут отброшены. Само собой, это в случае значения настройки = **Reject**. В случае **Accept** проверки не производится. Настройка **Accept** необходима для некоторых режимов работы NLB кластера Microsoft – это из известных мне примеров.

По сути, и **MAC Address Changes**, и **Forged Transmits** делают одно и то же – отсекают ВМ от сети, если ее MAC-адрес отличается от указанного в ее файле настроек (*.vmx). Но первая настройка блокирует входящий трафик, а вторая – исходящий.

2.4.4. Ограничение пропускной способности (Traffic Shaping)

Для ВКоммутатора целиком или для какой-то одной группы портов у нас есть возможность ограничить пропускную способность его портов.

Обратите внимание на то, что ограничению не подвергается трафик, остающийся только на виртуальном коммутаторе. Если виртуальные машины работают на одном сервере и подключены к одному ВКоммутатору, то трафик между ними не выходит за пределы данного ВКоммутатора (за исключением случая, когда эти ВМ в разных VLAN). В таком случае ограничения пропускной способности канала на трафик между этими двумя виртуальными машинами распространяться не будут.

Зайдя в окно настроек **Traffic shaping (Configuration ⇒ Network ⇒ Properties** для нужного ВКоммутатора ⇒ **Edit** для самого ВКоммутатора или одной группы портов), мы увидим три настройки:

- ❑ **Average Bandwidth** – столько килобит в секунду в среднем может проходить через каждый порт этого коммутатора/группы портов. Фактически средняя, обычная скорость сети;
- ❑ **Peak Bandwidth** – столько килобит в секунду может проходить через порт, если полоса пропускания занята не полностью. Фактически максимальная скорость сети. Это значение всегда должно быть не меньше **Average Bandwidth**;
- ❑ **Burst Size** – если ВМ пытается передавать данные со скоростью, большей, чем average bandwidth, то превышающие это ограничение пакеты помещаются в специальный буфер, размер его как раз и задается этой настройкой. Когда буфер заполнится, то данные из него будут переданы со скоростью **Peak Bandwidth** (если у коммутатора есть свободная полоса пропускания).

Обратите внимание. Эти настройки применяются к каждому виртуальному сетевому контроллеру (на самом деле к порту ВКоммутатора, куда те подключены). Таким образом, если ВМ имеет два виртуальных сетевых контроллера в одной группе портов, то для каждого из них эти настройки применяются независимо.

Для распределенных виртуальных коммутаторов возможно ограничение как исходящего, так и входящего трафика.

2.4.5. NIC Teaming. Группировка сетевых контроллеров

Если зайти в настройки vКоммутатора или группы портов, то последней вкладкой мы увидим **NIC Teaming**, группировку контроллеров. Она нам потребуется в том случае, если к vКоммутатору у нас подключен более чем один физический сетевой контроллер сервера (vmpnic).

А зачем мы можем захотеть, чтобы к одному vКоммутатору были подключены несколько vmpnic? Ответ прост: для отказоустойчивости в первую очередь и для повышения пропускной способности сети – во вторую.

Обратите внимание. Если мы подключаем к одному виртуальному коммутатору, стандартному или распределенному, несколько физических сетевых контроллеров, то они должны быть из одного домена широковещательной рассылки. VMware не рекомендует подключать к одному vКоммутатору сетевые карты, подключенные в разные несвязанные физические сети или несвязанные VLAN: виртуальные коммутаторы VMware являются коммутаторами второго уровня, обрабатывают только кадры Ethernet (второй уровень модели OSI) и не могут осуществлять маршрутизацию.

Если у vКоммутатора только один физический сетевой контроллер, то сам этот контроллер, его порт в физическом коммутаторе и физический коммутатор целиком являются единой точкой отказа. Поэтому для доступа в сеть критичных ВМ более чем правильно использовать два или более vmpnic, подключенных в разные физические коммутаторы.

Но здесь встает вопрос политики их использования. Мы можем использовать конфигурацию, дающую лишь отказоустойчивость: когда работает только один vmpnic, а остальные ожидают его выхода из строя, чтобы подменить его. Или мы можем задействовать сразу несколько сетевых контроллеров сервера, тем или иным образом балансируя между ними нагрузку.

Взглянем на окно настроек – рис. 2.32.

Failover Order. Самое нижнее поле позволяет выбрать используемые (**Active Adapters**), запасные (**Standby Adapters**) и неиспользуемые (**Unused Adapters**) физические сетевые контроллеры из подключенных к этому vКоммутатору. Если вы хотите, чтобы какие-то vmpnic стали резервными и не были задействованы в нормальном режиме работы, тогда перемещайте их в группу **Standby**. Все (или несколько) оставляйте в **Active**, если хотите балансировки нагрузки. Ну а **Unused** обычно нужна на уровне групп портов – когда у vКоммутатора много каналов во внешнюю сеть, но трафик именно конкретной группы портов вы через какие-то пускать не хотите ни при каких обстоятельствах.

Failback. Эта настройка напрямую относится к **Failover Order**. Если у вас vmpnic3 **Active**, а vmpnic2 **Standby**, то в случае выхода из строя vmpnic3 его подменит vmpnic2. А что делать, когда vmpnic3 вернется в строй? Вот если **Failback** выставлен в **Yes**, то vmpnic2 опять станет **Standby**, а vmpnic3 – опять **Active**. Соответственно, если **Failback** = **No**, то даже когда vmpnic3 опять станет работоспособным, он станет

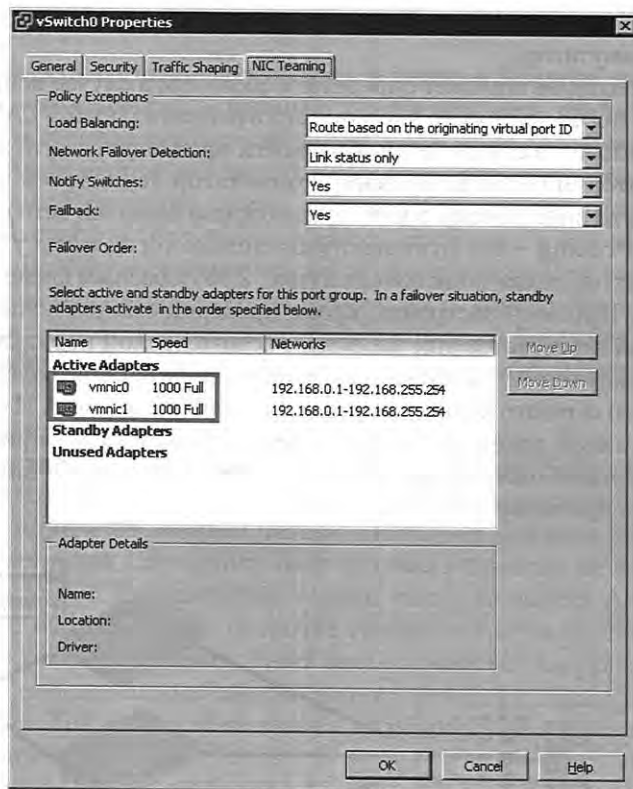


Рис. 2.32. Окно настроек группировки контроллеров – NIC Teaming

Standby. Каким образом ESXi понимает, что vnic не работоспособен? См. пункт **Network Failover Detection**.

Notify Switches. Эта настройка включает (Yes) или выключает (No) оповещение физических коммутаторов об изменениях в MAC-адресах VM на ESXi. Когда вы создаете новую VM и подключаете ее к группе портов (как вариант – добавляете в VM еще один виртуальный сетевой контроллер) или когда трафик VM начинает идти через другой vnic из-за сбоя ранее задействованного – тогда ESXi отправит пакет trap с оповещением вида «Такой-то MAC-адрес теперь доступен на этом порту».

Рекомендуется выставлять в **Yes** для того, чтобы физические коммутаторы максимально быстро узнавали о том, на каком их порту доступен MAC-адрес VM. Однако некоторые приложения потребуют выключения этой функции, например кластер Microsoft NLB, когда он использует multicast.

Network Failover Detection. Каким образом ESXi будет определять, что физический сетевой контроллер неработоспособен? Вариантов два:

- ❑ **Link Status Only** – когда критерием служит лишь наличие линка, сигнала. Подобный режим позволит обнаружить такие проблемы, как выход из

стройка самого vmpnic, отключенный сетевой кабель, обесточенный физический коммутатор.

Такой подход не поможет определить сбой сети в случае неправильной настройки порта, например внесение его в неправильный VLAN и т. п. Также он не поможет в случае, если обрыв сети произошел где-то за физическим коммутатором (если физический коммутатор не настроен на отключение порта, куда подключен ESXi в случае обрыва внешней сети);

- **Beacon Probing** – эта функция нужна только тогда, когда у вКоммутатора несколько внешних подключений (рис. 2.33) к разным физическим коммутаторам. При этой настройке, кроме проверки статуса подключения, виртуальный коммутатор еще рассылает (с интервалом порядка 5–10 секунд) через каждый свой vmpnic широковещательные пакеты, содержащие MAC-адрес того сетевого интерфейса, через который они ушли. И ожидается, что каждый такой пакет, посланный с одного vmpnic, будет принят на других vmpnic этого вКоммутатора. Если этого не происходит – значит, где-то по каким-то причинам сеть не работает.

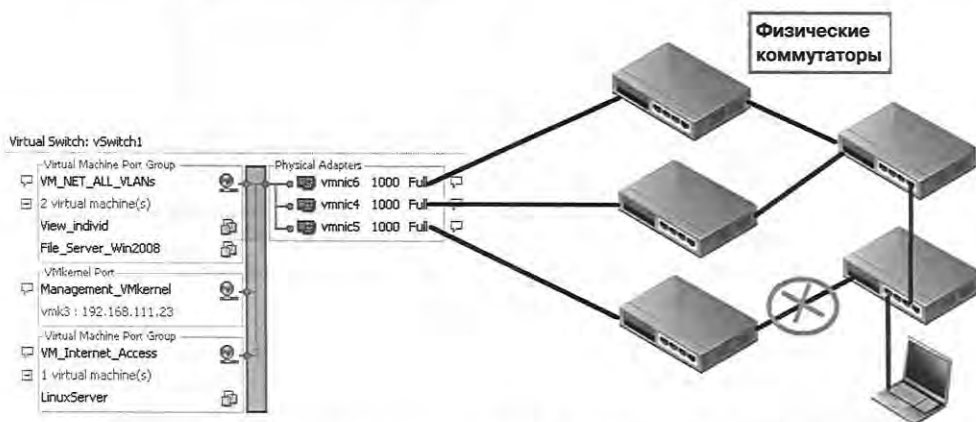


Рис. 2.33. Пример конфигурации сети, при которой имеет смысл использовать Beacon Probing

На что вы должны обратить внимание в данном примере: внешняя сеть не сумеет сама обработать сбой в указанном месте (однако в реальности было бы правильнее настроить физическую сеть так, чтобы подобная проблема была обработана самой сетью и связность сети не пострадала из-за этой проблемы).

В этом примере пакеты, посланные через vmpnic5, не дойдут до клиентов, подключенных к «дальним» физическим коммутаторам. Если для определения отказов сети используется «Link status only», то ESXi не сможет определить такую неработоспособность сети. А beaconing сможет – потому что широковещательные пакеты от vmpnic5 не будут приняты на vmpnic3 и vmpnic2.

Но обратите внимание: если beacon-пакеты отправляются и не принимаются в конфигурации с двумя vmpnic на вКоммутаторе, то невозможно определить,



какой из них не надо использовать – ведь с обоих beacon-пакеты уходят и на оба не приходят.

Тогда в Коммутатор начинает работать в режиме **Shotgun**, что здесь можно перевести как «двустволка», – начинает отправлять весь трафик через оба/все подключения, мол, через какой-то да дойдет. VMware не поддерживает использования механизма beaconing для виртуальных коммутаторов с двумя аплинками.

Конечно, такой ситуации лучше избегать. Сделать это можно правильной структурой физической сети, чтобы какие-то проблемы в ней решались за счет Spanning Tree. Вообще, механизм beaconing позиционируется как крайнее средство – если вы не можете повлиять на правильность конфигурации сети на физической стороне, то подстрахуйтесь от сбоев с помощью beaconing. Но эффективнее, когда подобные сбои в сети устраняются средствами этой сети и beaconing вам не нужен.

Наконец, самая интересная настройка.

Load Balancing. В этом выпадающем меню вы можете выбрать, по какому алгоритму будет балансироваться трафик виртуальных машин между каналами во внешнюю сеть виртуального коммутатора, к которому они подключены.

Вариант настройки **Use explicit failover order** указывает не использовать балансировку нагрузки. Используется первый vmpnic из списка Active – см. чуть выше описание **Failover Order**. А прочие три варианта настройки – это как раз выбор того, по какому принципу будет балансироваться нагрузка. Суть в чем – есть трафик, и есть несколько каналов наружу (vmpnic#). Надо трафик поделить между каналами. Три варианта настройки отличаются тем, каким образом будет делиться трафик:

❑ **Route based on the originating port ID** – балансировка по номеру порта.

У каждой ВМ (у каждого виртуального сетевого контроллера в ВМ) есть порт в вКоммутаторе, к которому она подключена. При данном варианте настройки балансировки нагрузки трафик будет делиться по этим портам – весь трафик от одного порта вКоммутатора будет идти в какой-то один vmpnic; трафик из другого порта – через другой vmpnic и т. д. Выбор очередного vmpnic осуществляется случайным образом, не по его загрузке. Данный метод балансировки нагрузки используется по умолчанию и является рекомендуемым. Рекомендуемым он является по той причине, что дает какую-никакую балансировку нагрузки, а накладные расходы на анализ трафика минимальны. Однако трафик от одного виртуального контроллера не получит полосы пропускания больше, чем дает один физический интерфейс, выступающий каналом во внешнюю сеть. Косвенный вывод отсюда – виртуальная машина с несколькими виртуальными сетевыми контроллерами сможет задействовать несколько каналов во внешнюю сеть. Однако выбор аплинка для очередного виртуального порта происходит при старте ВМ (или ее миграции) и не меняется потом – это значит, что возможны ситуации, когда один из аплинков занят несколькими ВМ, генерирующими много трафика, а остальные – несколькими ВМ, генерирующими мало трафика;

- ❑ **Route based on source MAC hash** – балансировка по MAC-адресу источника. В этом случае трафик делится на основе MAC-адреса источника пакета. Таким образом, исходящий трафик делится точно так же, как и в случае балансировки по портам. На практике метод практически не применяется;
- ❑ **Route based on ip hash** – балансировка по хешу (контрольной сумме) IP. Здесь критерием разделения трафика считается пара IP-источника – IP-получателя. Таким образом, трафик между одной ВМ и разными клиентами, в том числе за маршрутизатором, может выходить по разным vmpic. Этот метод балансировки нагрузки является самым эффективным, однако он может вызывать большую нагрузку на процессоры сервера, так как именно на них будет вычисляться хеш IP-адресов для каждого пакета.

Также этот метод балансировки требует настроенной группировки портов (известной как link aggregation, EtherChannel, Ethernet trunk, port channel, Multi-Link Trunking) на физическом коммутаторе, к которому подключен коммутатор виртуальный. Это вызвано тем, что при данном методе балансировки нагрузки MAC-адрес одной ВМ может одновременно числиться на нескольких vmpic, как следствие – на нескольких портах коммутатора физического. Что не является допустимым в штатном режиме работы, без группировки портов.

В обратную сторону – если вы хотите настроить группировку портов между физическим и виртуальным коммутаторами, то вы настраиваете ее на физическом; а на виртуальном ставите балансировку по хешу IP для нужных групп портов – и все.

Последний нюанс: если у вас один коммутатор виртуальный подключен к нескольким физическим (из соображений отказоустойчивости), то физические коммутаторы должны быть настроены на использование группировки портов (Etherchannel, 802.3ad и т. п.) в так называемом режиме «стека», то есть работая вместе (это обычно называется Multichassis Etherchannel или, для не Cisco, MLAG – Multichassis Link Aggregation). Далеко не с любыми физическими коммутаторами получится использовать этот тип балансировки нагрузки в такой конфигурации.

ESXi не поддерживает автоматической настройки группировки портов с помощью Link Aggregation Control Protocol (LACP) или Port Aggregation Protocol (PAgP).

Link Aggregation (Etherchannel) на физическом коммутаторе должен быть настроен, только если на виртуальном коммутаторе используется балансировка нагрузки по IP.

Обратите внимание. Для определения того, на какой аплинк пойдет очередная «сессия» трафика, в алгоритме ip hash load balancing используется хеш-функция пары IP-источника – IP-назначения. В некоторых случаях разные комбинации IP-адресов могут обладать одним и тем же хешем, что сведет эффективность балансировки к нулю. См. статью базы знаний <http://kb.vmware.com/kb/1007371>, или перевод <http://link.vm4.ru/iphash>.

Резюмирую. Для использования данного механизма балансировки нагрузки для виртуального коммутатора (или любой, даже одной группы портов на нем) следует:

1. Настроить группировку портов по стандарту Link Aggregation (или аналогичному) на портах физического коммутатора/коммутаторов, к которым подключены аплинки ESXi.

Стандарт группировки портов у разных производителей может носить разные названия. В нашем контексте вместо Link Aggregation могут быть использованы стандарты Ether-Channel, Ethernet trunk, port channel, Multi-Link Trunking (список может быть неполным). В общем, какой-нибудь стандарт группировки портов.

Следует настроить именно статичную группировку. Протоколы автоматической настройки группировки портов, такие как LACP и PAgP, не поддерживаются ESXi.

Если ESXi подключен больше чем к одному коммутатору, то группировку портов следует настроить, объединив эти коммутаторы в так называемый «стек». Если физические коммутаторы не поддерживают такого режима работы, то данный алгоритм балансировки нагрузки использовать будет нельзя.

2. Для этой группы портов следует указать алгоритм балансировки нагрузки. Обязательно должен быть выбран алгоритм балансировки нагрузки по IP-источника – IP-получателя (IP-Source-Destination).
3. После выполнения этих настроек на физическом сетевом оборудовании нужно зайти в настройки виртуального коммутатора (или группы портов) и выбрать в верхнем выпадающем меню тип балансировки нагрузки = **Route based on ip hash**.

Возможно, вам будет полезно ознакомиться с посвященной теме балансировки нагрузки записью в моем блоге – <http://link.vm4.ru/loadbalance>.

В распределенных коммутаторах VMware (начиная с версии 4.1) появился еще один тип балансировки нагрузки – **Route based on physical NIC load**. Этот метод балансировки нагрузки доступен только для распределенных коммутаторов. Суть данного механизма напоминает работу первого варианта балансировки – по Port ID. Однако есть и значительные различия. Во-первых, при принятии решения о том, через какой pNIC выпускать очередную «сессию», выбор осуществляется в зависимости от нагрузки на этот pNIC, а не случайным образом. Во-вторых, выбор повторяется каждые 30 секунд (в то время как во всех прочих вариантах однажды осуществленный выбор не меняется до выключения VM).

Резюме: рекомендуемым в общем случае является **Route based on the physical NIC load** – по совокупности характеристик. Он осуществляет балансировку нагрузки с минимальными накладными расходами (но использовать этот метод балансировки возможно только на распределенных коммутаторах, то есть обладая поддерживающей их лицензией vSphere). В случае если вы твердо уверены, что вам необходима большая эффективность балансировки, используйте **Route based on ip hash**. Пример такой ситуации – одна VM, генерирующая большой объем трафика и работающая с большим количеством клиентов. Однако если нет возможности настроить etherchannel на физическом коммутаторе, то Route based on ip hash использовать невозможно.

Если не подходят и Route based on ip hash, и Route based on physical NIC load, используйте **Route based on the originating port ID**.

Более эффективную балансировку нагрузки рекомендуется ставить лишь для той группы портов, для которой она необходима, – с целью свести к минимуму накладные расходы в виде нагрузки на CPU сервера.

2.4.6. Cisco Discovery Protocol, CDP и Link Layer Discovery Protocol (LLDP)

CDP – протокол от Cisco, позволяющий обнаруживать и получать информацию о сетевых устройствах. ESXi 5 поддерживает этот протокол и для стандартных, и для распределенных виртуальных коммутаторов.

LLDP – вендорнезависимый протокол для решения тех же самых задач. Поддерживается начиная с пятой версии ESXi и только для распределенных виртуальных коммутаторов.

Настройка CDP для стандартных виртуальных коммутаторов

Чтобы изменить настройки CDP для стандартных вКоммутаторов, вам понадобится командная строка. Команда

```
esxcfg-vswitch -b <vSwitch>
```

покажет текущую настройку CDP для вКоммутатора <vSwitch>.

Команда

```
esxcfg-vswitch -B <mode> <vSwitch>
```

поможет изменить значение настройки CDP для вКоммутатора <vSwitch>. Доступные значения параметра <mode>:

- ☐ Down – CDP не используется;
- ☐ Listen – ESXi получает и отображает информацию о коммутаторах Cisco, к которым подключен. На коммутаторы информация о вКоммутаторах не отправляется;
- ☐ Advertise – ESXi отправляет информацию о вКоммутаторах наружу, но не принимает и не отображает информацию о физических коммутаторах;
- ☐ Both – ESXi и обнаруживает подключенные физические коммутаторы, и отправляет на них информацию о коммутаторах виртуальных.

Когда CDP настроен в listen или both, нам доступна информация о коммутаторах Cisco. Для просмотра пройдите **Configuration** ⇒ **Networking** ⇒ иконка справа от vSwitch (рис. 2.34).

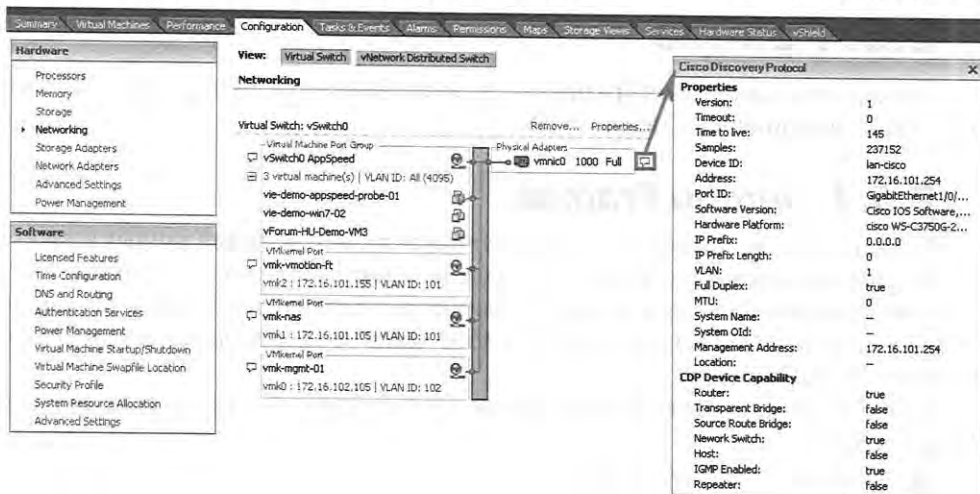


Рис. 2.34. Просмотр информации от CDP

Настройка CDP и LLDP для распределенных виртуальных коммутаторов

Для распределенных коммутаторов эта настройка выполняется из графического интерфейса.

Пройдите **Home** ⇒ **Inventory** ⇒ **Networking** ⇒ контекстное меню распределенного вКоммутатора ⇒ **Edit Settings** ⇒ строка **Advacend**.

- ☐ **Status** – включено ли использование протокола обнаружения;
- ☐ **Type** – какой из протоколов использовать;
- ☐ **Operation** – режим работы:
 - **Listen** – ESXi обнаруживает и показывает информацию о коммутаторах, к которым подключен. На коммутаторы информация о вКоммутаторах не отправляется;
 - **Advertise** – ESXi отправляет информацию о вКоммутаторах наружу, но не слушает и не принимает информацию о физических коммутаторах;
 - **Both** – ESXi и обнаруживает подключенные физические коммутаторы, и отправляет на них информацию о коммутаторах виртуальных;
- ☐ **Administrator Contact Information** – это информационные поля, информация отсюда будет сообщаться внешним коммутаторам при настройке Advertise или Both.

2.5. Разное

Несколько слов о разнообразных отдельных функциях, таких как Jumbo Frames, TSO, генерация MAC-адресов VM.

2.5.1. Jumbo Frames

Функция Jumbo Frames позволяет увеличить размер поля для данных в пакете IP. Получается, что мы тем же числом пакетов (то есть с теми же накладными расходами) передаем больше полезной информации. Если в стандартном IP-пакете поле для данных (MTU) имеет размер 1500 байт, то при использовании Jumbo Frames – до 9000 байт.

Jumbo Frames должны поддерживаться всеми узлами сети, то есть должны быть включены на:

- ☐ физических коммутаторах;
- ☐ виртуальных коммутаторах или распределенных виртуальных коммутаторах;
- ☐ а также в физических и виртуальных серверах (и прочих системах, например системах хранения данных).

Jumbo Frames могут использоваться виртуальными машинами и портами VMkernel для трафика NFS, iSCSI, Fault Tolerance и vMotion. Для начала использования Jumbo Frames нам необходимо включить их поддержку на физических коммутаторах, затем для vSwitch/dvSwitch, а далее настроить их использование внутри VM или для виртуального контроллера VMkernel.

В пятой версии ESXi эти настройки выполняются практически одинаково и для стандартных, и для распределенных виртуальных коммутаторов, и выполняются из графического интерфейса.

Чтобы включить Jumbo Frames для стандартного виртуального коммутатора, пройдите **Home** ⇒ **Hosts and Clusters** ⇒ вкладка **Configuration** для выбранного сервера ESXi ⇒ **Networking** ⇒ **Properties** для выбранного стандартного вКоммутатора ⇒ **Edit** для коммутатора на вкладке **Ports** ⇒ **MTU**. Указываем размер поля для данных. Чаще всего используется максимальный – 9000.

Чтобы включить Jumbo Frames для распределенного виртуального коммутатора, пройдите **Home** ⇒ **Networking** ⇒ в контекстном меню dvSwitch пункт **Edit Settings** ⇒ **Advanced** ⇒ **Maximum MTU**. Указываем размер поля для данных. Чаще всего используется максимальный – 9000.

Итак, первый шаг – включение поддержки Jumbo Frames на виртуальных коммутаторах – вы сделали. Шаг номер два – включить эту функцию на VM и/или на интерфейсах VMkernel.

Настройка Jumbo Frames для виртуальных машин

Чтобы использовать Jumbo Frames с VM, в качестве гостевых ОС должны использоваться Windows Server (2003 или 2008, Enterprise или Datacenter Edition), Red Hat Enterprise Linux 5.0, SUSE Linux Enterprise Server 10. Тип виртуального

сетевого адаптера должен быть vmxnet2 или vmxnet3. В документации VMware написано «Для включения Jumbo Frames смотрите документацию гостевой ОС». Но для Windows это делается примерно так, как показано на рис. 2.35.

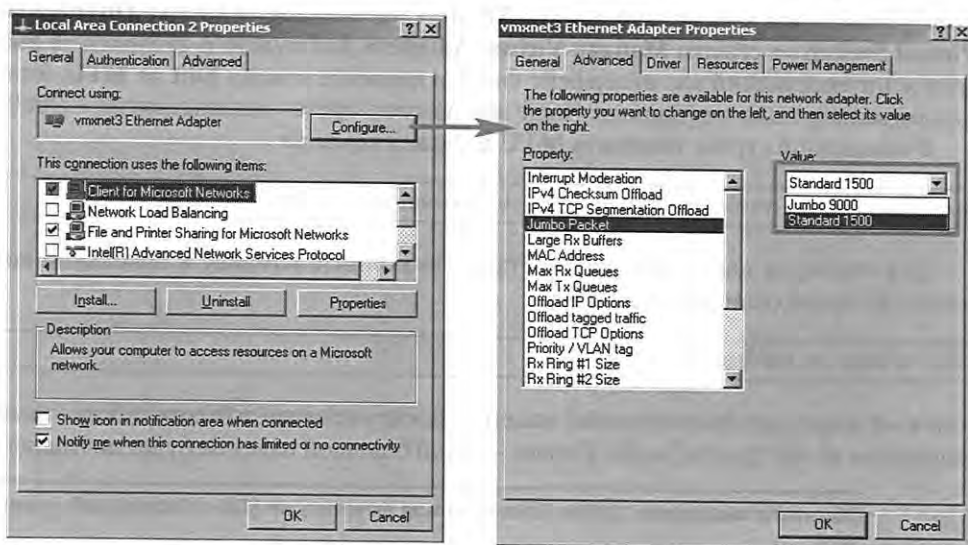


Рис. 2.35. Настройки Jumbo Frames в драйвере vmxnet3

Для проверки работы Jumbo Frames отправьте большой пакет на ту систему, для общения с которой вы настраиваете Jumbo Frames (напомню, что пинговать VM на этом же сервере ESXi может оказаться плохой идеей – общение между VM на одном вКоммутаторе и в одном vlan остается внутри ESXi, такая проверка не затронет физическую сеть):

```
ping -f -l 8972 <IP удаленной системы>
```

Ключ `-f` запрещает фрагментацию пакетов. Так что если где-то в сети между этими машинами не настроены Jumbo Frames – такой большой пакет не будет доставлен.

Настройка Jumbo Frames для VMkernel

Для использования Jumbo Frames с интерфейсами VMkernel необходимо включить эту функцию на них.

Для ESXi 5 версии это возможно из графического интерфейса. Вам потребуется зайти в свойства интерфейса VMkernel и увеличить параметр MTU.

Если нужный вам интерфейс VMkernel подключен к стандартному виртуальному коммутатору, то пройдите **Home** ⇒ **Hosts and Clusters** ⇒ вкладка **Configuration** для выбранного сервера ESXi ⇒ **Networking** ⇒ **Properties** для выбранного стандартного вКоммутатора ⇒ **Edit** для группы портов VMkernel на вкладке **Ports**

⇒ **MTU**. Указываем размер поля для данных. Чаще всего используется максимальный – 9000.

Если нужный вам интерфейс VMkernel подключен к распределенному виртуальному коммутатору, то пройдите **Home** ⇒ **Hosts and Clusters** ⇒ вкладка **Configuration** для выбранного сервера ESXi ⇒ **Networking** ⇒ кнопка **Distributed Virtual Switch** ⇒ ссылка **Manage Virtual Adapters** для нужного dvSwitch ⇒ выделите интересующий вас интерфейс vmk# и нажмите ссылку **Edit** ⇒ **MTU**. Указываем размер поля для данных. Чаще всего используется максимальный – 9000.

В командной строке изменить MTU поможет команда

```
esxcli network ip interface set -m 9000 -i <имя интерфейса vmkernel вида vmk#>
```

Для проверки настройки Jumbo Frames выполните команду в локальной командной строке (или SSH) для ESXi

```
ping -s 8000 -d <IP удаленной системы>
```

Ключ `-d` запрещает фрагментацию пакетов. Так что если где-то в сети между этими машинами не настроены Jumbo Frames – такой большой пакет не будет доставлен.

Обратите внимание. Jumbo Frames нельзя включить на ESXi с бесплатной лицензией.

Jumbo Frames имеет смысл использовать для интерфейсов VMkernel, задействованных под любые задачи. Единственное исключение – трафик управления ESXi, где, скорее всего, эта настройка не будет иметь смысла.

2.5.2. TSO – TCP Segmentation Offload, или TOE – TCP offload engine

TOE (TCP offload engine) – функция физического сетевого контроллера, когда часть работы по обработке стека TCP/IP, такая как формирование и подсчет контрольной суммы пакетов, выполняется не службой в ОС, а самим контроллером. Часть этого механизма – TSO, TCP Segmentation Offload, функция также известна как «large segment offload», или LSO. TSO позволяет обрабатывать большие пакеты (до 64 Кб) при любом размере MTU, формируя из большого пакета большее количество пакетов меньшего размера.

В документации VMware обычно употребляется термин TSO, прочие названия приведены для справки.

Включение этой функции позволяет снизить нагрузку на процессоры сервера и повысить скорость работы сети. Заметная разница в нагрузке на процессоры сервера будет, скорее всего, лишь в инфраструктурах со значительным сетевым трафиком.

Формально мы можем задействовать эту функцию для трафика ВМ и VMkernel. «Формально» – потому, что мне встречались утверждения инженеров VMwa-

ге, что в vSphere (в первых версиях, по крайней мере) сетевые контроллеры с TSO работают, но для трафика VM TSO не используется, так как внутренние тесты не показали значимой эффективности на разнообразных задачах (см. <http://communities.vmware.com/thread/217825>). Для трафика VM эту функцию можно задействовать двумя способами:

- ❑ используя в качестве виртуального сетевого контроллера контроллеры типа vmxnet 2 или vmxnet 3. То есть при использовании vNIC этого типа TSO будет использоваться, если физическое оборудование обладает поддержкой данной функции;
- ❑ прокинув физический сетевой контроллер в VM с помощью функции VMDirectPath.

Может потребоваться включение TSO в BIOS сетевого контроллера. Обратите внимание, что если контроллер с поддержкой TSO значится в списке совместимости ESXi, то это означает, что ESXi заработает с этим сетевым контроллером, но не гарантирует работу с его функциями TSO. Если вас интересует именно функционал TSO, то совместимость контроллера с ESXi нужно проверять именно с упором на TSO (по документации к сетевому контроллеру).

Для интерфейсов VMkernel TSO включен по умолчанию. Проверить это можно, выполнив команду

```
esxcfg-vmknic -l
```

Если в столбце TSO MSS значение 65535, то TSO включен. Если он выключен, то единственный способ его включить – пересоздать интерфейс, указав принудительное использование TSO параметром командной строки при создании.

Выключить использование TSO для ESXi можно через расширенные настройки. Пройдите в настройки сервера: **Configuration** ⇒ **Advanced Settings** для Software ⇒ Net ⇒ настройка UseHwTSO. Вам нужно присвоить значение нуль.

Скорее всего, выключение может потребоваться лишь в случае проблем с использованием этой функции с вашими сетевыми контроллерами. В случае проблем перед отключением TSO обновите прошивку контроллера и ESXi и драйвер для контроллера.

2.5.3. VMDirectPath

Функция VMDirectPath позволяет выделять в приватное пользование VM контроллер в слоте PCI сервера. Таким контроллером может быть сетевая карта. Подробности см. в разделе про компоненты VM.

2.5.4. Standalone (отдельные) порты

На распределенном виртуальном коммутаторе виртуальные машины подключаются к группам портов и к портам с конкретным номером. Например, в свойствах VM вы можете увидеть возможность подключения виртуального сетевого контроллера к конкретному порту – рис. 2.36.

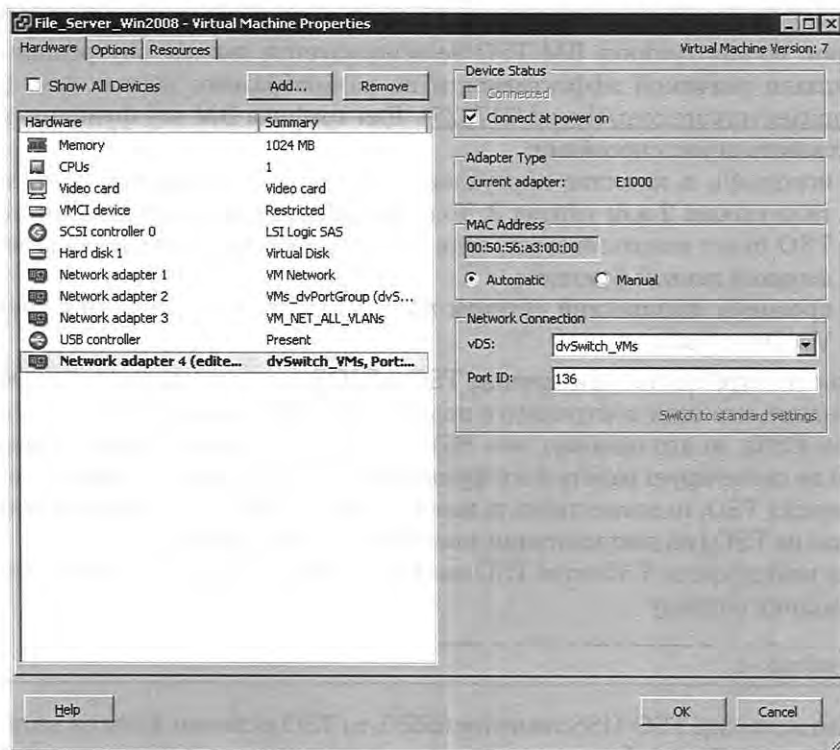


Рис. 2.36. Настройка сетевого подключения VM к отдельному порту dvSwitch

Это может быть важно по той причине, что для распределенных виртуальных коммутаторов изменять настройки (такие как VLAN, traffic shaping, security и др.) можно и для отдельного порта.

2.6. Рекомендации для сети

Рекомендации в общем случае следующие.

Разделяйте разные виды трафика для повышения безопасности и/или производительности. Разные виды трафика – это:

- управляющий трафик, то есть интерфейс(ы) VMkernel с флажком **management network**. Изоляция управляющего трафика весьма важна с точки зрения безопасности, ибо компрометация ESXi означает компрометацию всех работающих на нем VM. Несмотря на то что управление вроде бы не предполагает большого трафика, тем не менее именно по сети управления осуществляются многие задачи, могущие вызвать всплески трафика. Например:
 - конвертация физического сервера или VM с lheujj ubgthdbpjhf в виртуальную машину на ESXi;



- функция NetFlow;
- в некоторых случаях развертывание из шаблона или клонирование (когда хранилище, на котором создается новая ВМ, недоступно по сети хранилища для ESXi, где расположен шаблон);
- ❑ трафик vMotion. Более того, выделенная гигабитная сеть (в смысле выделенный гигабитный физический контроллер) под vMotion – это обязательное условие для того, чтобы конфигурация была поддерживаемой. Плюс к тому трафик vMotion передается незашифрованным. Получается, что его перехват потенциально дает доступ к содержимому оперативной памяти мигрируемой ВМ или возможность изменить это содержимое;
- ❑ трафик Fault Tolerance. Более того, выделенный гигабитный контроллер под Fault Tolerance – это обязательное условие для того, чтобы конфигурация была поддерживаемой;
- ❑ трафик IP-систем хранения – iSCSI/NFS;
- ❑ разумеется, трафик ВМ. Притом деление может быть и среди них, так что если у вас есть группа ВМ, трафик от которых может быть большим или к безопасности которого есть особые требования, имеет смысл подумать об изоляции их трафика.

Для разделения трафика используются VLAN или разграничение на уровне физических сетевых контроллеров. Изоляция на уровне VLAN удобнее и не требует большого количества сетевых контроллеров, но в редких случаях может быть неприменимой из таких соображений, как паранойя отдела безопасности, требования использовать сертифицированное оборудование под работу с VLAN или особенностями имеющейся физической сети.

Разграничение на уровне сетевых контроллеров следует делать созданием отдельных виртуальных коммутаторов под разные задачи.

В любом случае нельзя пренебрегать типичными для неvirtуализированных систем средствами сетевой безопасности, такими как межсетевые экраны. Напомним, что у VMware есть собственное решение с таким функционалом – VMware vShield Zones (<http://www.vmware.com/products/vshield-zones>).

Также сегодня на рынке существуют сторонние решения, имеющие отношение к безопасности сети. В частности, решения, обеспечивающие функционал обнаружения и предотвращения вторжений (intrusion-detection system, IDS и intrusion-prevention systems, IPS). Обзорную информацию про некоторые из таких продуктов вы можете получить по ссылке <http://link.vm4.ru/sec>.

Глава 3. Системы хранения данных и vSphere

Как администраторы vSphere мы являемся потребителями дисковых ресурсов, предоставляемых системами хранения данных. Поэтому я буду про них рассказывать в потребительском ключе – что надо потребовать от администратора системы хранения данных, о чем надо знать для этого. Само администрирование СХД, то есть «как делается создание LUN (или volume)», «как делается их “презентование” (presentation)» и т. п., – об этом говорить не буду ничего.

Сначала выскажу соображения по выбору системы хранения. Затем – подробности по использованию и настройке работы с СХД разных типов. Потом – про некоторые специфические особенности и функции ESXi в области работы с системами хранения. Ну а прямо сейчас – пара слов о специфике работы ESXi с дисковой системой (рис. 3.1):

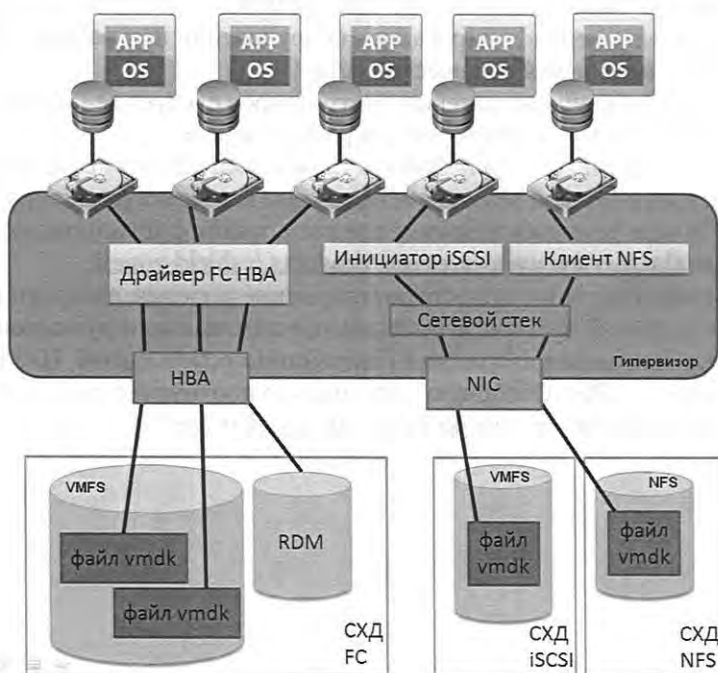


Рис. 3.1. Схема работы VM с дисковой подсистемой



На этом рисунке вы видите несколько виртуальных машин, к которым подключены дисковые ресурсы. Для всех ВМ, кроме третьей, диск является виртуальным. В том смысле, что содержимое такого диска находится в файле типа «Virtual Machine Disk» (*.vmdk). Такой файл размещается на разделе, отформатированном в файловую систему «VMware File System», VMFS. В VMFS форматируются дисковые ресурсы (LUN) блочных систем хранения (на рисунке это системы хранения FC и iSCSI).

Обратите внимание. Хочу сделать примечание для тех читателей, которые не имеют опыта работы с системами хранения данных. Я буду активно использовать термин LUN, особенно в этой главе. LUN – это то, что выглядит как диск с точки зрения сервера. Мы, потребители дисковых ресурсов, воспринимаем LUN как диск. Однако с точки зрения СХД LUN – это не физический диск и даже не RAID-массив. LUN – это логический объект, создаваемый по требованию администратором системы хранения.

Кроме того, к ESXi может быть подмонтирован сетевой ресурс по протоколу NFS – и на таком сетевом ресурсе также могут быть размещены файлы виртуальных машин.

На разделах VMFS и NFS могут быть расположены файлы виртуальных дисков сразу нескольких других ВМ, файлы настроек ВМ и файлы иных типов.

Диском третьей виртуальной машины является весь диск (LUN), который подключен к серверу ESXi и затем подключен напрямую к этой ВМ. Такое подключение называется RDM, Raw Device Mapping. Гостевая ОС обращается с этим LUN так же, как если бы была установлена на физический сервер, которому презентован данный LUN. Однако гипервизор скрывает от гостевой ОС тип системы хранения – и даже в случае RDM гостевая ОС воспринимает свой RDM-диск как локальный SCSI-диск. RDM-подключение возможно для дисков блочных систем хранения. Блочные системы хранения – это СХД, построенные по архитектуре SAN, с подключением по протоколам FibreChannel, iSCSI, FCoE, SAS.

HBA – это Host Bus Adapter, контроллер в сервере, через который идет обращение к системе хранения. Это могут быть Fibre Channel HBA для доступа к Fibre Channel SAN, локальный контроллер RAID для доступа к локальным дискам и др. Вне зависимости от того, какой контроллер (и СХД какого типа) используется, любая ВМ всегда видит свои диски как локальные диски SCSI, подключенные к локальному контроллеру SCSI. Этот контроллер SCSI гипервизор создает для ВМ наравне с прочим виртуальным оборудованием (единственный нюанс – ESXi 5 может подключать виртуальные диски к контроллеру IDE, а не SCSI). Доступом именно к СХД обладает только сам гипервизор, на рис. 3.1 показанный как «VMware virtualization layer». Гипервизор не сообщает ВМ о том, что ее диском является файл, расположенный на локальном диске сервера; или файл на сетевом ресурсе NFS; или LUN на iSCSI, подключенный как RDM. ВМ всегда видит своим диском локальный диск SCSI (или IDE) – и изнутри нее нельзя понять, СХД какого типа используется. Когда гостевая ОС адресует команды SCSI своему диску, эти обращения перехватываются гипервизором и преобразуются в операции с файлами на VMFS. Более подробно про VMFS (файловую систему для

хранения виртуальных машин) и RDM (прямой доступ к диску) расскажу после повествования о типах СХД.

3.1. Обзор типов СХД

Для начала поговорим о том, какие бывают и что из себя представляют разные типы систем хранения данных.

Итак, мы можем воспользоваться:

- ❑ SAN, Storage Area Network – данный тип систем хранения предполагает блочный доступ. Возможен доступ к одним и тем же ресурсам нескольких серверов одновременно.

«Блочный доступ» означает, что сервер видит диск. Совокупность блоков. И именно сервер создает на этом диске файловую систему. Однако команды, которые сервер отправляет на СХД, представляют из себя команды по работе с блоками, вида, грубо, «прочитай блок номер 5001, запиши такие-то данные в блок 456».

В качестве среды передачи данных может использоваться Fibre Channel и iSCSI (самые популярные сегодня варианты для SAN), но также возможны подключения по интерфейсам/протоколам Fibre Channel over Ethernet (FCoE) и SAS;

- ❑ NAS, Network Attached Storage – системы хранения этого типа предоставляют доступ на уровне файлов. Возможен доступ к одним и тем же ресурсам нескольких серверов одновременно.

«Файловый доступ» означает, что файловая система создана самой системой хранения.

ESXi поддерживают NAS только с протоколом NFS;

- ❑ DAS, Direct Attached Storage – данный тип систем хранения предполагает блочный доступ. Основная особенность системы хранения этой архитектуры – доступ к одним и тем же дисковым ресурсам нескольких серверов одновременно невозможен. Это чрезвычайно снижает интерес к ним для инфраструктур vSphere. Обычно СХД подобного типа представлены локальными дисками.

Системы хранения данных разных типов отличаются по следующим характеристикам:

- ❑ стоимость. Очевидно, сравнение СХД по этому параметру выходит за пределы книги;
- ❑ функциональность самой системы хранения, например возможность создания снимков состояния (snapshot) на уровне СХД. Про это тоже говорить здесь бессмысленно, потому что в каких-то решениях эти средства будут востребованы, а в каких-то – нет. Также различием являются функциональные особенности того или иного подхода, например файловый доступ на NAS или блочный на SAN;
- ❑ производительность. Производительность – непростое понятие, много от каких факторов зависящее. Думаю, оправданно будет сказать, что главным

отличием в производительности СХД разных типов является скорость используемого интерфейса/среды передачи данных. Впрочем, темы производительности дисковой подсистемы я касался в первой главе;

- ❑ функционал относительно vSphere – а вот по этому поводу поговорим подробнее сейчас.

Сравнение по поддерживаемым функциям я попытался отразить в табл. 3.1.

Таблица 3.1. Сравнение функционала СХД для vSphere

Тип СХД	Загрузка ESXi	Включение ВМ	VMotion sVMotion	HA DRS FT	API for Data Protection	VMFS RDM	MSCS MFC	SIOC
Fibre Channel	+	+	+	+	+	+	+	+
iSCSI	+	+	+	+	+	+	–	+
NAS	–	+	+	+	+	–	–	+
DAS	+	+	– / +	–	+	+	+ / –	–

Примечания к таблице:

- ❑ загрузка ESXi с iSCSI системы хранения возможна только с использованием аппаратного инициатора или сетевого контроллера, поддерживающего «iSCSI boot»;
- ❑ если оба узла кластера MSCS/MFC работают на одном ESXi, то СХД может быть любой; MSCS/MFC-кластер технически возможно реализовать с системой хранения iSCSI, но такая конфигурация не будет поддерживаться VMware, если система хранения подключена к ESXi. Однако решение будет поддерживаемым, если требуемые для работы MFC-кластера LUN подключить к программным iSCSI-инициаторам гостевых ОС;
- ❑ VMFS и RDM показаны в одном столбце потому, что они являются альтернативами друг другу. Дисковые ресурсы (LUN) системы хранения с блочным доступом ESXi может *или* отформатировать в VMFS, *или* подключить к виртуальной машине как RDM;
- ❑ SIOC расширяется как Storage IO Control, механизм контроля производительности СХД относительно виртуальных машин. Подробнее о нем рассказывается в главе 6.

Обратите внимание, что основные функции доступны для СХД любого типа.

Таким образом, если стоит вопрос по выбору системы хранения, то примерный план может быть таким:

1. Определиться с необходимыми функциями (в первую очередь имеется в виду табл. 3.1). Например, если нам требуется RDM, то NAS не подойдет. С другой стороны, у систем хранения может быть свой собственный интересный для вас функционал, например упрощающий резервное копирование, дедупликация, возможности кеширования и др.; и эти независимые от vSphere функции могут быть решающими при выборе.

2. Определиться с необходимой производительностью. О чем стоит задуматься и что стоит принять во внимание, см. в первой главе в разделе, посвященном сайзингу.
3. Если после пп. 1 и 2 выбор типа системы хранения еще актуален, сравниваем варианты по цене.

В тексте вам уже встречался и будет встречаться термин **Datastore**, или **Хранилище**. Этим термином обозначается раздел файловой системы, на котором ESXi способен располагать файлы виртуальных машин. Бывают VMFS-хранилища и NFS-хранилища. Если нашему серверу доступен диск (не важно, локальный диск, локальный RAID-массив или LUN с системы хранения данных), то это просто диск/LUN. А вот когда мы отформатируем этот диск/LUN в файловую систему VMFS – у нас появится VMFS-хранилище. NFS-хранилище появляется после подмонирования NFS-экспорта к ESXi.

3.2. DAS

Direct Attached Storage, DAS – самый простой тип систем хранения. Характерной чертой этого типа хранилищ является то, что их дисковые ресурсы доступны лишь одному серверу, к которому они подключены. Наиболее характерный пример – локальные диски сервера. Да, да – их можно считать хранилищем DAS. Те DAS, что являются отдельными устройствами, часто представляют из себя просто корзину (Enclosure) для дисков в отдельном корпусе. В сервере стоит контроллер SAS или SCSI, к которому и подключается полка DAS. Однако даже если вы используете СХД Fibre Channel, но пара серверов подключена к ней без использования коммутатора FC – некоторые (как правило, старые) модели СХД не смогут выделить один и тот же LUN сразу обоим серверам. Такая конфигурация полностью подпадает под определение DAS.

Для администраторов ESXi этот тип СХД обычно малоинтересен, потому что подробные системы не предоставляют доступа к файлам одной ВМ нескольким серверам:

- ❑ в мало-мальски крупных внедрениях это необходимо для живой миграции (VMware VMotion), для автоматической балансировки нагрузки (VMware DRS), для функций повышения доступности VMware HA и VMware FT;
- ❑ если речь идет про небольшие инфраструктуры, где эти функции и так не предполагаются к использованию, то ситуация примерно следующая: допустим, у нас есть несколько ESXi. Если ВМ расположены на каком-то разделяемом хранилище, то администратор может пусть вручную, но очень быстро и с минимальными усилиями переносить ВМ между серверами. Если же у нас только DAS (например, несколько больших дисков локально в сервере) – то перенос ВМ возможен лишь по сети, что медленно. А если выйдет из строя сервер, к которому подключен DAS, то ВМ будут недоступны.

По сути, DAS используются в тех случаях, когда на разделяемое хранилище с достаточной производительностью нет бюджета, ну или под наши задачи не нуж-

ны ни живая миграция, ни высокая доступность, ни прочие «продвинутые» функции. Возможен вариант, когда мы используем под ESXi сервера с локальными дисками, но программно реализуем доступ к этим дисковым ресурсам по iSCSI или NFS. Обычно соответствующая служба работает внутри ВМ, которая доступные для нее дисковые ресурсы ESXi предоставляет ему же, но уже по iSCSI/NFS. Кстати, в пятой версии vSphere был представлен продукт VMware Virtual Storage Appliance, задачей которого как раз и является реализация программного NAS(NFS) хранилища из локальных дисков двух или трех серверов ESXi. См. соответствующий раздел.

А еще есть возможность с помощью сторонних программных средств реализовать репликацию файлов ВМ между серверами. В таком случае мы получим дешевую инфраструктуру без СХД, с одной стороны, но с достаточно высоким уровнем доступности – в случае выхода из строя сервера есть возможность запустить реплику ВМ с него на другом сервере, с потерей данных с момента последней репликации.

Впрочем, реализация такого рода вариантов, с привлечением программного обеспечения третьих фирм, в данной книге рассматриваться не будет.

Кто-то может заметить, что на рынке присутствуют модели систем хранения, которые предполагают соединение с серверами по интерфейсу SAS, но позволяют множественный доступ. Такие варианты, по данной классификации, имеет смысл отнести к SAN, ключевой аспект тут – множественный доступ. Разные сервера ESXi могут обращаться на один и тот же LUN одновременно.

3.3. NAS (NFS)

Network Attached Storage, NAS – устройство хранения, подключенное к сети. Этот тип систем хранения также весьма несложен. Его характерные черты:

- ❑ доступ к дисковым ресурсам по локальной сети. На стороне сервера требуются обычные сетевые контроллеры;
- ❑ доступ к одним и тем же дисковым ресурсам возможен одновременно с нескольких серверов;
- ❑ доступ на уровне файлов – то есть файловая система создается и обслуживается системой хранения, а не сервером. Это ключевое отличие NAS от DAS и SAN, где доступ к СХД идет на уровне блоков и сервера имеют возможность использовать собственную файловую систему.

Характерный пример – файловый сервер. Он предоставляет по сети доступ на уровне файлов к разделяемой (Shared) папке сразу многим клиентам (серверам). Однако есть и специализированные системы хранения, использующие собственную ОС. Весьма сильные позиции в этом сегменте систем – у компании NetApp.

Существуют два основных протокола инфраструктур NAS – NFS для *nix и SMB для систем Windows.

ESXi позволяет запускать ВМ с NAS, использующих протокол NFS.

По сравнению с iSCSI с программным инициатором, NFS вызывает меньшую нагрузку на процессоры сервера.

Если сравнивать NAS с системами хранения других типов по функционалу относительно vSphere, то NFS-система хранения, с одной стороны, обеспечивает весь основной функционал. Такие функции, как VMotion, DRS, HA, FT, работают с хранилищем NFS.

С другой стороны, системы хранения NAS способны обеспечить достаточно интересный дополнительный функционал. На примере систем хранения NetApp будут доступны следующие функции:

- ❑ большой максимальный размер одного хранилища. Если брать NetApp как наиболее применимое NFS-хранилище, то в зависимости от уровня и поколения системы хранения ограничения на размер одного тома данных начинаются от 16 Тб и заканчиваются 100 Тб;
- ❑ дедупликация на уровне системы хранения;
- ❑ кроме увеличения размера хранилища (что возможно и для VMFS), допустимо осуществить уменьшение его размера (чего для VMFS невозможно);
- ❑ снимки состояния (snapshot) средствами системы хранения. Сами по себе снимки – не прерогатива NFS, однако в случае NFS мы получаем гранулярность на уровне отдельных файлов vmdk. То есть в снимке системы хранения у нас будут не LUN со множеством vmdk вместе, а отдельные vmdk. Так как снимок в NetApp доступен как простая копия, сделанная в определенный момент времени, то мы можем просто подключить эту копию по NFS только на чтение. Затем можно осуществить резервное копирование или восстановление файлов vmdk, а также подключить vmdk из снимка к ВМ и восстановить отдельные файлы «изнутри» этого vmdk. То есть при восстановлении одной ВМ из снимка состояния хранилища средствами СХД нет необходимости откатывать все хранилище целиком, со всем его содержимым. Можно восстановить или отдельный виртуальный диск, или даже отдельный файл;
- ❑ vmdk Thin Provisioning – вне зависимости от типа vmdk-файла, с точки зрения ESXi, система хранения сама способна обеспечить thin provisioning;
- ❑ Single-file FlexClone – функция систем хранения в NetApp, в чем-то сходная с работой дедупликации, только для конкретных файлов. Позволяет получить значительную экономию места и скорости развертывания для однотипных виртуальных машин;
- ❑ при использовании репликации как средства повышения доступности инфраструктуры в случае возникновения необходимости переключения ESXi на копию хранилища NFS требуется меньше шагов для этого, и сами шаги более просты. Для FC/iSCSI LUN требуется так называемый «resignaturing», NFS-хранилище же просто подключается без дополнительных условий.

Здесь NetApp упомянут лишь для примера того, какого рода функциями могут обладать системы хранения данного класса.

3.3.1. Настройка и подключение ресурса NFS к ESXi

Для того чтобы подключить дисковые ресурсы по NFS, на стороне ESXi необходимо настроить интерфейс VMkernel, через который и будет передаваться трафик NFS. Схема сети должна быть примерно такой, как на рис. 3.2.

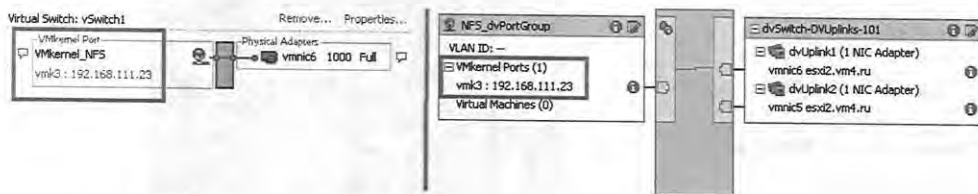


Рис. 3.2. Интерфейс VMkernel на стандартном или распределенном виртуальном коммутаторе

Почему «примерно такой»? Потому что на этом же вКоммутаторе могут располагаться и любые другие виртуальные сетевые контроллеры – и для ВМ, и для VMkernel под другие нужды. А также у этого вКоммутатора может быть больше одного канала во внешнюю сеть (как в правой части рисунка) – это даже рекомендуется, для большей надежности и скорости (multipathing для NFS реализуется через Link Aggregation или через статичные маршруты к разным IP-адресам одной СХД).

Проверить правильность настройки IP, VLAN (если используются) и вообще доступность системы хранения NFS по сети можно следующим образом:

- ☐ или командой ping (в локальной командной строке или ssh)

```
ping <IP NFS сервера>
```

- ☐ или соответствующим пунктом локального меню (рис. 3.3).

После создания интерфейса VMkernel необходимо подключить ресурс NFS. Для этого в настройках сервера пройдите **Configuration** ⇒ **Storage** ⇒ **add Storage**, на первом шаге мастера выберите подключение **Network File System**. На следующем шаге необходимо указать IP или имя системы хранения NFS и имя сетевого ресурса (см. рис. 3.4). В самом нижнем поле вы указываете метку, название хранилища – под этим именем этот ресурс NFS будет отображаться в интерфейсе ESXi и vCenter.

Обратите внимание на флажок **Mount NFS read only**. Он вам пригодится для подключения сетевого ресурса в режиме только чтения. Сам ESXi в любом случае требует разрешений read/write на NFS. Если вы монтируете NFS-хранилище для неизменяемых данных, таких как файлы шаблонов ВМ, образы iso, то единственный способ указать «только чтение» – со стороны самого ESXi вышеупомянутым флажком.

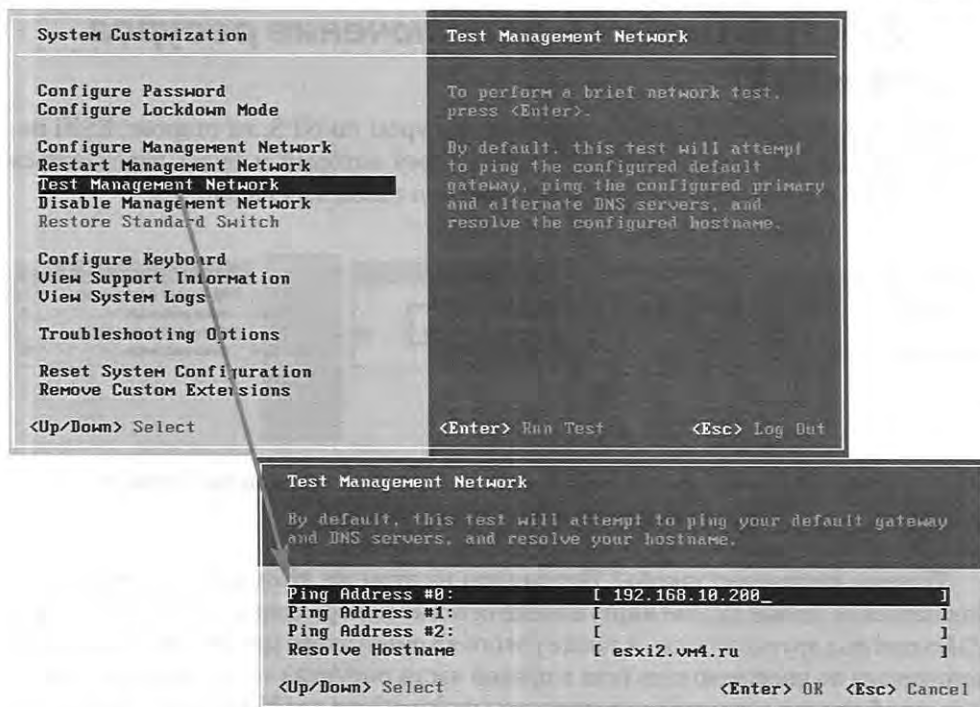


Рис. 3.3. Проверка доступности NFS-сервера из локальной консоли ESXi

Также важно указывать абсолютно идентичные данные при подключении одного и того же NFS-хранилища к разным серверам. Указав в одном случае систему хранения по имени, а в другом – по IP, вы получите дублирование этого хранилища в списках хранилищ интерфейса vSphere. К тому же приведет еще один пример, указание имени «расшаренного» каталога на NFS-хранилище в виде «/esx_nfs» и «/esx_nfs/». Возможно, оптимально будет использовать Host Profiles или сценарий для подключения NFS-хранилищ ко многим серверам ESXi.

Обратите внимание. Избегайте символов, отличных от символов английского алфавита, и цифр в названиях в принципе любых объектов, и особенно в названии файлов. Известны случаи недоступности или нестабильной работы серверов ESXi с NFS-хранилищем, на котором был создан каталог с названием русскими буквами.

Для справки: при использовании Power Shell + PowerCLI для автоматизации настроек vSphere автоматически подключить NFS-хранилище ко всем серверам позволит следующий сценарий:

```
Connect-VIServer <имя или IP-адрес сервера vCenter>
Get-VMHost | New-Datastore -Nfs -Name <имя для отображения> -Path <имя подключаемого каталога> -NfsHost <IP-адрес системы хранения NFS>
```

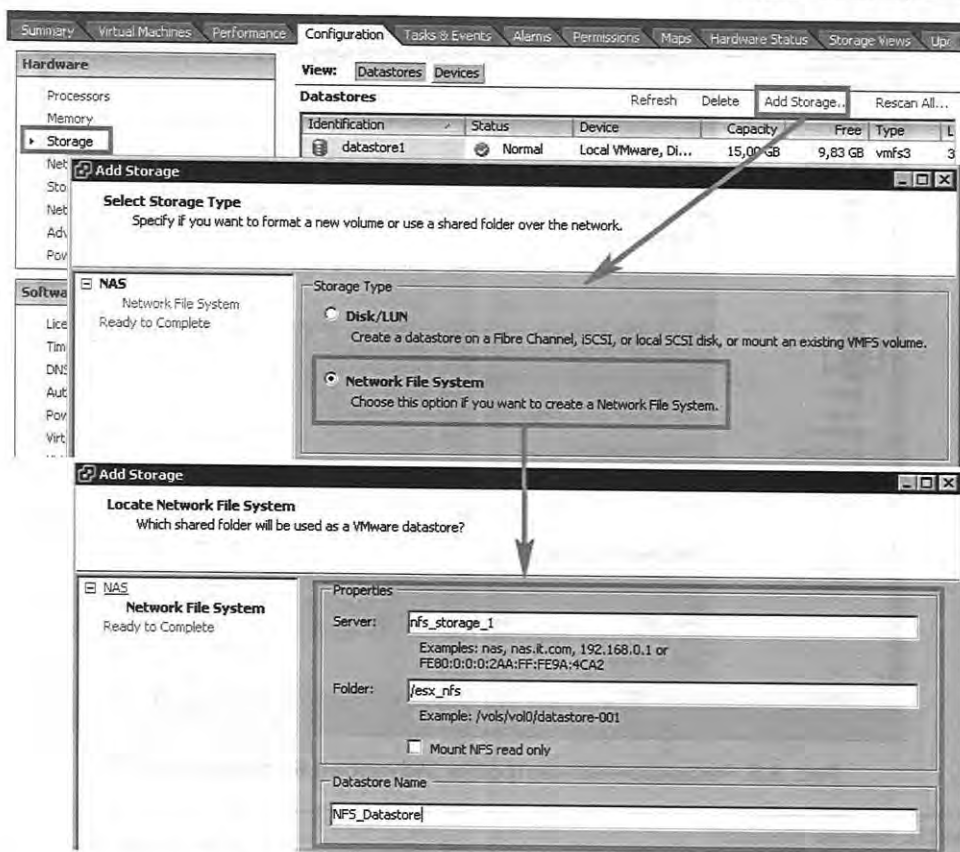


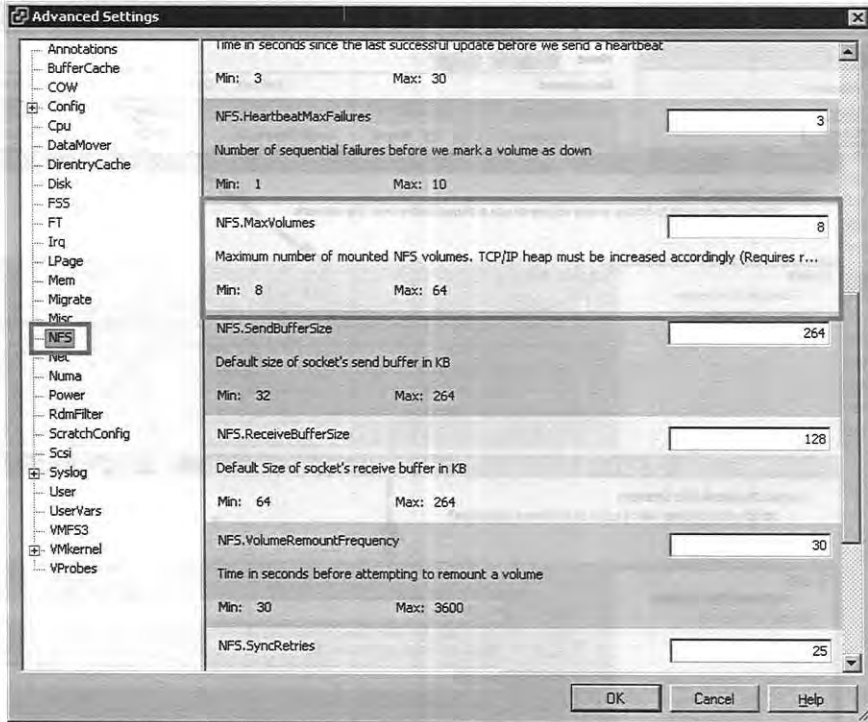
Рис. 3.4. Подключение ресурса NFS к серверу ESXi

По умолчанию вы можете подключить к ESXi до восьми ресурсов NFS. Если вам необходимо больше, перейдите в **Configuration** ⇒ **Advanced Settings** для Software (рис. 3.5). Там вам нужны раздел NFS и настройка **NFS.MaxVolumes**. Описание прочих расширенных настроек для NFS см. в базе знаний VMware, статья <http://kb.vmware.com/kb/1007909>.

Если необходимо отмонтировать NFS-хранилище от сервера ESXi, то достаточно пройти **Configuration** ⇒ **Storage** ⇒ правый клик на отключаемом хранилище ⇒ **Unmount**. Если отключить следует от всех серверов ESXi, то удобнее будет пройти **Home** ⇒ **Datastores** ⇒ правый клик на отключаемом хранилище ⇒ **Unmount**.

Если отключение не произошло (могут быть накладки, если хранилище уже стало недоступно, а для него был включен Storage DRS), то поможет командная строка.

Сначала получим список подключенных nfs-хранилищ и найдем имя отключаемого:

Рис. 3.5. Расширенные настройки (**Advanced Settings**) для NFS

```
esxcli storage nfs list
```

Затем отключим:

```
esxcli storage nfs remove -v [имя хранилища]
```

Обратите внимание. ESXi использует для блокировки файлов не средства NFS, а собственную систему. Когда сервер открывает файл VM на NFS, создается файл с именем вида .lck-XXX, препятствующий открытию этого же файла VM с другого сервера. Не удаляйте файлы .lck-XXX, если не считаете, что файл существует по ошибке (теоретически возможна ситуация, что из-за какого-то сбоя файл блокировки не удален и мы не можем получить доступ к вроде бы выключенной VM).

3.4. SAN, Fibre Channel

Storage Area Network, SAN – сеть хранения данных. Инфраструктура SAN предполагает создание выделенной сети только под данные. В случае Fibre Channel SAN такая сеть строится на оптоволокне. Требуется специальные Fibre Channel коммутаторы и специальные Fibre Channel контроллеры в сервера. Обычно, и

в этой книге в частности, их называют FC HBA, Host Bus Adapter. Как правило, приобретение всего этого плюс покупка и прокладка оптоволокна приводят к удорожанию инфраструктуры FC SAN, по сравнению с другими решениями. К плюсам FC SAN можно отнести все остальное – максимальная производительность, минимальные задержки, полный функционал относительно ESXi.

Схема Fibre Channel SAN показана на рис. 3.6.

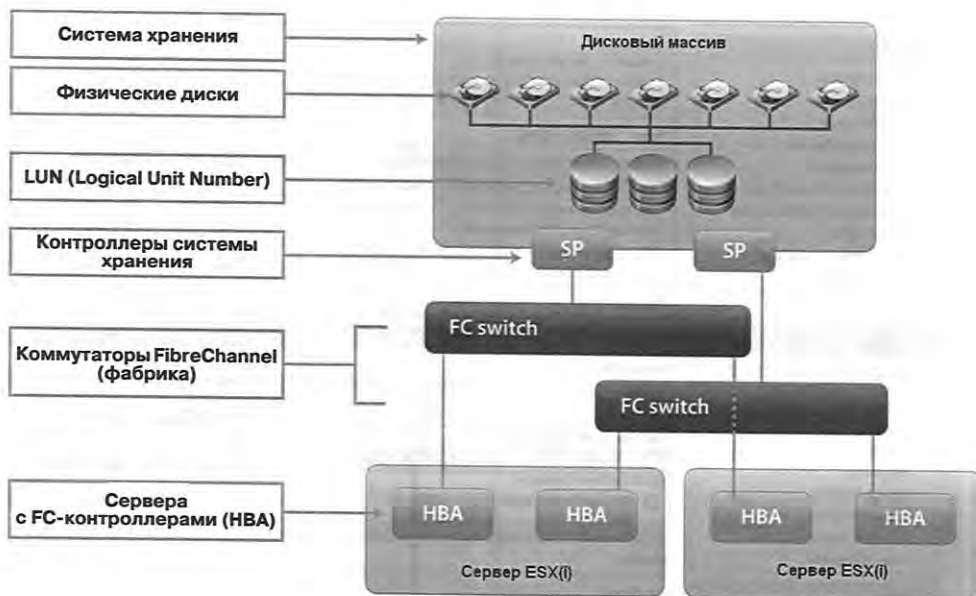


Рис. 3.6. Схема FC SAN
Источник: VMware

Здесь мы никак не будем касаться настроек со стороны SAN. После того как администраторы SAN презентуют нашим серверам ESXi какие-то новые LUN (или отключают старые), администраторам ESXi надо выполнить команду **Rescan** (на странице **Configuration** ⇒ **Storage Adapters** ⇒ справа наверху **rescan**), и вуаля – со стороны ESXi мы увидели дисковые ресурсы (рис. 3.7).

HBA, с точки зрения ESXi, представляет собой контроллер SCSI, и гипервизор с ним обращается как с обычным контроллером SCSI. Этот контроллер принимает на вход команды SCSI, их же отдает на выход. В SAN же HBA отдает команды SCSI, обернутые в пакеты Fibre Channel. Дисковые ресурсы, предоставляемые FC CXД, для серверов выглядят как обычные диски – см. рис. 3.7.

На рисунке вы видите выделенный FC HBA (наверху) и видимые через него диски (LUN), а также локальный контроллер SCSI (внизу) с видимыми ему дисками. В случае локального контроллера мы видим, скорее всего, созданный на нем RAID из локальных дисков сервера.

А теперь взгляните на рис. 3.8.

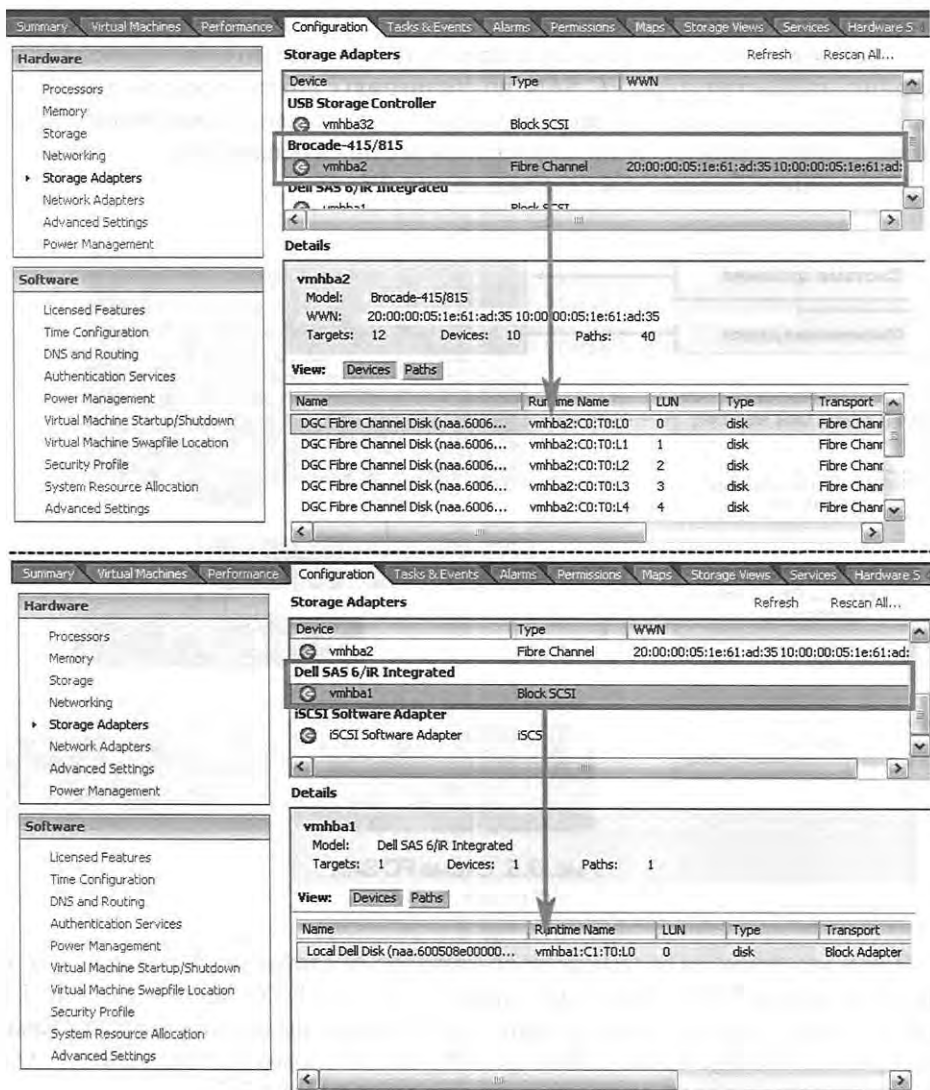


Рис. 3.7. Диски на FC и локальном SCSI-контроллере

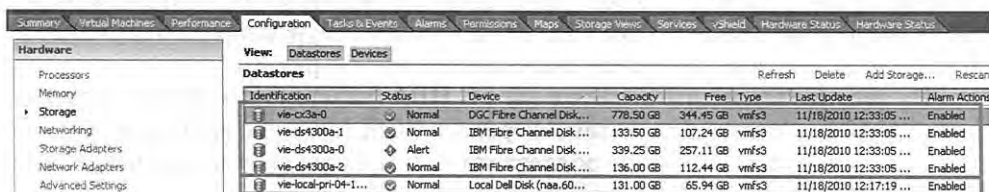


Рис. 3.8. Список хранилищ ESXi

Здесь вы видите список хранилищ (то есть разделов для хранения файлов ВМ), доступных так или иначе. И диски локальные здесь отображаются (да и используются) так же, как и FC LUN (впрочем, то же справедливо и для LUN iSCSI, и для NFS).

Что нам полезно будет знать про SAN как администраторам ESXi?

Топологию хотя бы с одним коммутатором FC называют SAN fabric, иногда прямо по-русски говорят «фабрика». Альтернатива этому – подключение СХД напрямую к HBA сервера. Как правило, конфигурации без коммутатора FC предполагают подключение двух серверов к двум контроллерам одного дискового массива – для экономии на стоимости коммутатора. Однако здесь возможна проблема следующего рода: некоторые FC СХД, особенно начального уровня, обладают возможностью работать только в режиме Active-Passive. Такой режим означает, что один LUN в какой-то момент времени может принадлежать лишь одному контроллеру системы хранения. И если разные сервера будут пытаться обратиться к одному LUN через разные контроллеры, СХД будет непрерывно переключать LUN между контроллерами, что может привести к их зависанию или, по крайней мере, к сильному падению производительности.

В общем-то, я это к чему – читать документацию системы хранения необходимо не только при использовании СХД высокого класса, но и в случаях попроще – могут быть специфические нюансы.

Что же мы можем настроить в случае Fibre Channel SAN на стороне ESXi? Именно настроить, по большому счету, совсем немного.

3.4.1. Адресация и multipathing

Вернитесь к схеме SAN на рис. 3.6. На нем вы видите, что в каждом сервере два HBA (или один двухпортовый), подключенных каждый к своему коммутатору FC, и в СХД тоже два контроллера. Это – рекомендованная конфигурация, когда у нас все компоненты SAN задублированы. Следствием дублированности является наличие нескольких путей от каждого сервера к каждому LUN. Например, первый сервер со схемы может добраться до LUN 1 четырьмя путями:

1. HBA 1 : SP 1 : LUN1.
2. HBA 1 : SP 2 : LUN1.
3. HBA 2 : SP 1 : LUN1.
4. HBA 2 : SP 2 : LUN1.

(Строго говоря, для того чтобы заработали пути 2 и 3, необходимо соединить между собой коммутаторы FC на рис. 3.6, чего на этом рисунке не показано.)

Модуль multipathing есть в ESXi по умолчанию, поэтому он определит, что видит не четыре разных LUN, а один с несколькими путями. См. пример на рис. 3.9.

Здесь вы видите два пути к LUN под названием FC_LUN_7. Это записи vmhba2:C0:T1:L0 и vmhba2:C0:T0:L0. Расшифровываются эти обозначения следующим образом:

- **vmhba#** – это имя контроллера в сервере. Физического контроллера (или порта на многопортовом HBA), который используется в сервере ESXi, а не

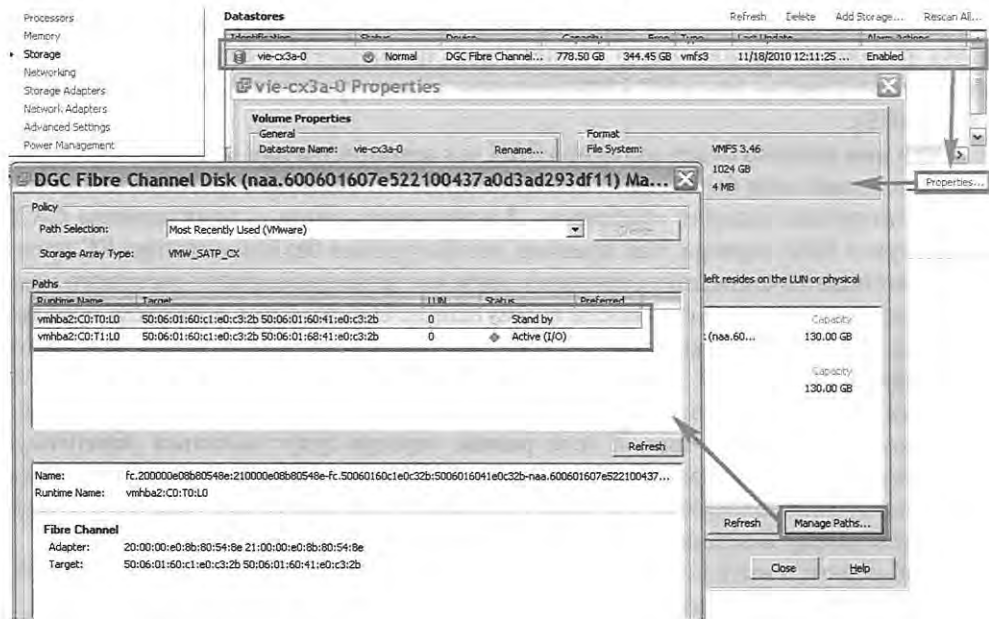


Рис. 3.9. Пример LUN с двумя путями к нему

виртуального контроллера SCSI, который создается для ВМ. Их список можно увидеть в настройках сервера **Configuration** ⇒ **Storage Adapter**;

- ❑ **C#** – номер канала SCSI. Обычно равен 0, но некоторые контроллеры каждую сессию SCSI выделяют в отдельный «канал». Более актуально для программного инициатора iSCSI – он отображает разными каналами разные интерфейсы VMkernel, через которые может подключиться к LUN;
- ❑ **T#** – номер «target», таргета, контроллера в CXD (Storage Processor). Разные ESXi, обращаясь к одним и тем же контроллерам, могут нумеровать их по-разному;
- ❑ **L#** – номер LUN. Эта настройка задается (или в простейших случаях выбирается автоматически) со стороны системы хранения.

Таким образом, в примере мы видим, что оба пути к FC_LUN_7 проходят через один контроллер сервера, но через разные контроллеры в системе хранения. Косвенный вывод – HBA в сервере является единой точкой отказа в данном случае.

Когда путей к LUN несколько, в отдельно взятый момент времени ESXi может работать только с каким-то одним (используемый в данный момент помечен строкой «(I/O)» в столбце **Status**, рис. 3.9). Переключение на другой путь произойдет лишь в случае отказа используемого. Для выбора того, какой путь использовать для доступа к LUN, у нас есть настройка политики (обратите внимание на выпадающее меню в верхней части рис. 3.9):

- ❑ **Fixed (VMware)** – если выбрана эта настройка, то сервер всегда будет использовать путь, выбранный предпочитаемым для доступа к LUN. Если

- путь выйдет из строя, произойдет переключение на другой, но когда предпочитаемый вернется в строй – опять начнет использоваться он;
- ❑ **Most Recently Used (VMware)** – если выбрана эта настройка, то сервер будет использовать текущий путь для доступа к LUN. Если путь выйдет из строя, произойдет переключение на другой, и он продолжит использоваться, даже когда предыдущий вернется в работоспособное состояние;
 - ❑ **Round Robin (VMware)** – в случае round robin большой поток данных делится на части и передается через разные пути поочередно. Теоретически это позволяет повысить производительность на участке между дисками и драйверами в ОС. Но это не спасет от задержек, если не хватает производительности дисков. Также чтобы использовать эту функцию, массив должен быть полностью active/active. По умолчанию другой путь начинает использоваться после передачи 1000 команд. Эта политика не используется по умолчанию по той причине, что она недопустима при реализации кластеров Майкрософт (MSCS/MFC) между виртуальными машинами.

Дополнительную информацию вы сможете найти в статье базы знаний – <http://kb.vmware.com/kb/1011340>.

Обратите внимание. В названии политик multipathing в скобках указано «VMware» по той причине, что это реализация данных политик от VMware. Но при установке на ESXi сторонних модулей multipathing возможно появление реализаций этих же политик от поставщика модуля multipathing.

В общем случае дать рекомендацию по выбору политики сложно, ищите рекомендации в документации вашей системы хранения.

Некоторые производители рекомендуют использовать для некоторых моделей своих систем хранения настройку round-robin. При этой настройке ESXi чередует все активные пути, переключаясь на следующий через определенное количество SCSI-команд. По умолчанию это количество команд равняется 1000, но некоторые производители рекомендуют уменьшить это значение до 1. Как это сделать, см. лучше в документации производителя СХД или в документации VMware (vSphere 5 Command Line Documentation ⇒ vSphere Command-Line Interface Documentation ⇒ vSphere Command-Line Interface Concepts and Examples ⇒ Managing Storage).

Не изменяйте настройку multipathing, если не понимаете, зачем. Внимательно изучайте документацию производителя системы хранения и ищите рекомендации по этим настройкам там.

3.4.2. Про модули multipathing. PSA, NMP, MMP, SATP, PSP

К одному LUN у нас могут вести несколько путей. Через разные HBA и разные контроллеры в СХД. За счет нескольких путей до LUN мы получаем дублирование и возможность продолжить работу при выходе из строя какого-то компонента

инфраструктуры SAN благодаря переходу к работающему пути. Еще наличие нескольких путей может помочь повысить производительность за счет распределения нагрузки между путями. Настройки multipathing определяют:

- ☐ когда переключаться – через какое количество непрочитанных/незаписанных блоков посчитать используемый путь сбойным;
- ☐ какой target использовать – любой или явно указанный;
- ☐ какой HBA использовать – любой, явно указанный или наименее загруженный.

После названия политик multipathing, описанных чуть выше, в скобках написано «VMware». Это потому, что это настройки встроенного модуля multipathing, разработанного VMware. В ESXi версии 5 есть возможность использовать сторонние модули.

При чтении документации на эту тему вам могут встретиться следующие термины:

- ☐ PSA – Pluggable Storage Architecture;
- ☐ NMP – Native Multipathing;
- ☐ SATP – Storage Array Type Plugins;
- ☐ PSP – Path Selection Plugins.

Как они соотносятся, см. на рис. 3.10.

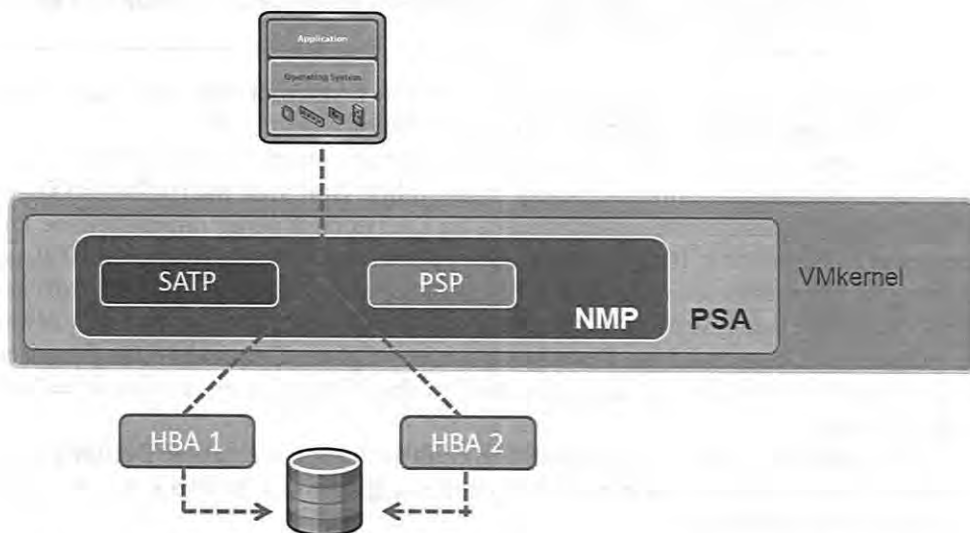


Рис. 3.10. Схема связей модулей гипервизора для работы с дисковой подсистемой
Источник: VMware

PSA – это общее название архитектуры, которая позволяет ESXi использовать сторонние модули для работы с SAN. Также PSA – это название набора API для VMkernel. Они были добавлены в состав гипервизора и выполняют такие задачи, как:

- ☐ загрузка и выгрузка модулей MPP;
- ☐ обнаружение и удаление путей к LUN;
- ☐ перенаправление запросов ввода-вывода к нужному MPP;
- ☐ разделение пропускной способности между ВМ;
- ☐ сбор статистики по вводу-выводу;
- ☐ координация действий Native Multipathing Module и сторонних плагинов, разработанных с помощью PSA API.

NMP – стандартный модуль работы с системой хранения, используемый по умолчанию и включающий multipathing. Ассосирует набор путей с LUN. NMP поддерживает все системы хранения из списка совместимости vSphere. NMP содержит в себе два компонента – SATP и PSP.

SATP – управляет переключением путей, отслеживает доступность путей и сообщает об изменениях в NMP, и все это – для конкретного типа СХД. То есть NMP работает со всеми типами поддерживаемых хранилищ за счет того, что для каждого типа у него в составе есть свой SATP, осведомленный о том, как работать с той или иной моделью СХД. Каждый SATP для конкретной модели может обладать возможностью выполнять специфичные для модели действия по обнаружению отказа пути и переходу на резервный путь. VMware поставляет SATP для всех поддерживаемых систем хранения – generic Storage Array Type Plugins для типовых систем хранения и local Storage Array Type Plugin для DAS.

Увидеть информацию о SATP и их настройках можно при помощи команды `esxcli` и пространства имен `storage`, выполните следующую команду для получения дополнительной информации:

```
esxcli storage nmp
```

PSP (Path selection plugin) – выбирает лучший путь. По сути, этот компонент обрабатывает настройку multipathing (ту, что Fixed/MRU/Round Robin). Сторонний PSP может уметь балансировать нагрузку по более сложным алгоритмам, нежели стандартный. В частности, задействовать несколько путей одновременно, а не последовательно.

Самое главное – архитектура PSA предполагает возможность использования сторонних модулей multipathing, которые могут работать вместо или вместе со стандартными. Их VMware называет MPP.

MPP – multipathing plugin. Сторонний модуль работы с системой хранения (сторонний модуль multipathing) является альтернативой NMP, то есть стандартному модулю работы с системами хранения. Разрабатывается MPP поставщиками СХД, которые могут усовершенствовать способы определения сбоев и перехода на другие пути за счет использования специфичных возможностей системы хранения. Также с помощью этих сторонних модулей возможна работа ESXi с массивами, изначально не поддерживаемыми.

Сторонние модули делятся на три категории (рис. 3.11):

- ☐ сторонние MPP обеспечивают поддержку специфичной модели СХД и делают работу с ней более производительной и надежной. Модуль или моду-

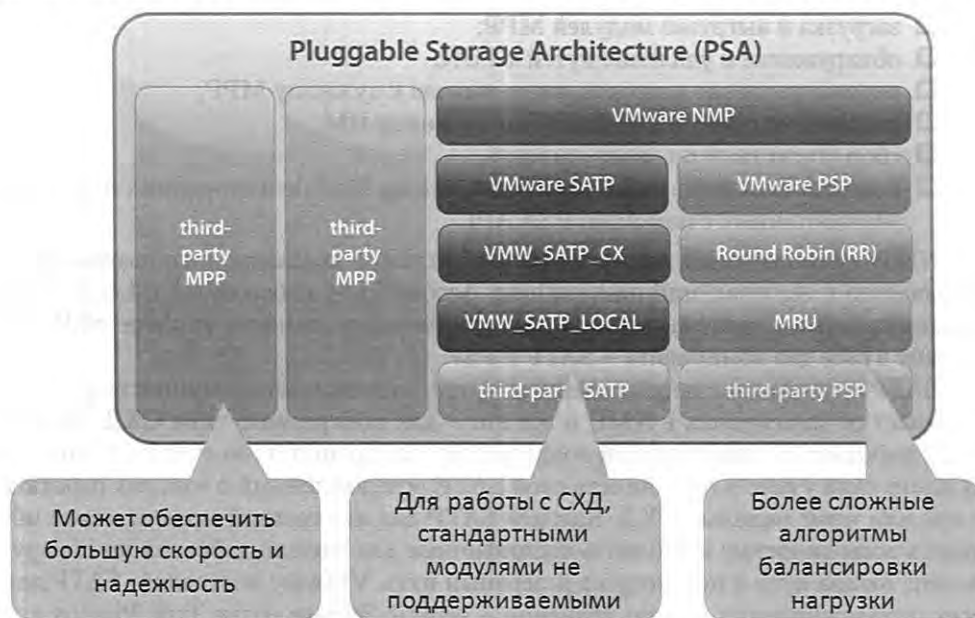


Рис. 3.11. Схема сосуществования
встроенного и сторонних модулей multipathing
Источник: VMware

ли MPP работают одновременно с NMP – стандартным модулем работы с системами хранения, если тот используется ESXi для других систем хранения (например, локальных дисков);

- ❑ сторонние SATP интегрируются в NMP, обеспечивают его работу с системами хранения, которых тот не поддерживает;
- ❑ сторонние PSP интегрируются в NMP. Содержат в себе описания более сложных алгоритмов балансировки I/O.

Когда сервер загружается, PSA обнаруживает все пути к LUN'ам. В соответствии с правилами, описанными в файле настройки (esx.conf), PSA определяет, какой из модулей multipathing должен управлять путями к тому или иному хранилищу. Этими модулями могут быть NMP от VMware или MPP от стороннего производителя.

NMP, опять же в зависимости от настроек, выбирает, какой из доступных SATP использовать для мониторинга путей, ассоциирует нужный модуль PSP для управления этими путями. Например, для семейства EMC CLARiiON CX по умолчанию используются SATP-модуль «VMW_SATP_CX» и PSP-модуль «Most Recently Used».

С помощью vSphere CLI или локальной командной строки и команды

```
esxcli storage nmp satp rule list
```

можно посмотреть на загруженные модули и указать настройки (claim rules) для PSA, NMP SATP, настроить маскировку LUN.

Командой

```
esxcli storage nmp satp list
```

можно увидеть список SATP для NMP.

А командой

```
esxcli storage nmp psp list
```

можно увидеть список PSP для NMP.

Эти настройки хранятся в файле /etc/vmware/esx.conf, и их можно просмотреть. Увидите строчку вида:

```
/storage/plugin/NMP/config[VMW_SATP_SYMM]/defaultpsp = "VMW_PSP_FIXED"
```

Эта строка файла настроек сообщает:

Для SATP с именем «VMW_SATP_SYMM» использовать PSP с именем «VMW_PSP_FIXED». Этот SATP применяется для работы с EMC Symmetrix, этот PSP предполагает политику путей «Fixed».

Из графического интерфейса мы можем увидеть следующее: **Configuration** ⇒ **Storage Adapters** ⇒ нужный HBA ⇒ кнопка **Paths** (рис. 3.12).

Здесь мы видим информацию о доступных путях и о том, какие из путей являются активными. Для ESXi четвертой версии под «Active» путем понимается

The screenshot shows the ESXi configuration console. On the left, the 'Hardware' tree has 'Storage Adapters' selected. The main pane shows 'Storage Adapters' with a list of devices: vmhba32 (USB Storage Controller), vmhba2 (QLA2340-Single Channel 2Gb Fibre Channel to PCI-X HBA), and vmhba1 (Dell SAS 6/iR Integrated). The 'Details' section for vmhba2 shows its WWN and target information. The 'Paths' tab is active, displaying a table of storage paths.

Runtime Name	Target	LUN	Status
vmhba2:C0:T1:L5	50:06:01:60:c1:e0:c3:2b 50:06:01:68:41:e0:c3:2b	5	Active (I/O)
vmhba2:C0:T4:L0	20:06:00:a0:b8:0f:4b:18 20:07:00:a0:b8:0f:4b:19	0	Active (I/O)
vmhba2:C0:T4:L1	20:06:00:a0:b8:0f:4b:18 20:07:00:a0:b8:0f:4b:19	1	Active (I/O)
vmhba2:C0:T4:L2	20:06:00:a0:b8:0f:4b:18 20:07:00:a0:b8:0f:4b:19	2	Active (I/O)
vmhba2:C0:T3:L0	50:06:01:60:c1:e0:c3:2b 50:06:01:61:41:e0:c3:2b	0	Stand by
vmhba2:C0:T3:L1	50:06:01:60:c1:e0:c3:2b 50:06:01:61:41:e0:c3:2b	1	Stand by
vmhba2:C0:T3:L2	50:06:01:60:c1:e0:c3:2b 50:06:01:61:41:e0:c3:2b	2	Stand by
vmhba2:C0:T3:L3	50:06:01:60:c1:e0:c3:2b 50:06:01:61:41:e0:c3:2b	3	Stand by

Рис. 3.12. Информация о доступных путях

тот, который можно задействовать для доступа к LUN. Задействованный в данный момент путь помечается как «Active (I/O)». «Standby» – такой путь можно задействовать в случае сбоя активного пути. «Broken» (на рисунке таких нет) – сбойный путь. Звездочкой помечается предпочитаемый (Preferred) путь в случае политики multipathing «Fixed».

Configuration ⇒ **Storage** ⇒ кнопка **Devices** ⇒ интересующий LUN (рис. 3.13).

Hardware

- Processors
- Memory
- Storage**
- Networking
- Storage Adapters
- Network Adapters
- Advanced Settings
- Power Management

Software

- Licensed Features
- Time Configuration
- DNS and Routing
- Authentication Services
- Power Management
- Virtual Machine Startup/Shutdown
- Virtual Machine Swapfile Location
- Security Profile
- System Resource Allocation
- Advanced Settings

View: Datastores **Devices**

Devices

Name	Backing Name	CDN	Type	Transport
IBM Fibre Channel Disk (naa.600a0b...	vmhba2:CD:T4:L1	1	disk	Fibre Channel
IBM Fibre Channel Disk (naa.600a0b...	vmhba2:CD:T4:L2	2	disk	Fibre Channel
DGC Fibre Channel Disk (naa.6006...	vmhba2:CD:T0:L5	5	disk	Fibre Channel
DGC Fibre Channel Disk (naa.6006...	vmhba2:CD:T0:L4	4	disk	Fibre Channel
Local USB CD-ROM (mpx.vmhba32...	vmhba32:CD:T0:L0	0	cdrom	Block Adapter
DGC Fibre Channel Disk (naa.6006...	vmhba2:CD:T0:L0	0	disk	Fibre Channel

Device Details

IBM Fibre Channel Disk (naa.600a0b...

Location: /vmfs/devices/disks/naa.600a0b8000... ID: naa.600a0b80000f4e1b000030344c
 Type: disk Capacity: 133.67 GB
 Owner: NMP

Primary Partitions	Capacity	Transport
1. VMFS	133.67 GB	Fibre Channel

Рис. 3.13. Информация о путях

Здесь можно увидеть, какой модуль управляет доступом к LUN. В данном примере это NMP, то есть стандартный модуль от VMware.

Ну и наконец, к чему все это.

Поставщик вашей системы хранения может предложить вам модуль multipathing, работающий лучше стандартного из состава ESXi. Если так, имеет смысл его установить. Подробности ищите в документации конкретного производителя. Например, EMC предлагает PowerPath for VMware vSphere – это и есть MPP.

Резюме: для ESXi 5 VMware предоставляет возможность и механизмы для разработки сторонних модулей работы с системами хранения, которые могут работать эффективнее стандартного.

Несколько слов про NMP, стандартный модуль multipathing. Достаточно важный вопрос: сколько времени требуется, чтобы перейти на другой путь в случае отказа используемого? NMP начинает задействовать другой путь сразу же, как только драйвер HBA вернет отказ I/O. В случае Fibre Channel это время меньше 30 секунд. Затем NMP нужно менее 30 секунд, чтобы переключиться на другой путь. Таким образом, время недоступности LUN для ESXi можно оценить как «меньше 60 секунд». Все запросы к дисковой подсистеме от VM ESXi поместит в очередь. Но именно из-за возможности такой паузы VMware рекомендует уста-



навливать время ожидания реакции ввода-вывода для гостевой ОС в 60 секунд (подробности см. в разделе 5.2.4 «Рекомендации для эталонных ВМ»).

3.4.3. Про зонирование (Zoning) и маскировку (LUN masking, LUN presentation)

В администрировании SAN есть два понятия: Zoning и LUN Masking. LUN Masking, маскировку, иногда называют LUN Presentation, «презентование». Они важны, поэтому их коснусь чуть подробнее.

Вот смотрите – у вас есть система хранения, и к ней подключены сервера. Скорее всего, это сервера с разными задачами. Какие-то используются под виртуальные машины, и на них установлен ESXi. Какие-то используются без виртуализации, и на них установлены Windows, Linux и другие операционные системы.

И на системе хранения мы создаем отдельные LUN'ы для этих серверов. В подавляющем большинстве случаев нам не надо, чтобы LUN, на котором ESXi хранит свои ВМ, был доступен с физического сервера с, к примеру, Windows, и наоборот. Более того, если он будет доступен – это очень опасно, так как может привести к повреждению данных. ESXi на таком LUN создаст свою файловую систему, VMFS, а Windows понятия не имеет, что это за файловая система. И будет считать этот диск пустым. И даст нам его отформатировать в NTFS. А это уничтожит VMFS и ВМ на ней. Вот для предотвращения подобных эксцессов и необходимы правильное зонирование и маскировка.

Зонирование – процесс настройки доступности СХД со стороны серверов. Выполняется на уровне коммутатора FC, в котором мы указываем видимость портов друг другом. То есть мы настраиваем, какие HBA к каким SP смогут обращаться. Некоторые коммутаторы FC требуют настройки зонирования всегда, вне зависимости от того, необходима ли эта настройка нам. Но как правильно настраивать зоны – надо смотреть документацию конкретного производителя.

Зонировать можно по портам или по WWN. WWN, World Wide Name – это уникальный идентификатор устройства в сети SAN, аналог MAC-адреса Ethernet. Например, если вы посмотрите на рис. 3.14, то увидите в нижней части рисунка WWN контроллера в сервере (HBA) и контроллера в системе хранения (SP).

VMware рекомендует зонировать по портам, а не по WWN. Связано это с тем, что одному порту ESXi могут соответствовать несколько WWN. Такое может произойти в случае, если мы назначаем уникальные собственные WWN каким-то виртуальным машинам. Позволяющий это механизм называется NPIV – N-Port ID Virtualization. Если наше оборудование FC поддерживает данный стандарт, то в свойствах ВМ мы можем активировать соответствующую настройку. Подробности см. в разделе, посвященном ВМ.

Маскировка – настройка, выполняемая на системе хранения. Суть настройки – в указании того, какому HBA (следовательно, серверу) какие LUN должны быть видны. Делается на уровне WWN, то есть в интерфейсе управления системой хранения мы должны выбрать LUN и выбрать WWN тех серверов (их HBA), которые должны его увидеть. В интерфейсах разных систем хранения эта опера-

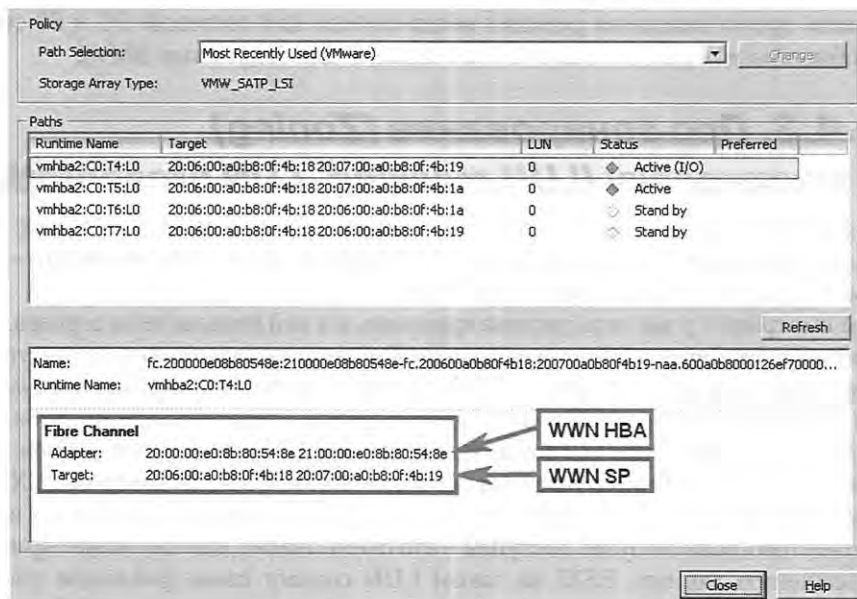


Рис. 3.14. Информация о путях к LUN

ция может называться по-разному; где-то она называется «презентованием LUN», LUN Presentation.

Настройки зонирования и маскировки являются взаимодополняющими. С точки зрения администратора ESXi, необходимо, чтобы корректно было выполнено зонирование – наши сервера ESXi имели доступ к СХД. И на этих СХД была сделана маскировка – то есть сервера ESXi имели доступ к выделенным под VM LUN, не имели доступа ни к каким другим LUN и чтобы к их LUN не имели доступа никакие другие сервера.

Обратите внимание. Маскировка может выполняться на самом сервере ESXi. Это необходимо в том случае, когда мы хотим скрыть какой-то LUN от ESXi, но сделать это на SAN мы по каким-то причинам не можем. Для того чтобы замаскировать LUN со стороны ESXi (фактически спрятать самому от себя), нам понадобится командная строка и раздел **Mask Paths** в документе **vSphere Storage ⇒ Understanding Multipathing and Failover**.

3.5. SAN, iSCSI

Storage Area Network, SAN – сеть хранения данных. Инфраструктура SAN предполагает создание выделенной сети только под данные. В случае iSCSI SAN такая сеть строится поверх обычной инфраструктуры IP.

Используются обычные коммутаторы Ethernet, а в серверах в качестве контроллеров используются более или менее специализированные аппаратные ини-

циаторы iSCSI(iSCSI HBA) или обыкновенные сетевые контроллеры (за счет использования программного инициатора – службы, входящей в состав ESXi любых редакций). За счет использования стандартной, относительно дешевой и часто уже имеющейся инфраструктуры IP iSCSI SAN может оказаться дешевле FC SAN.

Меньшая цена, простота внедрения, практически все те же функции, что и у FC SAN, – это плюсы iSCSI. В минусы записывают меньшую максимальную скорость и, возможно, большие задержки (при использовании дешевого 1 Гб Ethernet).

Для iSCSI актуально практически все, что выше написано для Fibre Channel SAN. Теперь напишу про моменты, специфичные именно для iSCSI.

В контексте iSCSI часто употребляется термин «инициатор», initiator. В широком смысле инициатор iSCSI – это тот, кто инициирует обращение к ресурсам (здесь – дисковым), то есть сервер. В нашем случае ESXi-сервер.

Инициатор iSCSI в более узком смысле – это контроллер, который обеспечивает подключение к iSCSI CXД.

Инициатор iSCSI – это контроллер SCSI, который принимает команды SCSI от VMkernel, упаковывает их в пакеты IP и отправляет по Ethernet на хранилище. Получает в ответ пакеты IP, извлекает из них команды SCSI и отдает их гипервизору.

iSCSI-инициатор бывает аппаратный, с аппаратной поддержкой и программный (рис. 3.15).

В случае аппаратного инициатора ESXi видит его как обычный дисковый контроллер, HBA. Гипервизор отдает SCSI-команды его драйверу, тот передает их

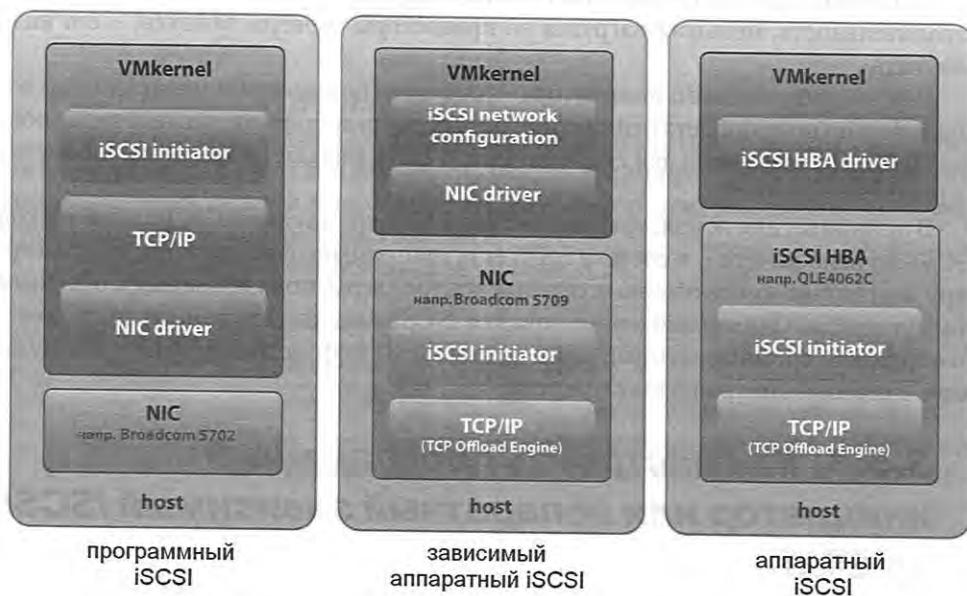


Рис. 3.15. Иллюстрация разницы между вариантами инициатора iSCSI
Источник: VMware

контроллеру, и контроллер на своем чипе инкапсулирует их в IP, которые сам отдает в сеть.

В случае программной реализации гипервизор передает команды SCSI на вход службе программного инициатора iSCSI, та запаковывает SCSI в IP, а IP отдает стеку TCP/IP VMkernel. А VMkernel через обычные сетевые контроллеры отдает эти пакеты в сеть. Однако во множестве случаев производительности и обеспечиваемой скорости у программного iSCSI оказывается вполне достаточно.

В случае промежуточной реализации у нас есть контроллеры, которые одновременно и являются сетевыми, и предоставляют функционал iSCSI инициатора. Отличие таких сетевых контроллеров от аппаратных инициаторов iSCSI – в том, что каждый такой контроллер отображается одновременно и как сетевой, и как дисковый контроллер в интерфейсе ESXi. Зависимые инициаторы требуют настройки сети – такие же, как и программный инициатор, то есть создание виртуальных сетевых интерфейсов VMkernel, назначение их на соответствующие физические сетевые контроллеры, настройка Discovery.

С точки зрения ОС (здесь – ESXi), аппаратный iSCSI HBA – это контроллер SCSI, не отличающийся от FC HBA. То, что FC HBA оборачивает SCSI в пакеты fibre channel, а iSCSI HBA – в пакеты IP, для ОС прозрачно, и она их воспринимает одинаково. То есть если у вас есть аппаратный инициатор, то нужно настроить только его – указать ему собственный IP-адрес и IP-адреса ЦХД (iSCSI target).

Но, в отличие от FC HBA, инициатор iSCSI может быть программным. Самое для нас главное – в ESXi такой программный инициатор есть.

Плюсы аппаратного и зависимого инициаторов – потенциально большая производительность, меньшая нагрузка на процессоры сервера. Минусы – его надо покупать.

Плюсы программного инициатора – для его использования можно ничего дополнительно не приобретать или приобретать лишь дополнительные сетевые контроллеры. Но максимальная скорость может быть меньше, и возрастает нагрузка на процессоры сервера.

В принципе, для использования iSCSI с ESXi, кроме самой ЦХД с поддержкой iSCSI, не надо ничего – в составе ESXi есть программный инициатор, то есть серверу достаточно иметь обычные сетевые контроллеры, подключенные к обычным коммутаторам. Однако для использования iSCSI в производственной среде обычно оправдано организовать выделенную под iSCSI инфраструктуру IP – коммутаторы и сетевые контроллеры серверов.

3.5.1. Как настроить программный инициатор или аппаратный зависимый iSCSI на ESXi

Краткий план настройки программного инициатора iSCSI таков.

1. Настраиваем vКоммутатор, включаем Jumbo Frames (при необходимости включаем использование Jumbo Frames на физическом сетевом оборудо-

- вании). Включение Jumbo Frames является не обязательным, но рекомендуемым.
2. Назначаем на вКоммутатор физические сетевые контроллеры, крайне желательно хотя бы два (из обычных соображений отказоустойчивости). Из соображений производительности их может быть и больше. Если речь идет о настройке аппаратного зависимого инициатора iSCSI, то привязывать к вКоммутатору необходимо именно те сетевые контроллеры, которые являются еще и контроллерами iSCSI.
 3. Добавляем порты VMkernel по числу физических сетевых контроллеров на этом виртуальном коммутаторе. Если Jumbo Frames будут использоваться, то увеличить MTU следует и для интерфейсов VMkernel. Контроллеров больше одного необходимо для того, чтобы задействовать механизмы multipathing для программного инициатора iSCSI.
 4. В настройках **NIC Teaming** для групп портов привязываем каждый интерфейс VMkernel из п. 3 к какому-то одному физическому сетевому контроллеру (то есть все, кроме одного, переносим в группу Unused. Один не перенесенный – разный для каждого vmk#).
 5. Включаем VMware iSCSI Software Initiator (только для чисто программного инициатора).
 6. Привязываем порты VMkernel к iSCSI Software Initiator (вкладка **Network Configuration** в свойствах программного iSCSI-инициатора).
 7. Настраиваем подключение ESXi к системе хранения iSCSI, для этого настраиваем **Discovery** и, при необходимости, аутентификацию.
 8. Создаем хранилище VMFS.

Теперь подробнее.

Настройка сети для iSCSI

Первое, что необходимо сделать, – это настроить сеть для работы iSCSI. Вернитесь на рис. 3.15 – служба инициатора iSCSI формирует пакеты IP, а в сеть они попадают через сетевой стек VMkernel. Это значит, что нам понадобится виртуальный сетевой контроллер VMkernel.

Примерная конфигурация показана на рис. 3.16.

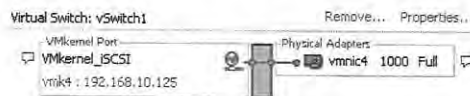


Рис. 3.16. Необходимая настройка виртуальной сети для программного инициатора iSCSI

«Примерная» потому, что на этом же вКоммутаторе могут располагаться и любые другие группы портов. Потому что физических сетевых интерфейсов лучше бы использовать хотя бы два, чтобы не было единой точки отказа. Потому что интерфейсов VMkernel нужно несколько, если вас интересует балансировка нагрузки.

Небольшой нюанс. Взгляните на рис. 3.17.

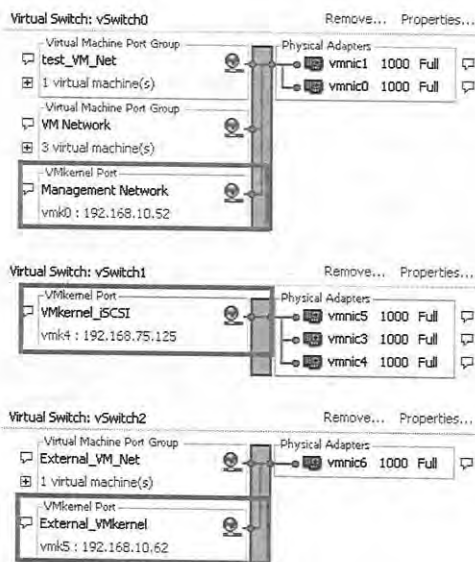


Рис. 3.17. Пример настройки сети на ESXi. Выделены все порты VMkernel

Здесь вы видите пример настройки сети на ESXi, и в этой сети есть три интерфейса VMkernel. Если я попытаюсь подключить к этому серверу ESXi хранилище iSCSI с IP-адресом 192.168.75.1, то через какой из этих трех интерфейсов гипервизор попытается обратиться на этот адрес? Ответ на данный вопрос прост. Гипервизор представляет собой операционную систему. Эта операционная система использует три сетевые карты (в данном примере это интерфейсы vmk0, vmk4, vmk5). Какую из них выбрать, она решает в соответствии с таблицей маршрутизации. Очевидно, что в моем примере она выберет vmk4, находящийся в той же подсети. Поэтому я и назвал его «VMkernel_iSCSI», так как предполагал, что через него пойдет iSCSI-трафик. И привязал к его вКоммутатору сразу три физических сетевых контроллера – так как для трафика IP-СХД рекомендуется несколько выделенных сетевых контроллеров.

Для настройки маршрутизации для гипервизора существует специальная команда `esxcfg-route`.

Необходимо, чтобы доступ с сервера на iSCSI СХД не требовал маршрутизатора.

Включение iSCSI-инициатора и настройка Discovery

Следующий шаг – необходимо включить службу программного инициатора iSCSI. Для этого перейдите **Configuration** ⇒ **Storage Adapters** ⇒ ссылка **add** в правой верхней части окна. В появившемся окне подтвердите, что вы хотите

активировать iSCSI-инициатор. Появится еще один дисковый контроллер (vmhba#), который находится в группе iSCSI Software Adapter.

В свойствах этого инициатора есть возможность узнать или поменять идентификатор. В контекстном меню инициатора (vmhba#) выберите **Properties** ⇒ кнопка **Configure** (рис. 3.18).

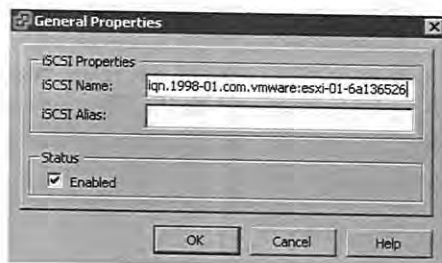


Рис. 3.18. Настройки программного инициатора iSCSI

Обратите внимание на строку «iSCSI Name». Это iSCSI qualified names (IQN), уникальный идентификатор устройства iSCSI, аналог WWN для Fibre Channel. Формируется он автоматически. Хотя мы можем его изменить, обычно этого нам не нужно. Если вдруг изменяем – тогда в наших интересах озаботиться его уникальностью в нашей инфраструктуре. Однако подробности об именовании iSCSI ищите в разделе 3.9 «Адресация SCSI».

Таким образом, в случае iSCSI у каждого устройства есть несколько идентификаторов – IP-адрес (часто несколько) и iSCSI Name вида iqn.* или другого. В случае ESXi IP-адрес мы задаем в свойствах интерфейса VMkernel, через который собираемся выпускать трафик iSCSI.

Следующий шаг – это настройка Discovery.

Сессии discovery – это часть протокола iSCSI, возвращающая список target с системы хранения iSCSI. Discovery бывают динамическими (иногда называемыми Send Targets) и статическими.

В случае динамического мы указываем адрес системы хранения, и после discovery нам возвращается список LUN, доступных на ней. Найденные LUN записываются на вкладке **Static Discovery**.

В случае статического discovery ESXi пытается обратиться к конкретному LUN, не опрашивая систему хранения о других доступных.

Для настройки того или иного метода discovery надо зайти в свойства программного iSCSI HBA (**Configuration** ⇒ **Storage Adapters**) и перейти на вкладку **Dynamic Discovery** (обычно удобнее) или **Static Discovery**. Нажимаем кнопку **Add**, указываем IP-адрес системы хранения iSCSI, используемый сетевой порт и, в случае статического discovery, имя таргета.

Обратите внимание. LUN, найденные через Dynamic Discovery, ESXi автоматически помещает в Static Discovery для ускорения обращения к ним при следующих включе-

ниях. Не забудьте удалить из списка **Static Discovery** записи о LUN, доступ к которым для ESXi был отменен.

Далее надо настроить аутентификацию, но про нее чуть позже.

После нажатия **ОК** появится окно с сообщением о том, что после изменения настроек НБА требуется пересканировать систему хранения, согласимся с этим. Если все было сделано правильно, выделив НБА, в нижней части окна мы увидим доступные LUN.

Теперь про аутентификацию. На стороне системы хранения нам необходимо презентовать LUN серверу. Обычно в случае iSCSI это делается по его идентификатору (IQN или другому). Кроме того, ESXi 5 поддерживает аутентификацию по протоколу CHAP, в том числе взаимную.

Чтобы указать имя и пароль на стороне ESXi, при задании таргета надо нажать кнопку **CHAP** – см. рис. 3.19.

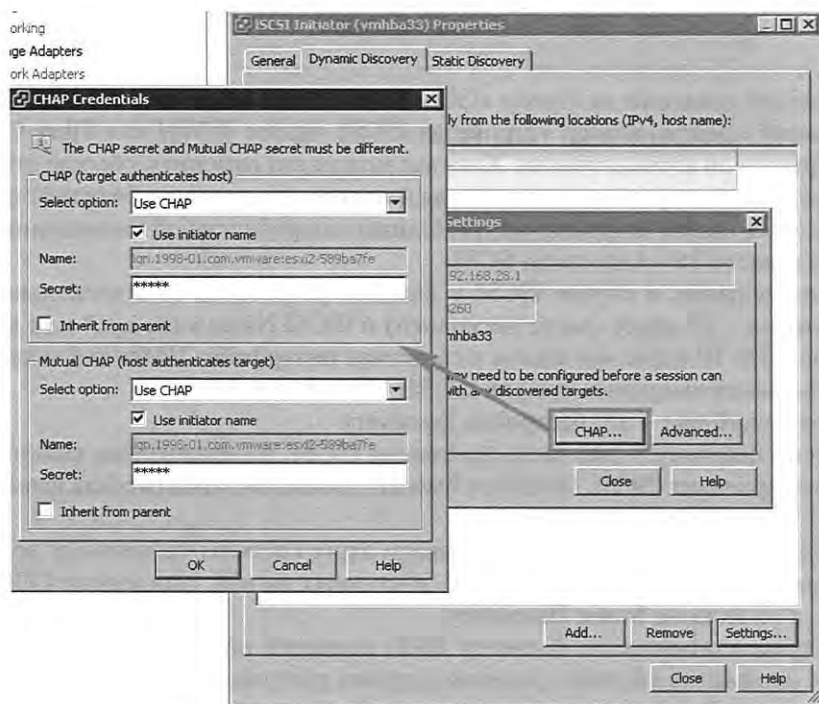


Рис. 3.19. Настройки аутентификации CHAP

Здесь настраиваем аутентификацию CHAP в соответствии с нашими требованиями.

Если там же нажать кнопку **Advanced**, то попадем в окно указания расширенных настроек iSCSI (рис. 3.20).

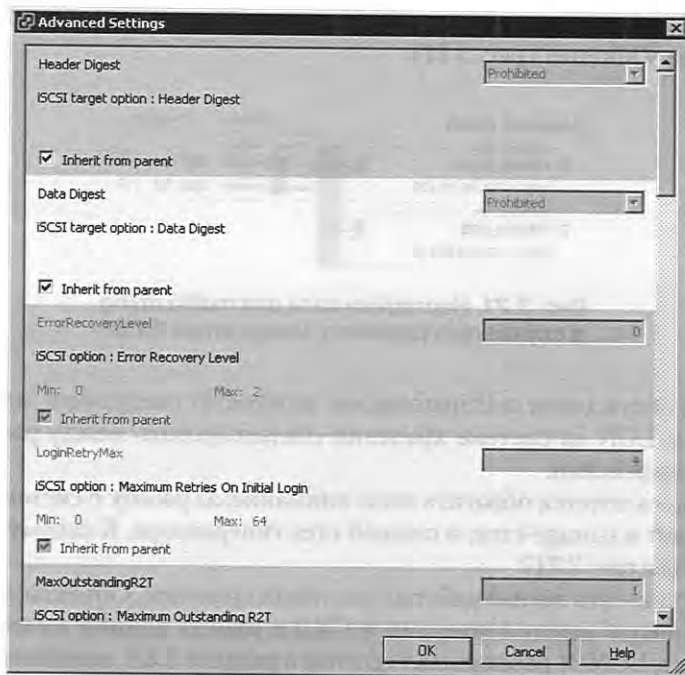


Рис. 3.20. Расширенные настройки для iSCSI

Здесь следует что-то изменять, лишь если вы хорошо понимаете, что делаете. Обычно поменять какие-то из этих настроек советует поддержка VMware в случае возникновения каких-либо проблем. Для справки см. <http://link.vm4.ru/adviscsi>.

3.5.2. iSCSI Multipathing

А теперь пару слов про iSCSI multipathing для программного инициатора.

Во-первых, у одной системы хранения iSCSI может быть (и обычно бывает) несколько контроллеров, каждый со своим IP-адресом (иногда не одним). Надо добавить их все, и ESXi сам разберется, что эти несколько таргетов показывают на самом деле на одни и те же LUN.

Во-вторых, ESXi может использовать multipathing между своими контроллерами. А раз речь идет про Ethernet (поверх которого передается iSCSI), эти контроллеры будут сетевыми.

Но есть один нюанс. Программный инициатор iSCSI – это служба, сервис в операционной системе ESXi. Для доступа в сеть она использует виртуальные сетевые интерфейсы VMkernel. Эти интерфейсы подключены к vКоммутаторам, а к тем подключены несколько физических сетевых контроллеров.

Рассмотрим два варианта:

1. На vКоммутаторе несколько каналов во внешнюю сеть, но только один интерфейс VMkernel.

2. На коммутаторе несколько каналов во внешнюю сеть и несколько интерфейсов VMkernel (рис. 3.21).

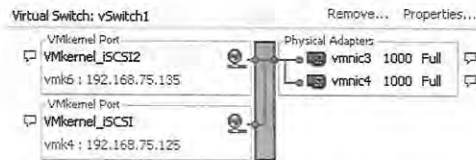


Рис. 3.21. Настройки сети для multipathing в случае программного инициатора iSCSI

В рамках обсуждения multipathing нас интересует следующая задача: обращения к разным LUN на системе хранения сбалансировать между разными физическими интерфейсами.

На что здесь хочется обратить ваше внимание: за работу с системой хранения iSCSI отвечают и storage-стек, и сетевой стек гипервизора. К какому из них относятся объекты с рис. 3.21?

Storage-стек – это логика работы с системой хранения. Объекты, участвующие в обмене данными между серверами и СХД в рамках данной логики, отображаются в путях к LUN. Я рассказывал об этом в разделе 3.4.1, но напомним основное здесь.

Зайдя в свойства любого LUN, мы увидим один или несколько путей к нему, вида `vmhba37:C0:T0:L0`.

- `vmhba##` – это дисковый контроллер. Он может быть локальным RAID-контроллером, контроллером Fibre Channel, Fibre Channel over Ethernet или, в нашем случае, программным контроллером iSCSI.
- `C#` – это канал SCSI. Важно – в нашем случае программного инициатора iSCSI разными каналами являются *разные интерфейсы VMkernel* (`vmk#`), через которые служба программного инициатора получает доступ к СХД.
- `T#` – это (обычно) контроллер на стороне СХД.
- `L#` – это номер LUN.

Выводы, важные в рамках данного раздела: к разным LUN мы можем обращаться через разные каналы (здесь – интерфейсы `vmk#`) и через разные таргеты (правда, это зависит от возможностей и настройки СХД). А вот выбрать, к примеру, два разных физических сетевых контроллера (`vmnic#`) для трафика к двум разным LUN в рамках данной логики у нас возможности нет.

Для сетевого стека ситуация во многом обратна – трафик, сгенерированный на каком-то интерфейсе `vmk#`, может быть передан на физический сетевой интерфейс (`vmnic#`), и затем физический коммутатор передаст его на порт СХД. Здесь мы можем балансировать трафик между физическими интерфейсами (`vmnic#`), но нет возможности определить, какой трафик к какому LUN относится.

В связи с этими объяснениями еще пара слов про multipathing на примере конфигурации с рис. 3.21.

Если конфигурация не такая, как на этом рисунке, – интерфейс VMkernel создан только один, то multipathing может быть обеспечен лишь на уровне виртуального коммутатора. Но в силу логики работы алгоритмов балансировки нагрузки далеко не всегда трафик iSCSI будет балансироваться оптимальным образом. Даже если vКоммутатор настроен на балансировку нагрузки по хешу IP, то через разные физические сетевые контроллеры будет пересылаться трафик к разным IP-адресам системы хранения. Более тонкого разделения проводиться не будет – а для системы хранения нам было бы удобно иметь возможность управлять сессиями к каждому LUN.

А вот в случае, как на этом рисунке, трафик iSCSI наверняка сможет задействовать несколько каналов во внешнюю сеть за счет того, что балансировать нагрузку сможет storage-стек VMkernel между интерфейсами VMkernel, а затем мы можем связать интерфейсы VMkernel с физическими сетевыми контроллерами. Таким образом, мы обеспечим желаемую балансировку нагрузки за счет совместной работы на уровне и storage-стека, и сетевого стека.

Для организации подобной конфигурации multipathing нам необходимо на vКоммутатор назначить несколько физических интерфейсов и создать на нем же столько же портов VMkernel (напомню, что на рис. 3.21 приведен пример описанной мной конфигурации). Затем следует сопоставить их один к одному, то есть для каждого порта VMkernel указать свой vmnics как единственный активный.

Для осуществления этого зайдите в свойства виртуального коммутатора, выберите группу портов с первым из интерфейсов VMkernel и нажмите **Edit**. Вкладка **NIC Teaming** ⇒ флажок **Override vSwitch failover order** ⇒ все vmnics, кроме одного выбранного, перенесите в группу **Unused Adapters** (рис. 3.22).

Повторите этот шаг для каждого интерфейса VMkernel, выбирая каждый раз следующий vmnics.

Однако если мы этим ограничимся, то будет актуальна следующая проблема: для доступа в сеть служба программного инициатора iSCSI выбирает интерфейс VMkernel по таблице маршрутизации. Выбирается тот интерфейс, который в одной IP-подсети с системой хранения. Если таких интерфейсов несколько – выбирается и используется один из них (!). А нам необходимо, чтобы использовались все, созданные на предыдущем шаге.

Для этого зайдём в настройки инициатора iSCSI – в его контекстном меню **Properties** ⇒ вкладка **Network Configuration** ⇒ **Add**. В открывшемся окне мы увидим список интерфейсов VMkernel этого сервера ESXi. Нам следует выбрать первый из подготовленных ранее интерфейсов и нажать **OK**. Затем повторить добавление для каждого из подготовленных интерфейсов (на примере рис. 3.21 мне надо добавить vmk4 и vmk6).

Смысл этой настройки: инициатор iSCSI будет использовать для доступа к СХД *все* интерфейсы VMkernel, выбранные на вкладке **Network Configuration**.

После завершения настройки подключения к системе хранения iSCSI вы увидите несколько путей к каждому LUN. Те пути, что отличаются каналом (C#), идут через разные vmk# (и, как следствие, через разные vmnics).

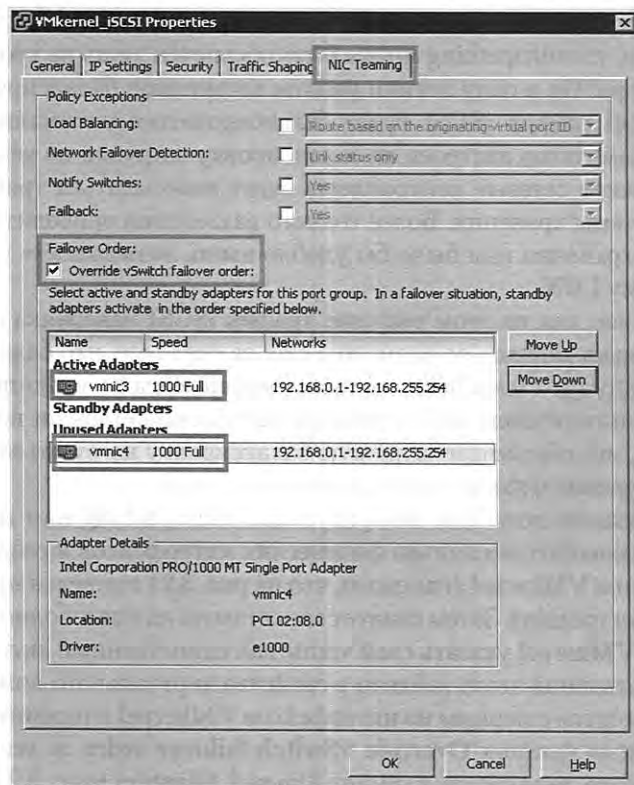


Рис. 3.22. Настройка NIC Teaming для портов VMkernel, используемых инициатором iSCSI

Обратите внимание. Эта настройка не отработает, если есть используемые в данный момент пути через эти интерфейсы VMkernel. Таким образом, лучше всего эти настройки делать перед тем, как настраивать Discovery и подключаться к iSCSI СХД. Если у вас инициатор iSCSI уже настроен и через него уже подключена система хранения, правильным будет полностью ее отключить – мигрировав с этого сервера все ВМ, использующие это хранилище, удалив все записи из вкладок **Static Discovery** и **Dynamic Discovery** и выполнив после этого rescan.

Следующий нюанс – система хранения iSCSI может по-разному адресовать LUN. Вспомним физический адрес LUN, путь к нему вида `vmhba#:C#:T#:L#`. Какие-то системы хранения могут презентовать LUN показывать как разные LUN одного таргета или как разные LUN разных таргетов. То есть мы увидим пути вида:

- ☐ `vmhba33:C0:T0:L0;`
- ☐ `vmhba33:C0:T0:L1;`
- ☐ `vmhba33:C0:T0:L2.`

Или:

- ☐ `vmhba33:C0:T0:L0;`

- ☐ vmhba33:C0:T1:L0;
- ☐ vmhba33:C0:T2:L0.

Программный iSCSI-инициатор ESXi устанавливает одно соединение на таргет. Таким образом, в первом случае, когда сервер работает с одним таргетом и тремя LUN на нем, весь трафик iSCSI пойдет через один внешний интерфейс. Во втором случае, когда три LUN подключены через три таргета, – через несколько. С точки зрения производительности, второй вариант может быть интереснее. Я упоминаю об этом на случай, если у вас есть выбор.

Последний шаг – задать настройку политики multipathing. С модулем multipathing по умолчанию вам доступны политики Fixed, Most Recently Used и Round-Robin (их описание доступно в разделе 3.4.1). Обязательно читайте документацию вашей системы хранения и следуйте ее рекомендациям.

3.6. VMFS, Virtual Machine File System

Дисковые ресурсы, к которым осуществляется блочный доступ, – это локальные диски (или другие DAS) и LUN на системах хранения FC/FCoE и iSCSI – ESXi может отформатировать в файловую систему VMFS. Это проприетарная, закрытая файловая система от VMware, обладающая полезным под нужды виртуализации функционалом:

- ☐ кластерность. VMFS является кластерной файловой системой. Это означает, что с отформатированным в эту файловую систему разделом могут работать одновременно несколько серверов. VMFS обладает системой блокировок, которая обеспечивает целостность данных;
- ☐ поддержка файлов больших размеров. В разделе VMFS могут быть расположены файлы размером до 2 Тб (это ограничение актуально на момент написания);
- ☐ поддержка LUN большого размера. В VMFS можно отформатировать диск/LUN размером до 64 Тб ($64 \times 1024 \times 1024 \times 1024 \times 1024$ байт);
- ☐ VMFS является журналируемой файловой системой;
- ☐ минимальные накладные расходы – при размещении файла виртуального диска в разделе VMFS какого-то LUN мы получаем производительность, практически идентичную тому, как если бы этот же LUN отдавали ВМ как RDM, напрямую.

Создать раздел VMFS достаточно просто. Если у нас есть видимые ESXi-диски (LUN) и на каком-то из них мы хотим создать VMFS, то для этого идем **Configuration** ⇒ **Storage** ⇒ **Add Storage**.

Запустится мастер. По шагам:

Мы хотим отформатировать в VMFS какой-то диск/LUN, а не подмонтировать NFS. Оставляем вариант «Disk/LUN» (рис. 3.23).

Затем выбираем, на каком из LUN мы хотим создать VMFS. Нам будут предложены только те, на которых раздела VMFS еще не создано, потому что на одном LUN мы можем создать только один раздел VMFS (рис. 3.24).

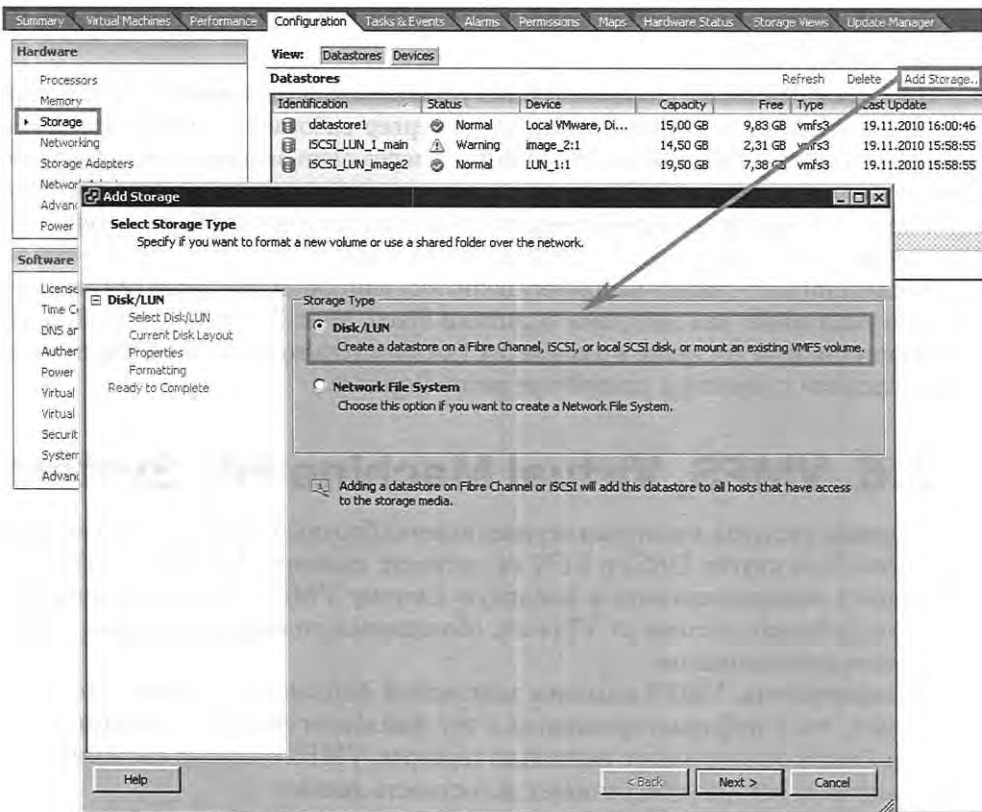


Рис. 3.23. Добавление хранилища VMFS

Обратите внимание: на этом этапе мы можем попасть в крупные неприятности, потому что в пустых незадействованных LUN в этом списке могут находиться LUN'ы с данными, но подключенные как RDM (а в редких случаях, при ошибках презентирования LUN на СХД, мы вообще увидим LUN, не предназначенный для ESXi). Если мы выберем такой LUN и создадим на нем VMFS, то данные гостевой ОС будут уничтожены. Поэтому очень важно наверняка знать адрес LUN, который мы создали под VMFS. Идентификатором может служить номер LUN в первую очередь. Еще – путь к LUN, так как из него можно понять, через какой контроллер ESXi этот LUN видит. Наконец, косвенным идентификатором может служить размер. В общем, будьте внимательны. К счастью, для каждого LUN мы можем указать произвольный, легкий в восприятии идентификатор – см. раздел 3.9 «Адресация SCSI».

На следующем шаге нам покажут информацию о выбранном диске. Это очень важно! На шаге **Current Disk Layout** мы должны увидеть надпись **The hard disk is blank**. Если это так – это означает, что диск пуст, и, отформатировав его, мы не затронем ничьих данных. Иначе, если мы увидели информацию о таблице разделов, нам надо быть твердо уверенными в том, что мы пытаемся отформатировать

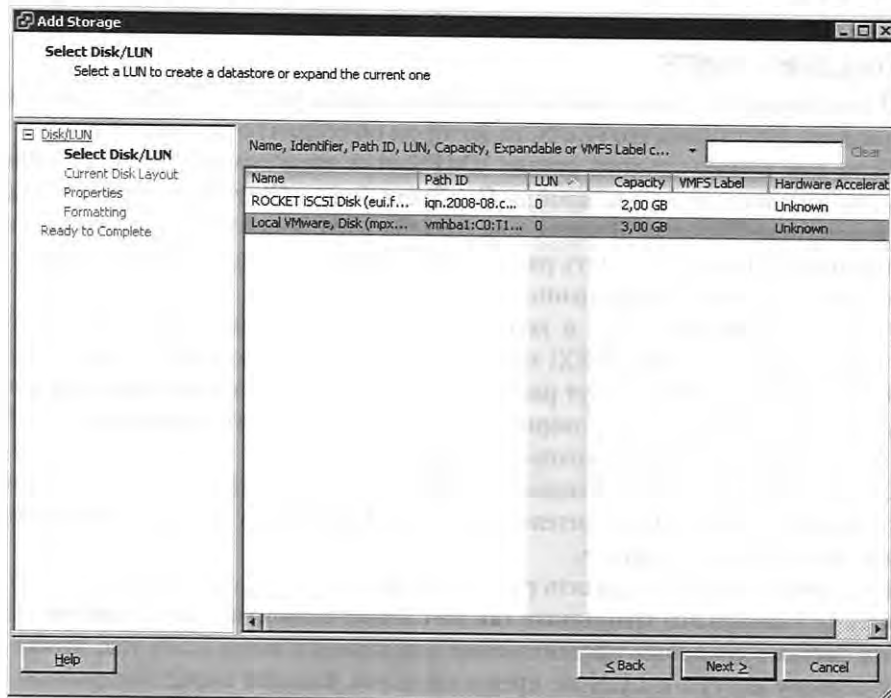


Рис. 3.24. Выбор LUN для создания раздела VMFS

правильный диск/LUN и информация с существующих на нем разделов никому не нужна.

Затем предложат ввести **Datastore Name**, имя хранилища. По сути, это метка тома, которую мы потом будем видеть в интерфейсе vSphere Client. В ваших интересах сделать ее максимально информативной, это опять же сведет к минимуму вероятность ошибок в будущем. Имеет смысл указать тип СХД, задачу, если LUN выделен под конкретные цели. Например, «fc_lun_db_production». В этом имени также имеет смысл избегать пробелов и спецсимволов. Впрочем, появившийся в vSphere 5 механизм «Profile Driven Storage» может нам помочь решить вопрос с описанием возможностей каждого хранилища проще, чем шифровать всю информацию в его названии.

Шаг **Formatting** – на этом шаге мастера вы можете указать размер создаваемого раздела, занимает ли он LUN целиком или только его часть. Думаю, вы всегда будете оставлять вариант по умолчанию – и VMFS будет занимать все место на LUN. Причин поступать по-другому я не вижу.

Обратите внимание. Для VMFS версии 5 размер блока равен 1 Мб. Но это не значит, что даже пустой файл будет занимать минимум мегабайт – для хранения маленьких файлов ESXi умеет использовать специальные области заголовков. Таким образом, волноваться из-за потерь места на небольшие файлы конфигурации и т. п. не стоит, потерь нет.

Корректное отключение LUN или удаление раздела VMFS

В некоторых ситуациях нам необходимо удалить раздел VMFS или, без удаления раздела, отключить его (LUN, на котором он создан) от сервера ESXi.

Если вам нужно удалить раздел VMFS, то в этом поможет ссылка **Remove**, которую вы сможете найти, пройдя **Configuration** ⇒ **Storage** и выделив удаляемое хранилище. Нажатие **Remove** для раздела VMFS именно удалит раздел и все данные на нем. Однако удалять раздел следует лишь тогда, когда вы уверены, что на нем нет представляющих ценность данных.

Для пятой версии ESXi в контекстном меню хранилища появился пункт **Unmount**. При его выборе ESXi проверяет, можно ли отключить данный раздел VMFS. Помешать в этом могут работающие на нем виртуальные машины и некоторые настройки. Если эта проверка не пройдена, то удалять данный раздел VMFS или бесполезно, или опасно потерей данных.

Таким образом, при необходимости удалить раздел VMFS первым шагом вы отмонтируете его пунктом контекстного меню **Unmount** и только после успешного отмонтирования – удалите.

Если необходимо не удалять раздел, а сделать его недоступным конкретному серверу, то сначала это хранилище так же следует отмонтировать, а затем настроить на системе хранения презентацию/маскировку этого LUN так, чтобы данному серверу доступ к LUN не предоставлялся. Однако перед настройкой маскирования LUN на СХД обязательно следует выполнить операцию **Detach**. Эта процедура удаляет упоминание о LUN, и гипервизор больше не пытается его обнаружить. Для выполнения данной процедуры пройдите **Configuration** ⇒ **Storage** ⇒ кнопка **Devices** ⇒ в контекстном меню отключаемого LUN пункт **Detach**.

Дополнительная информация доступна в базе знаний – <http://kb.vmware.com/kb/2004605>.

Если хранилище следует отмонтировать сразу для всех серверов, то удобнее это сделать, пройдя **Home** ⇒ **Datastores** ⇒ контекстное меню отмонтируемого хранилища ⇒ **Unmount**. На первом шаге мастера отметьте флажками сервера ESXi, от которых это хранилище следует отмонтировать.

Способ массово осуществить **Detach** из графического интерфейса мне неизвестен.

Технические особенности VMFS

Если посмотреть свойства только что созданного хранилища VMFS, то мы увидим, что несколько сотен мегабайт на нем сразу заняты. Заняты они под так называемые метаданные, «metadata», описание раздела.

Сами файлы метаданных расположены в корне раздела:

- ☐ .fdc.sf – file descriptor system file;
- ☐ .sbc.sf – sub-block system file;
- ☐ .fbb.sf – file block system file;
- ☐ .pbc.sf – pointer block system file;
- ☐ .vh.sf – volume header system file.

В этих метаданных хранится информация о самом разделе:

- ☐ размер блока. Он может отличаться от 1 Мб, если этот раздел был отформатирован в VMFS предыдущей версии (где размеров блока было несколько, на выбор администратора), а затем VMFS обновлен до версии 5;
- ☐ Number of Extents – количество расширений раздела VMFS (extent), то есть на какие еще LUN распространен этот том VMFS;
- ☐ Volume Capacity – размер раздела;
- ☐ VMFS Version – версия VMFS;
- ☐ Volume Label – метка раздела VMFS;
- ☐ VMFS UUID – уникальный идентификатор раздела VMFS.

Просмотреть эту информацию можно командой

```
vmkfstools -P -h /vmfs/volumes/<метка тома VMFS>
```

Информация ниже вряд ли будет актуальна для ESXi версии 5 в связи с развитием так называемого vStorage APIs for Array Integration (VAAI, раздел 3.10). Однако твердой уверенности в этом у меня нет, и для справки я не стал удалять ее при обновлении материала книги.

Еще в метаданных хранится информация о файлах на разделе. О том, что они существуют, о том, какие блоки под них выделены, о том, с какого сервера они открыты.

Важный момент здесь в следующем.

VMFS предлагает совместный доступ с нескольких серверов одновременно за счет блокировки на уровне файлов. Когда сервер запускает какую-то ВМ, он блокирует лишь принадлежащие ей файлы. Информация об их блокировке записывается в метаданные.

Вот у вас есть две ВМ, их файлы расположены на одном хранилище VMFS, и они работают на разных серверах. В этом случае производительность данного LUN просто делится между этими ВМ, с минимальными накладными расходами. Эти ВМ одновременно читают и пишут данные на одном и том же VMFS, каждая в свои файлы.

Но если необходимо внести изменения в метаданные, то для внесения изменений LUN отдается в монопольное пользование серверу, который вносит изменения. Делается это с помощью команды **Reservation** (Блокировка) протокола SCSI.

SCSI Reservation происходит при:

- ☐ включении и выключении ВМ – потому что в метаданных надо прописать, что файлы этой ВМ теперь открыты или закрыты;
- ☐ создании файлов. Это такие операции, как создание виртуальной машины или шаблона, клонирование ВМ, Storage VMotion и миграция выключенной виртуальной машины на другое хранилище, добавление диска ВМ, создание снимков состояния;
- ☐ удалении файла;
- ☐ смене владельца файла;
- ☐ установке метки времени последнего обращения/изменения;

- ☐ увеличении раздела VMFS;
- ☐ изменении размера файлов – а это происходит регулярно при работе ВМ со снимками состояния и если ее диски (файлы vmdk) находятся в формате thin.

Файлы изменений (*-delta.vmdk) снимков состояния растут блоками по 16 Мб. Тонкие диски увеличиваются по одному блоку VMFS. Мы записали внутрь ВМ еще сколько-то мегабайт, файл vmdk стало необходимо увеличить, для указания нового размера файла в метаданных раздела произошел SCSI reservation. Затем vmdk-файл еще увеличился, SCSI reservation повторился. И так далее.

И что может получиться: на одном VMFS расположены десятки ВМ, все работают на разных серверах, у всех снимки состояния. И каждый раз, когда у какой-то ВМ надо увеличить размер файла-дельты, происходит SCSI reservation, и какое-то маленькое время (обычно ~10 миллисекунд) девять других серверов с этим LUN не работают, потому что десятый вносит изменения в метаданные. Разово эта операция вообще нестрашна и нормальна. По данным VMware, и при средней нагрузке негативный эффект на производительность от этих блокировок нулевой. Но в граничных случаях, когда на одном хранилище много ВМ, у всех диски растущие и эти ВМ работают на многих разных серверах, мы можем недополучить часть производительности LUN.

Вывод: лучше стараться избегать постоянной работы при существующих снимках состояния (snapshot) для производственных ВМ, а если для каких-то из них это необходимо – стараться держать их на отдельных LUN от тех производственных ВМ, которым особенно важна производительность дисковой подсистемы.

Отследить потери производительности из-за регулярных блокировок мы можем, проанализировав соответствующий журнал. Это файл /var/log/vmkernel, где мы можем отследить произошедший «reservation conflict»:

```
Apr 24 15:59:53 esx35-1 vmkernel: 5:14:57:01.939 cpu0:1083)StorageMonitor: 196:
vmhba1:0:3:0 status = 24/0 0x0 0x0 0x0
Apr 24 15:59:53 esx35-1 vmkernel: 5:14:57:01.939 cpu0:1041)SCSI: vm 1041: 109: Sync CR
at 64
Apr 24 15:59:56 esx35-1 vmkernel: 5:14:57:04.982 cpu0:1151)StorageMonitor: 196:
vmhba1:0:3:0 status = 24/0 0x0 0x0 0x0
Apr 24 15:59:56 esx35-1 vmkernel: 5:14:57:04.982 cpu3:1041)SCSI: vm 1041: 109: Sync CR
at 16
Apr 24 15:59:56 mel-esx-02 vmkernel: 5:14:57:05.050 cpu0:1161)StorageMonitor: 196:
vmhba1:0:3:0 status = 24/0 0x0 0x0 0x0
Apr 24 15:59:57 esx35-1 vmkernel: 5:14:57:06.047 cpu3:1041)SCSI: vm 1041: 109: Sync CR
at 0
Apr 24 15:59:57 esx35-1 vmkernel: 5:14:57:06.047 cpu3:1041)WARNING: SCSI: 119: Failing
I/O due to too many reservation conflicts
```

Если вы столкнулись с ситуацией, когда блокировки оказывают негативное влияние на работу дисковой подсистемы, можно попробовать следующую конфигурацию.

Сделать расширение раздела (extent) из нескольких LUN, где первый LUN будет небольшим, порядка 2 Гб. Тогда на нем не будет файлов-дисков ВМ (не

влезут), а останутся только метаданные. И блокировки SCSI не будут оказывать влияния на скорость работы виртуальных машин на прочих LUN.

Также на VMFS есть так называемый «Heartbeat Region». В этой области сервера периодически обновляются записи с целью сообщить о своей работоспособности. Если сервер вышел из строя или потерял связь с хранилищем и своевременно не обновил свою запись, блокировки файлов этим сервером считаются недействительными.

3.6.1. Увеличение размера хранилища VMFS. Grow и Extent

Мы создаем хранилище VMFS на диске/LUN какого-то объема. Если обстоятельства сложились неудачно, мы можем оказаться в ситуации, когда места на этом VMFS хватать перестанет. Для преодоления данной проблемы есть два пути – использовать операцию Grow или Extent.

Grow – это когда у нас на том же LUN появилось еще свободное место, тогда мы увеличиваем размер раздела VMFS за счет него. Если наша система хранения не позволяет нам увеличивать размер LUN, тогда Grow нам не пригодится.

Extent – это когда мы берем раздел VMFS и «натягиваем» его на еще один LUN, где VMFS нет.

VMFS Grow

Если мы (или администратор системы хранения по нашему запросу) добавили место на LUN, чтобы его задействовать под VMFS, то идите **Configuration** ⇒ **Storage** ⇒ раздел VMFS с этого LUN ⇒ **Properties** ⇒ кнопка **Increase**. Нам покажут список LUN (рис. 3.25), и в этом списке мы должны увидеть тот из них,

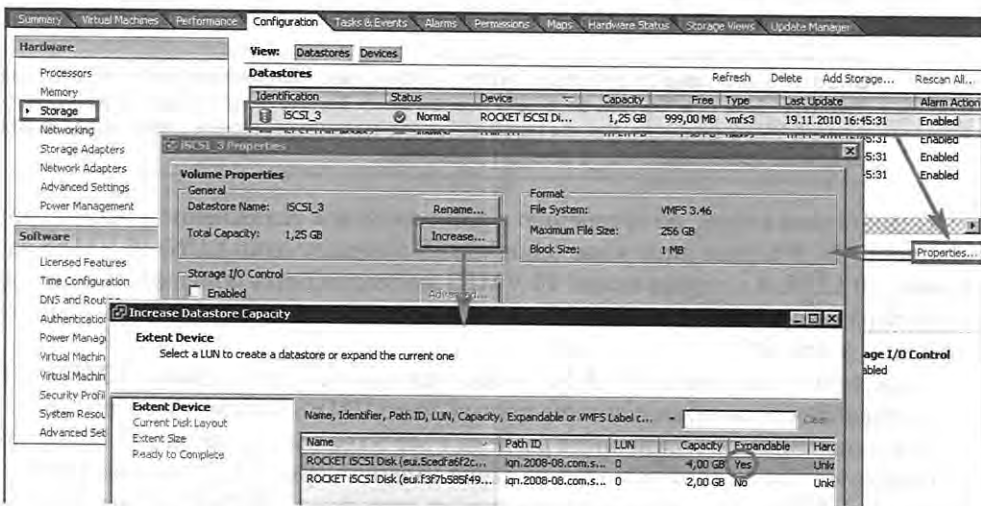


Рис. 3.25. Увеличение VMFS за счет свободного места того же LUN

VMFS на котором хотим увеличить. В столбце **Expandable** для него должно быть «Yes» – это значит, на нем есть не занятое под VMFS место.

Выбираем его, **Next, Next** – и все.

Увеличить размер LUN на системе хранения, а затем увеличить VMFS – это предпочтительный вариант решения проблемы с недостатком свободного места на хранилище VMFS.

VMFS Extent

Один раздел VMFS может существовать сразу на нескольких LUN. Вам может потребоваться такая конфигурация, если необходимо увеличить размер раздела VMFS, а увеличение LUN (и последующий grow раздела) невозможен. Например, если вы не имеете доступа к управлению SAN, а ваши администраторы систем хранения не могут или не намерены увеличивать ранее созданные LUN.

Обратите внимание. Для VMFS предыдущих версий было актуально ограничение на размер LUN в 2 Тб. Для VMFS версии 5 максимальный размер LUN составляет 64 Тб.

Для добавления в существующий том VMFS еще одного LUN следует пройти **Configuration** ⇒ **Storage** ⇒ раздел VMFS, который хотим увеличить ⇒ **Properties** ⇒ кнопка **Increase**. Нам будет показан список LUN, которые можно задействовать для увеличения выбранного раздела VMFS. В этом списке будут те LUN, на которых нет VMFS.

Обратите внимание: кроме пустых, среди них могут быть LUN, задействованные как RDM (хотя обычно все-таки система их прячет, но могут быть накладки), а то и вообще LUN посторонних серверов при неправильном зонировании или маскировке. То есть мы можем расширить том VMFS на LUN с данными, что приведет к их уничтожению. Будьте внимательны при выборе.

После завершения мастера по выбору LUN для расширения размер раздела VMFS будет увеличен (рис. 3.26).

Обратите внимание: операция extent необратима. Если вы расширили VMFS на какой-то LUN, освободить этот LUN невозможно. Только если удалить весь расширенный VMFS целиком. Как вы понимаете, это потребует перемещения файлов ВМ на другие хранилища, что не всегда приемлемо.

Недостатком extent, по сравнению с grow, является усложнение администрирования SAN. В случае grow у нас один VMFS занимает один LUN. 10 VMFS занимают 10 LUN. А в случае extent 10 VMFS могут занимать большее количество LUN. Банально, количество LUN больше – и повышается вероятность ошибки администратора SAN.

Даже если у нас всего 10 LUN, но все они принадлежат одному VMFS, все равно вероятность ошибки и потери данных всего VMFS хоть немного, но выше.

Все метаданные объединенного раздела VMFS хранятся на первом LUN. Если по ошибке или из-за сбоя выходит из строя именно первый LUN одного распределенного VMFS, то мы теряем все данные на всем томе VMFS. Если выходит из строя любой LUN, кроме первого, мы теряем данные только с него.

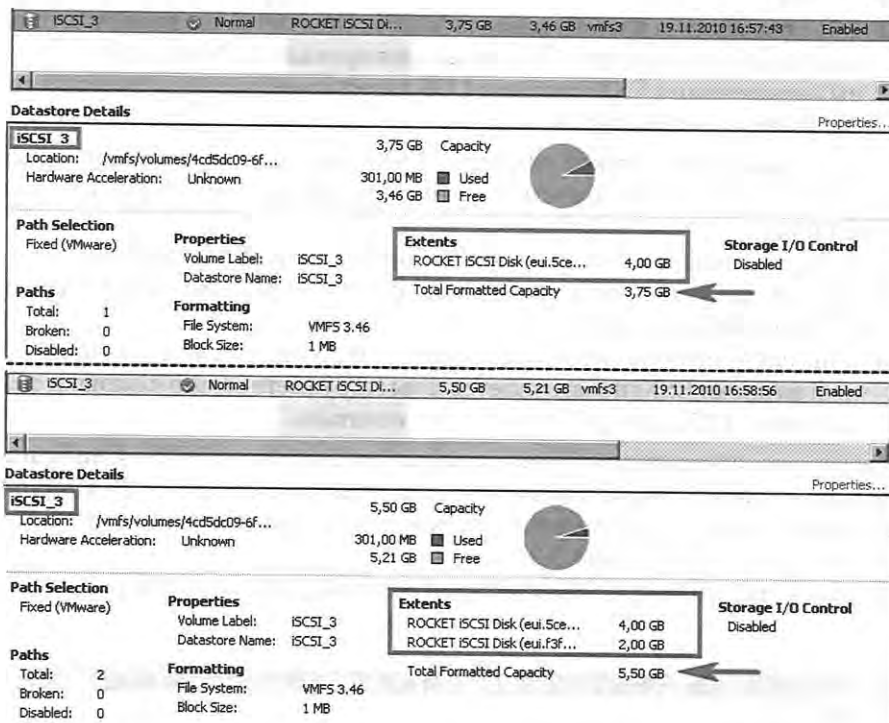


Рис. 3.26. До и после выполнения extent

Впрочем, вероятность отказа именно LUN (а не отдельного диска или RAID-группы) мне видится крайне низкой – исключая человеческий фактор.

Еще одним потенциальным минусом такой конфигурации является тот факт, что для VMFS после extent не работает функция Storage IO Control, SIOC.

Обратите внимание. ESXi 5 поддерживает увеличение VMFS, RDM и vmdk. Уменьшить VMFS невозможно. Уменьшить vmdk можно при помощи VMware Converter. ESXi 5 поддерживает уменьшение RDM, так как оно обрабатывается лишь гостевой ОС.

3.6.2. Доступ к клонированному разделу VMFS, или к разделу VMFS с изменившимся номером LUN

В метаданных VMFS хранятся уникальный идентификатор (UUID, universally unique identifier) раздела VMFS и номер LUN, на котором он был создан. Если в силу каких-то причин изменился номер LUN (или имя iqn.*, для iSCSI), то ESXi перестанет работать с этим VMFS – в списке **Configuration** ⇒ **Storage** вы его не увидите. Ситуация, при которой номер LUN отличается от номера, записанного на разделе VMFS, может быть штатной, когда:

- ☐ ESXi обращается к клону раздела VMFS. Обычно такое происходит, когда настроена репликация LUN или вы клонировали LUN вручную;
- ☐ вы подключили к ESXi снапшот LUN (здесь имеется в виду функция системы хранения «snapshot»).

То есть, оказавшись в такой ситуации, ESXi предполагает, что видит реплику LUN. А раз это реплика, то записывать что-то на нее означает разрушить целостность реплики.

Если же вы оказались в такой ситуации незапланированно, например:

- ☐ изменились какие-то настройки на стороне системы хранения и у какого-то LUN поменялся номер;
 - ☐ или (например) у вас произошел какой-то программный сбой, установленный на диске ESXi не загружается и вы загружаете сервер с флэшки с ESXi.
- Для этого ESXi номер LUN может меняться.

Так вот, в такой ситуации обратитесь к мастеру создания VMFS – **Configuration** ⇒ **Storage** ⇒ **Add Storage**. Вы должны увидеть проблемный LUN и в столбце **VMFS Label** – метку существующего на нем (но не отображаемого в штатном интерфейсе) раздела VMFS (рис. 3.27).

Затем вы увидите вопрос «Как поступить с этим разделом VMFS?» (рис. 3.28).

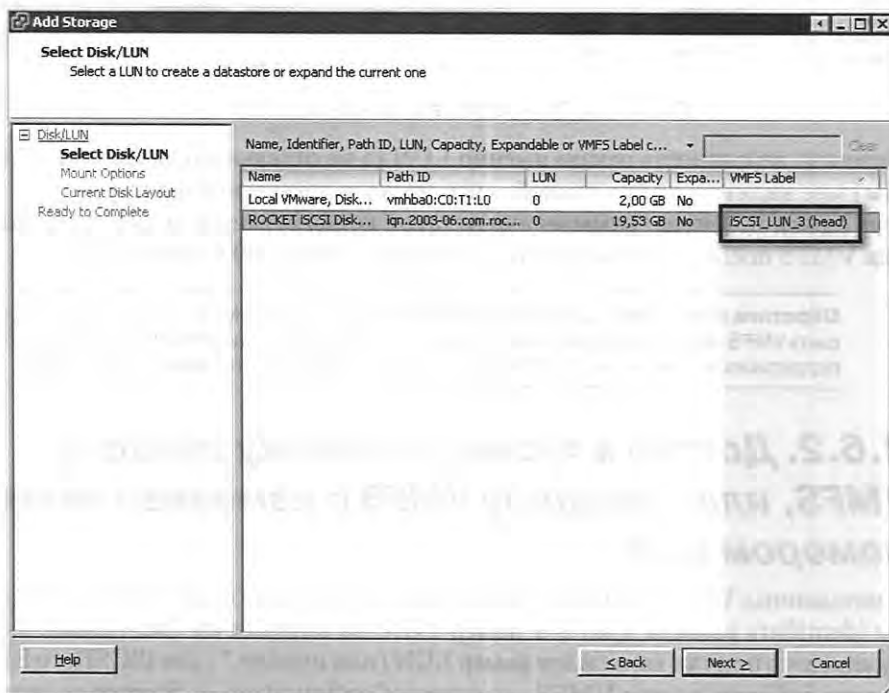


Рис. 3.27. Восстановление доступа к разделу VMFS на LUN с изменившимся номером

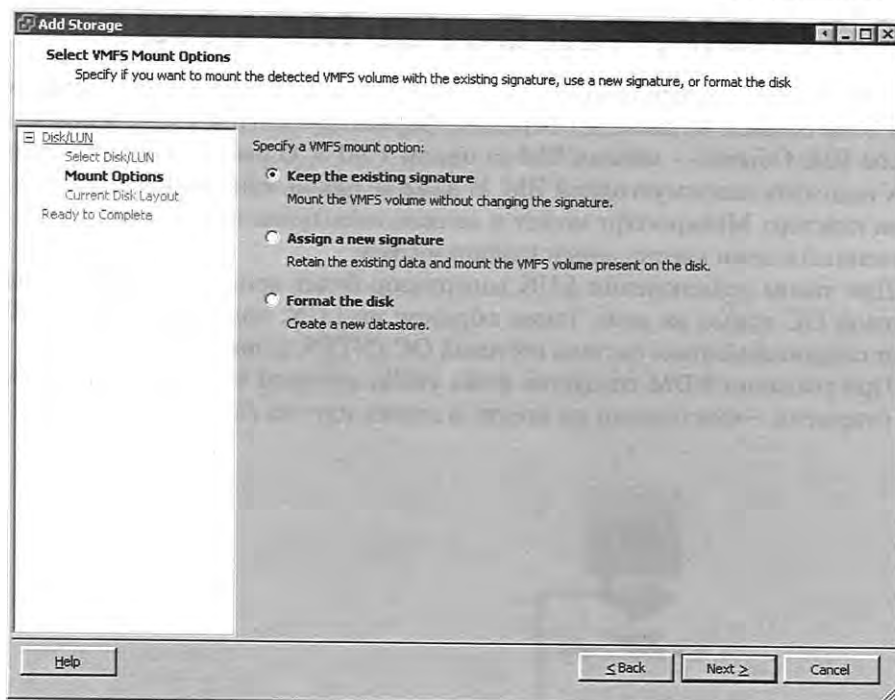


Рис. 3.28. Восстановление доступа к разделу VMFS на LUN с изменившимся номером

Варианты следующие:

- ☐ **Keep the existing signature** – подключить VMFS как есть, без изменений. Используется в случае, когда вы хотите подключить реплику LUN и получить к ней доступ. Например, в случае сбоя основной площадки запустить виртуальные машины на резервной площадке, из реплики. Изменений метаданных не происходит. Однако не получится подключить к ESXi и исходный том, и его реплику одновременно, потому что UUID у них совпадает. В таком случае выбор данного варианта будет недоступен. Однажды подключенные таким образом разделы VMFS в дальнейшем будут подключаться к серверам и после их перезагрузок;
- ☐ **Assign a new signature** – сгенерировать и записать новый UUID для этого VMFS. Все VM и шаблоны с этого хранилища придется заново добавить в иерархию vCenter (ESXi);
- ☐ **Format the disk** – заново создать VMFS на этом LUN. Уничтожит все имеющиеся данные.

Обратите внимание. Из командной строки эти операции возможны с помощью команды `esxcfg-volumes`. Вам пригодятся ключи `-l` для просмотра информации о разделах и `-m` или `-M` для их подмонтирования. При использовании `-M` раздел VMFS останется подмонтированным и после перезагрузки.

3.7. RDM, Raw Device Mapping

Raw Device Mapping (RDM) – это альтернатива VMFS. В случае хранилища VMFS мы создаем на диске/LUN раздел, форматируем его в VMFS и храним там файлы VM. Обычно – многих VM на одном VMFS. Однако мы можем какой-то LUN выделить напрямую одной VM. И даже не одной, например для диска с данными кластера Майкрософт может и должен использоваться как раз RDM, подключенный к двум виртуальным машинам сразу.

При таком подключении LUN гипервизор будет пропускать SCSI команды гостевой ОС прямо на него. Таким образом, на LUN, подключенном как RDM, будет создана файловая система гостевой ОС (NTFS, к примеру).

При создании RDM создается файл vmdk, который выполняет роль ссылки для открытия, – фактически же чтение и запись идут на сам LUN (рис. 3.29).

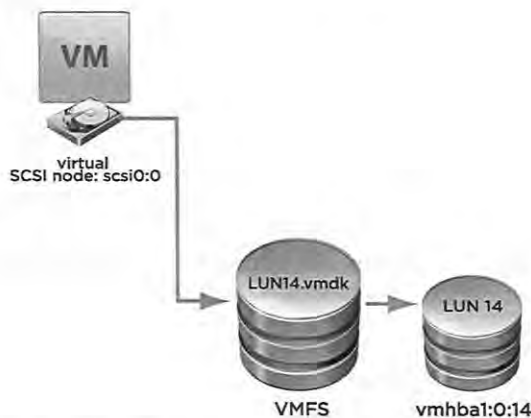


Рис. 3.29. Иллюстрация подключения RDM
Источник: VMware

Размер этого файла vmdk отображается равным объему RDM LUN (объему LUN 14 в моем примере), однако на самом деле он занимает пренебрежимо мало места.

RDM вам интересен в случае, если:

- ☐ в пятой версии vSphere RDM может быть полезен тогда, когда вам необходимо дать для виртуальной машины один диск размером более 2 Тб. На момент написания это невозможно при использовании vmdk и возможно для RDM;
- ☐ происходит миграция физической инфраструктуры на виртуальную. Переносимый физический сервер использует LUN для хранения своих данных. Мы можем не копировать данные с этого LUN внутрь файла vmdk на каком-то VMFS, а прямо этот LUN подключить к перенесенному в VM серверу как RDM;

- ❑ вы хотите поднять кластер Майкрософт с переходом по отказу (MSCS/MFC), хотя бы одним из узлов которого будет ВМ. В таком случае кворумным диском и диском с общими данными должен выступать RDM;
- ❑ вы хотите использовать механизм снапшотов или какие-то другие функции на уровне системы хранения для данных виртуальных машин. Например, мы можем средствами системы хранения создать снапшот LUN и этот снапшот подключить к серверу резервного копирования. В случае VMFS + vmdk такая схема, скорее всего, не заработает, потому что сервер резервного копирования не сможет забрать данные с проприетарной файловой системы VMFS. А если этот LUN подключен как RDM к виртуальной машине, то файловую систему на нем создает гостевая ОС, и эта файловая система может быть знакома серверу резервного копирования;
- ❑ из политических соображений – хранение каких-то данных в проприетарном формате (VMFS + vmdk) противоречит корпоративным политикам или предписаниям регулирующих органов.

Однако использование RDM не дает заметных изменений в скорости работы с дисковой подсистемой. По данным VMware, разница в скорости доступа к одному и тому же LUN как к RDM или к файлу vmdk на нем различается на проценты, и иногда VMFS + vmdk даже быстрее.

Чтобы добавить RDM к ВМ:

1. Зайдите в свойства виртуальной машины, на вкладке **Hardware** нажмите **Add**. Вам нужно добавить **Hard Disk**.
2. **Select a disk** – выберите **Raw Device Mapping**.
3. **Select Target LUN** – выберите LUN из списка. В этом списке только те LUN, на которых нет VMFS. Важно! Среди них могут быть уже задействованные как RDM с другими ВМ, обращайте внимание на адреса и номера LUN во избежание ошибок и потери данных.
4. **Select Datastore** – здесь вы выбираете, на каком хранилище VMFS будет располагаться файл виртуального диска (vmdk), являющийся ссылкой на этот LUN. Скорее всего, вариант по умолчанию вас устроит.
5. **Compatibility Mode** – тип RDM-подключения, о нем чуть ниже.
6. **Advanced Options** – здесь мы, как и для файлов виртуальных дисков, указываем адрес SCSI добавляемого диска с точки зрения ВМ. SCSI (0:1) означает, что диск будет подключен на первый SCSI ID контроллера 0. А если мы выберем SCSI (1:0), то диск будет подключен как ID 0 контроллера 1. В частности, второй вариант означает, что в ВМ будет добавлен и второй контроллер SCSI – это часто нам надо для MSCS/MFC (первый SCSI-контроллер с номером 0 обычно уже существует, если добавляемый RDM – не первый диск этой ВМ). Если RDM **Virtual**, то мы можем поставить флажок **Independent**. Если он стоит, то к этому диску ВМ не будут создаваться снимки состояния (snapshot). Дополнительные настройки в режиме **Independent**:
 - **Persistent** означает монолитный диск, к которому не применяются снимки состояния (snapshot). Все изменения сразу пишутся на диск;

- **Nonpersistent** означает, что при включении ВМ именно для этого ее диска создается файл дельты, в который записываются все изменения. После выключения ВМ эта дельта отбрасывается. То есть диск в режиме nonpersistent автоматически возвращается в исходное состояние после выключения ВМ.

RDM бывает двух типов:

- ☐ **Physical** означает, что гипервизор подавляющее большинство команд SCSI пропускает до LUN без изменений;
- ☐ **Virtual** разрешает перехватывать и изменять команды SCSI.

С точки зрения использования, Virtual RDM не препятствует снятию снимков состояния (средствами ESXi) и позволяет клонировать и создавать шаблон из ВМ. То есть дает возможность RDM LUN использовать так же, как файл виртуального диска. Физические характеристики диска (LUN) будут скрыты.

Physical RDM дает прямой доступ к LUN. Пригодится для кластера MSCS/MFC в варианте cluster-across-boxes и physical-to-virtual. Однако если внутри ВМ у вас будет ПО, которому требуются прямой доступ на диск и работа с физическими характеристиками системы хранения, physical RDM – ваш выбор.

Выбирайте Virtual, если задача, под которую создается RDM, явно не требует использования physical RDM.

Однако в пятой версии vSphere появилось самое, наверное, заметное отличие этих режимов RDM – подключенный как physical RDM LUN может быть более 2 Тб размером.

Если к ВМ подключен RDM, то с ней можно осуществлять большинство операций типа VMotion, Storage VMotion и др. Также для VMotion необходимо, чтобы отдаваемый как RDM LUN был виден всем серверам (виден с точки зрения zoning и masking).

Невозможно как RDM подключить раздел – только LUN целиком.

Управлять путями к RDM LUN можно точно так же, как к LUN с VMFS. Только доступ к этим настройкам осуществляется из другого места – зайдите в свойства ВМ, выделите ее диск RDM и нажмите кнопку **Manage Path**.

Иногда ESXi не позволяет подключить LUN как RDM. Обычно это происходит, когда LUN подключен к локальному контроллеру.

Можно попробовать проделать следующее: **Configuration** ⇒ **Advanced Settings** для Software ⇒ **RDM Filter** ⇒ снять единственный флажок **RDMFilter.HBAisShared**.

Если не помогло, то можно попробовать из командной строки:

```
vmkfstools -r /vmfs/devices/disks/naa.5xxxxxxxxxxx VM1_rdm.vmdk
```

С помощью этой команды вы создадите файл vmdk с именем VM1_rdm.vmdk, который будет являться ссылкой на LUN/диск с идентификатором naa.5xxxxxxxxxxx. Затем следует подключить этот файл vmdk к виртуальной машине через **Add HDD** ⇒ **Use Existing vmdk**.

Идентификатор устройства (вида пaa., eui., vpx.) можно посмотреть через клиент vSphere: **Configuration** ⇒ **Storage Adapters** ⇒ выбираем нужный контроллер ⇒ в нижней части экрана смотрим на доступные через него диски. В контекстном меню LUN, кстати, будет пункт **Copy Identifier** – с его помощью вы можете скопировать идентификатор LUN в буфер обмена.

Обратите внимание. Подключенный к виртуальной машине RDM LUN не является препятствием для VMotion. Однако если у виртуальной машины по умолчанию изменено значение настройки SCSI Bus Sharing (это настройка виртуального контроллера SCSI), то тогда VMotion для нее будет невозможен. RDM LUN должен быть подключен к контроллеру SCSI с таким значением настройки SCSI Bus Sharing, если виртуальная машина является узлом кластера Майкрософт и узлы этого кластера запущены на разных физических серверах (а также в других случаях, когда требуется подключить LUN к VM с разных серверов).

3.8. NPIV

NPIV – это стандарт, описывающий, как один порт FC HBA может регистрировать несколько WWN на коммутаторах FC. ESXi поддерживает NPIV, в данном смысле это означает возможность сгенерировать для каждой VM уникальный WWN.

Теоретически эта функция пригодится нам для:

- ☐ анализа нагрузки на SAN со стороны отдельной VM, чтобы отделить трафик одной VM от всего остального по ее уникальному WWN;
- ☐ осуществления зонирования и презентирования LUN для отдельных VM;
- ☐ предоставления определенного качества обслуживания для отдельных VM;
- ☐ улучшения производительности для отдельных виртуальных машин путем индивидуальной настройки кеширования на уровне СХД.

На практике же я вижу данную функцию малоиспользуемой. Причин тому несколько:

- ☐ уникальным WWN помечается только трафик к RDM LUN, который так и так презентован одной (за исключением кластерных конфигураций) VM;
- ☐ для представления оговоренного качества обслуживания появилась эффективная и простая в использовании функция SIOC (доступна не для всех лицензий vSphere);
- ☐ эти LUN все равно должны быть доступны (с точки зрения зонирования и презентирования) каждому серверу ESXi, где использующая LUN виртуальная машина может оказаться;
- ☐ для виртуальной машины от включения NPIV не меняется ничего. Как VM работала с локальным SCSI-контроллером, так и продолжает. NPIV – это указание гипервизору, какой WWN подставлять в соответствующие обращения на СХД.

Однако два применения NPIV мне кажутся более оправданными:

- ☐ если SIOC мы не можем использовать (например, потому что наша лицензия этого не позволяет), то некоторые FC HBA позволят нам указать

приоритет для трафика с тем или иным WWN. Таким образом, мы можем приоритезировать трафик некоторых виртуальных машин (напомним, работает только для RDM LUN этих ВМ);

- ❑ если мы используем RDM, то без настроенного NPIV для ВМ с RDM нам сложно массово сопоставить виртуальные машины и прокинутые как RDM LUN с СХД.

Чтобы ESXi позволил настраивать NPIV для виртуальных машин, необходимо, чтобы HBA и коммутаторы FC поддерживали NPIV.

Сама настройка осуществляется в свойствах конкретной ВМ, на вкладке **Options** (рис. 3.30). Если для ВМ не указан свой уникальный WWN, ее обращения к RDM LUN исходят от WWN сервера.

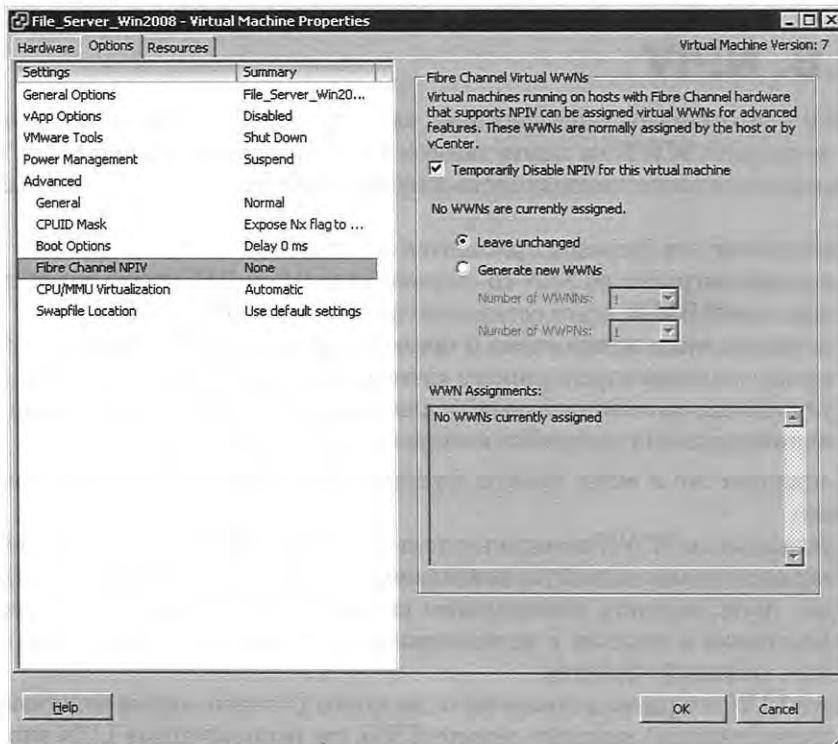


Рис. 3.30. Настройка NPIV

ВМ с включенным NPIV можно мигрировать с помощью VMotion, но нельзя с помощью Storage VMotion.

Есть еще некоторые мелкие нюансы, но их имеет смысл уточнять уже в документе vSphere **Storage ⇒ Configuring Fibre Channel Storage ⇒ N-Port ID Virtualization**.

3.9. Адресация SCSI

Для каждого LUN существуют несколько идентификаторов. Увидеть их можно, например, пройдя **Configuration** ⇒ **Storage** ⇒ кнопка **Devices**.

Вы увидите несколько столбцов с идентификаторами LUN:

- ❑ **Name** – это имя, сгенерированное для LUN самим ESXi. Для удобства его можно поменять на произвольное;
- ❑ **Identifier** – это имя LUN, сообщаемое системой хранения. Обычно одного из следующих форматов:
 - **naa** – Network Addressing Authority id, формат именования SCSI-устройств. Таким именем обычно представляются FibreChannel-устройства. Подобное имя всегда начинается со строки **naa**;
 - **t10** – еще один стандарт именования, некоторые системы хранения используют его. Такое имя всегда начинается со строки **t10**;
 - **iqn** – для iSCSI используются имена стандартов **iqn**. В отличие от **naa** и **t10**, идентификатор **IQN** не обязан быть уникальным глобально (**naa** и **t10** уникальны глобально, как MAC-адреса), однако, однажды настроенное на системе хранения iSCSI, это имя не должно меняться;
 - **eui** – для iSCSI может использоваться имя стандарта **eui**. Как и **naa** с **t10**, идентификатор **eui** уникален глобально;
 - **mpx** – для тех устройств, которые не представились именем по одному из стандартов **naa/t10/iqn/eui**, ESXi дает имя **mpx** (от VMware multipath X). Эти имена не являются уникальными глобально, и они могут меняться после перезагрузок. Обычно такие имена присваиваются локальным устройствам, чаще CD-ROM.
- ❑ **Runtime name** – активный путь к LUN вида **vmhba#:C#:T#:L#**. Здесь:
 - **vmhba#** – физический дисковый контроллер сервера;
 - **C#** – номер канала. Программный iSCSI-инициатор использует разные номера каналов для отображения разных путей к LUN;
 - **T#** – номер SCSI target. Номер таргета определяется сервером и может меняться в случае изменений в настройках видимости таргетов сервером. Один таргет для разных ESXi может показываться под разными номерами;
 - **L#** – номер LUN. Сообщается системой хранения.

Знание имени LUN пригодится вам для каких-либо настроек, на стороне СХД или в командной строке сервера ESXi.

Найти эти идентификаторы можно в графическом интерфейсе и в командной строке.

В графическом интерфейсе пройдите на вкладку **Storage Views** для сервера и выберите в выпадающем меню **Show all SCSI Volumes (LUNs)** (рис. 3.31).

Из командной строки можно выполнить команду

```
ls -l /vmfs/devices/disks/
```

и увидеть диски, доступные с этого сервера (рис. 3.32).

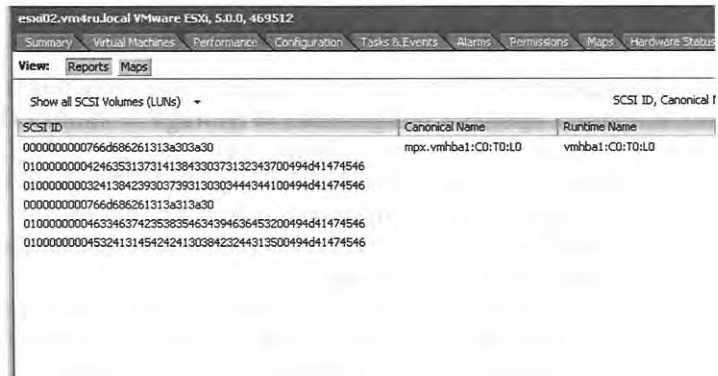
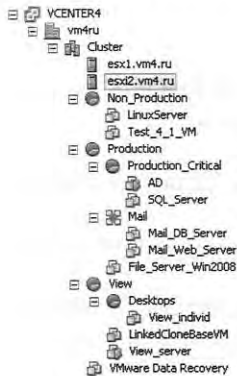


Рис. 3.31. Вкладка Storage Views

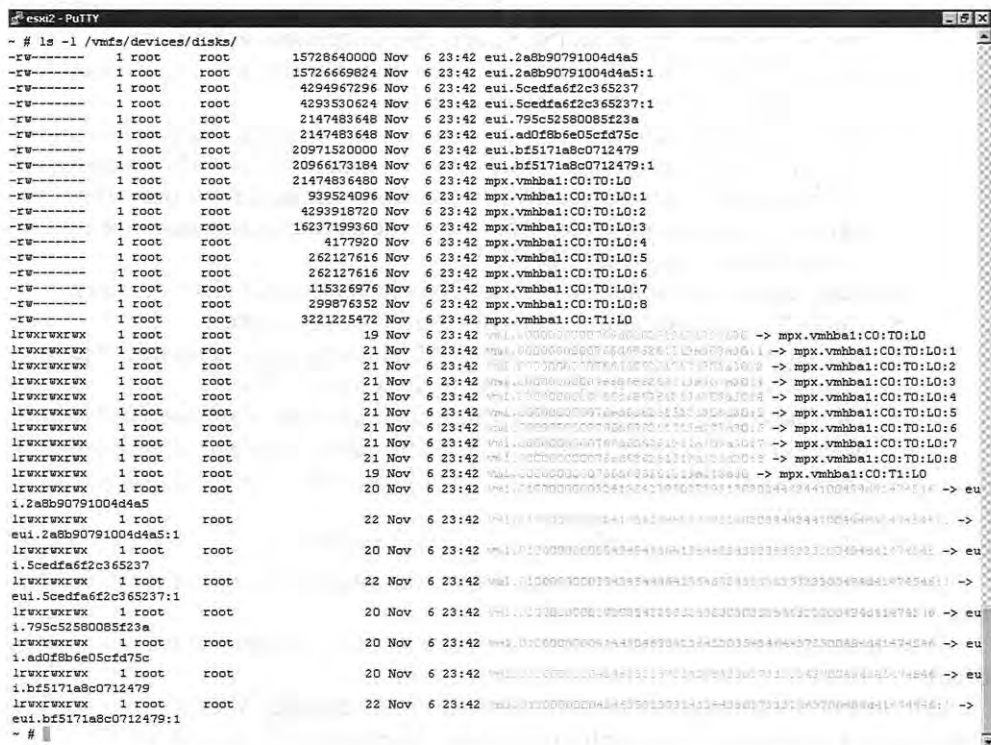


Рис. 3.32. Данные о дисках, полученные из командной строки

Также если перейти в **Configuration** ⇒ **Storage** ⇒ и в контекстном меню хранилища выбрать **Copy to Clipboard**, то в буфер обмена скопируется информация об этом разделе. Это простой способ связать имя VMFS и идентификатор (типа паа.) LUN.

Однако есть способ упростить идентификацию LUN. Пройдите **Configuration** ⇒ **Storage Adapters** ⇒ выделите контроллер ⇒ в нижней части окна отобразятся LUN. Если кликнуть два раза в столбце **Name**, то можно будет задать произвольное имя (рис. 3.33).

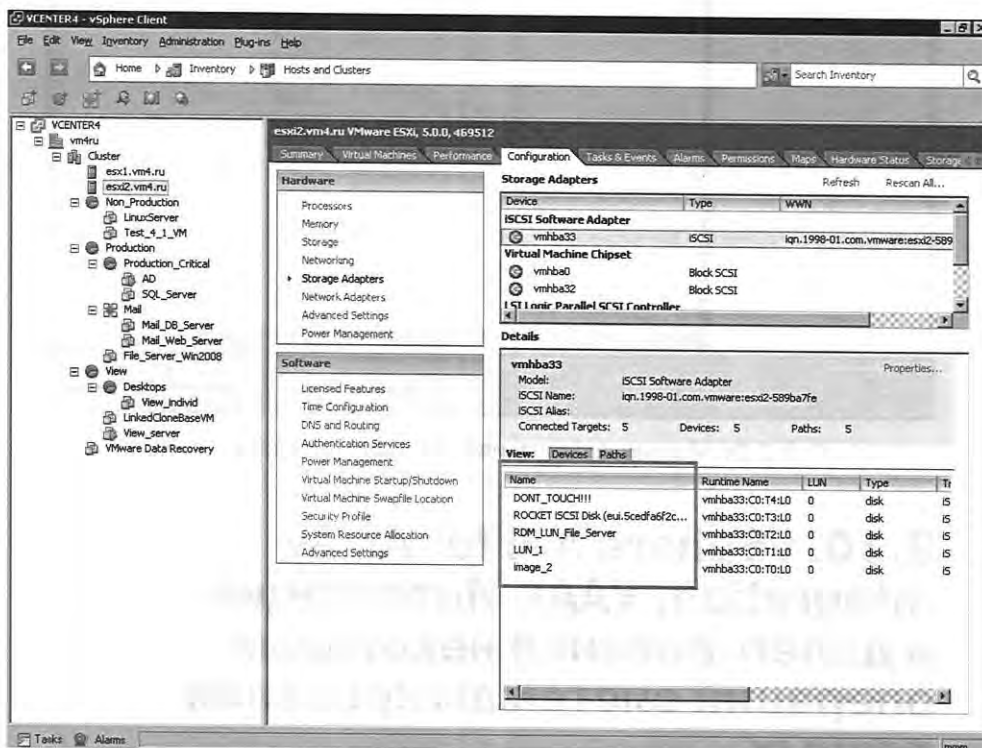


Рис. 3.33. Указание произвольного имени для LUN

Теперь это имя будет фигурировать в таких мастерах, как:

- ☐ создание хранилища VMFS;
- ☐ Extent или grow для хранилища VMFS;
- ☐ подключение LUN к ВМ как RDM.

Например, см. рис. 3.34.

Теперь не ошибиться при выборе диска в подобных операциях намного проще.

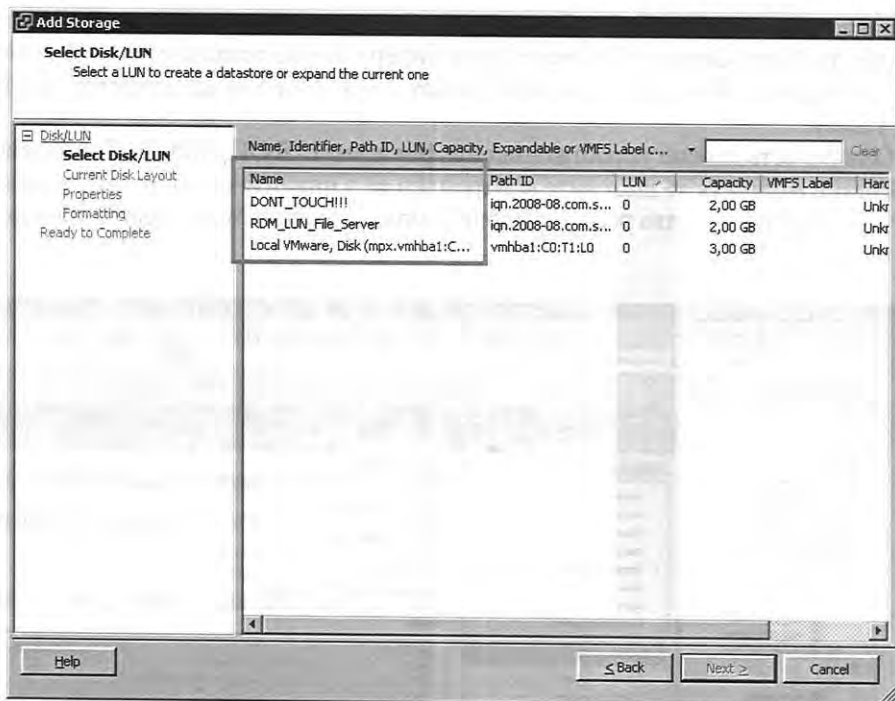


Рис. 3.34. Мастер добавления RDM к виртуальной машине

3.10. vSphere API for Array Integration, VAAI. Интеграция и делегирование некоторых операций системам хранения данных

Еще в ESXi версии 4.1 VMware реализовала поддержку так называемых vSphere API for Array Integration (VAAI), программных интерфейсов для интеграции с системами хранения. Суть этой технологии – в том, что если ваши сервера работают с системой хранения, поддерживающей этот интерфейс, то некоторые операции сервера могут не выполнять сами, а передать работу на СХД.

Таковыми атомарными операциями являются:

- ☐ создание полной копии файла – поможет при операциях vSphere, связанных с копированием;
- ☐ многократная запись одинакового фрагмента данных – поможет при обновлении виртуальных дисков (вспомните о thick provisioned eager zeroed дисках);

- ❑ блокировка отдельной области данных – поможет снизить вероятность того, что блокировки метаданных раздела VMFS окажут негативное влияние на производительность.

Список неполон, после разговора об этих возможностях, повышающих производительность некоторых операций, мы поговорим про другие возможности VAAI.

Перечислим операции vSphere, которые получают выигрыш от использования VAAI:

- ❑ Storage vMotion;
- ❑ развертывание ВМ из шаблона и клонирование ВМ;
- ❑ ускорение работы виртуальных машин, использующих «тонкие» диски, thin provisioning;
- ❑ ускорение создания «предобнуленных», eagerzeroedthick-дисков – они необходимы для защиты виртуальных машин при помощи VMware Fault Tolerance или Microsoft Failover Cluster.

В некоторых случаях возможны 10–20-кратное ускорение операций и значительное сокращение нагрузки на канал между системой хранения и сервером, а также на процессоры сервера.

Определить поддержку VAAI со стороны системы хранения можно в интерфейсе клиента vSphere (см. рис. 3.35).

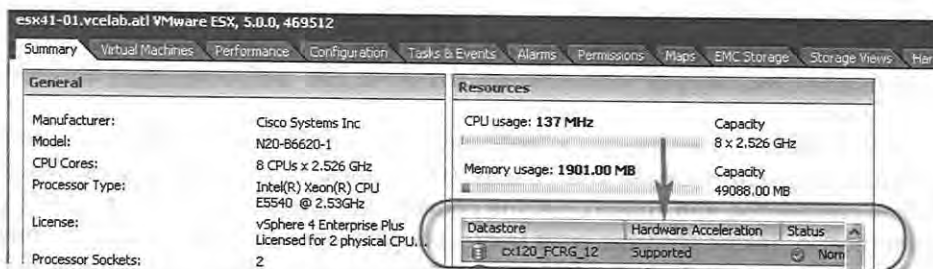


Рис. 3.35. Информация о поддержке VAAI в интерфейсе vSphere

Есть определенные ограничения для использования этого механизма:

- ❑ если содержимое RDM LUN копируется не на RDM LUN;
- ❑ если исходный vmdk типа eagerzeroedthick, а целевой – thin;
- ❑ если LUN, между которыми осуществляется операция, принадлежат разным системам хранения. Все операции, доступные через VAAI, возможны лишь между LUN одной системы хранения. Хранилища VMFS с extent поддерживаются также, лишь если все составляющие их LUN принадлежат одной СХД;
- ❑ если ваша инфраструктура мигрировала на версию 5 с какой-то из предыдущих версий, то VMFS вы могли не пересоздавать, а обновить до версии 5. А могли даже и не обновлять. В обоих случаях актуален следующий факт: при создании VMFS старых версий администратор выбирал размер блока (1/2/4/8 Мб) – при обновлении с VMFS 3 до VMFS 5 этот размер блока на-

следует. Если в операции копирования задействованы хранилища VMFS с разным размером блока – VAAI не работают. Если VMFS 5 создавался с нуля – проблема неактуальна, размер блока не поддается изменению и всегда равен 1 Мб (это не влияет на максимальный размер LUN и файла vmdk. Для VMFS-5 даже с блоком в 1 Мб файл vmdk может быть размером до 2 Тб, и LUN не ограничен 2 Тб после обновления на VMFS 5).

В пятой версии vSphere появилась возможность при помощи VAAI сообщать системе хранения о том, какие блоки были заняты уже не существующими сейчас файлами виртуальных машин.

Это пригодится в том случае, когда вы используете функционал thin provisioning на уровне системы хранения. Допустим, у вас есть «тонкий» LUN или ресурс pfs, чей реальный размер увеличивается лишь при записи данных в LUN. Если данных записано мало, то реальное потребление места таким LUN меньше, а то и много меньше его номинального объема.

Долго это работало только в одну сторону – при увеличении объемов информации на LUN его реальный размер рос, а при удалении информации – не уменьшался. А теперь будет уменьшаться, потому что ESXi сможет сообщить о том, какие блоки были заняты удаленными или мигрированными на другое хранилище файлами.

Есть еще одна функция интерфейсов VAAI для тонких LUN. Если произошло так, что реальный размер LUN расти не может, хотя своего максимального размера LUN еще не достиг, то ранее все ВМ на таком LUN останавливались. А теперь в подобной ситуации будут остановлены только те ВМ, файлы которых потребуют увеличения.

ВМ потребует увеличения в двух случаях – при существующем активном снапшоте или если у самой ВМ тонкий (thin) диск. В этих случаях увеличение файлов снапшотов или тонких дисков происходит блоками, по 16 и по 1 Мб соответственно. Таким образом, если место для расширения тонкого LUN закончилось, то это же произойдет для ВМ на нем, как только те запишут еще 1 или 16 Мб (максимум). А вот если на этом же хранилище расположены ВМ с «толстыми» дисками и без снапшотов – они продолжают работать.

Как идею можно рассмотреть возможность увеличить размер блока, которым происходит увеличение тонкого диска или дельты снапшота: если вы предполагаете использование thin provisioning на уровне системы хранения.

3.11. Profile-Driven Storage

Функция Profile Driven Storage, появившаяся в пятой версии vSphere, позволяет присвоить хранилищам метки (предполагается с информацией об их характеристиках) и затем выбирать для размещения ВМ хранилища в соответствии с этими метками.

В некоторых инфраструктурах могут быть хранилища с разными характеристиками. В первую очередь с разной производительностью и с разной стоимостью за гигабайт места. Сейчас если перед пользователем стоит задача выбрать храни-



лище для размещения на нем создаваемой ВМ или при миграции ВМ – он может ориентироваться только на название. Это не всегда удобно.

Система упомянутых меток призвана помочь решить эту проблему. Администратор помечает метками хранилища (также это может происходить автоматически – см. VASA). Теперь для ВМ мы можем выбирать хранилища с соответствующей меткой.

Про автоматически назначаемые метки будет рассказано в следующем разделе, здесь поговорим про те, что мы можем назначить собственноручно.

Для начала работы с этим механизмом пройдите **Home** ⇒ **VM Storage Profile** ⇒ кнопка **Manage Storage Capabilities** в верхней части окна. После нажатия **Add** в открывшемся мастере нам предложат описать возможные характеристики хранилищ нашей инфраструктуры. То есть мы просто указываем те самые метки, которые затем будем присваивать разным хранилищам. Первыми на ум приходят такие метки, как:

- ☐ медленное;
- ☐ быстрое;
- ☐ дешевое;
- ☐ дорогое.

В вашей собственной инфраструктуре, разумеется, набор меток может отличаться – в зависимости от того, какие характеристики хранилищ будут иметь для вас значение.

Однако на самом деле предложенная было классификация никуда не годится. Создаваемые на этом этапе метки мы будем присваивать хранилищам. И, к сожалению, больше, чем одну метку на хранилище, назначить нельзя. Это означает, что хранилище не может иметь метки, например, «быстрое» и «дорогое» одновременно.

Пример другой, более применимой классификации:

- ☐ бронза – хранилища для непроизводственных ВМ, с невысокими характеристиками производительности и доступности;
- ☐ серебро – хранилища для некритичных производственных ВМ;
- ☐ золото – хранилища для критичных производственных ВМ, максимальная производительность и доступность;
- ☐ только чтение – для шаблонов и образов iso.

Разумеется, список меток можно изменять произвольным образом (рис. 3.36).

Следующий шаг – сопоставить профиль хранилищу. Перейдите **Home** ⇒ **Datastores**. В контекстном меню хранилища выберите пункт **Assign User-Defined Storage Capability**. В открывшемся окне выберите из списка метку хранилища (**Storage Capabilities**). Повторите эту процедуру для каждого хранилища.

Следующий шаг – создать так называемый «профиль» хранилища, **Storage Profile**. Эти профили будут применяться к виртуальным машинам. Оставаясь в том же разделе интерфейса, **Home** ⇒ **VM Storage Profile**, нажмите кнопку **Create Storage Profile**.

Создаваемые профили – набор характеристик, созданных ранее. Если продолжить мой пример, то я могу определить хранилища как (к примеру):

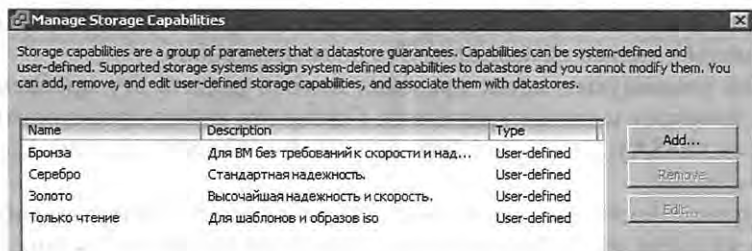


Рис. 3.36. Список меток для хранилищ

- ☐ бронза;
- ☐ серебро;
- ☐ золото;
- ☐ для шаблонов и образов iso.

Так что в запустившемся мастере:

- ☐ **Profile Properties** – укажите имя создаваемого «профиля» хранилища, и крайне желательно не пренебречь описанием. Так как данный «профиль» носит информационный характер, важно оставить максимум информации;
- ☐ **Select Storage Capabilities** – отметьте флажками те из ранее созданных характеристик хранилищ, которыми они должны обладать в рамках данного профиля. Если для вашей СХД настроены программные интерфейсы VASA, то вы можете увидеть на этом этапе характеристики хранилища, не созданные вами, а полученные от СХД.

Оставаясь все в том же разделе интерфейса, нажмите кнопку **Enable VM Storage Profiles** – в открывшемся окне вы можете (и должны один раз) включить данную функцию для тех серверов/кластеров ESXi, где она вам требуется. Данная функция входит не во все лицензии vSphere, и если в кластере будет хотя бы один сервер с лицензией без поддержки VM Storage Profiles – для кластера включить ее не удастся.

Теперь мы готовы к эксплуатации этого механизма. Например, при создании ВМ (с нуля или из шаблона) на шаге выбора хранилища мы можем выбрать нужный профиль хранилища в выпадающем меню **VM Storage Profile** – и система немедленно отобразит, какие из доступных хранилищ соответствуют выбранному профилю (рис. 3.37).

Как видите, создающий эту ВМ пользователь очень легко сопоставляет понятный ему (предполагается) профиль хранилища с самими хранилищами.

Впрочем, выбрать «несовместимое» хранилище тоже можно. Если произошло такое, то на вкладке **Summary** для этой ВМ мы увидим оповещение об этом несоответствии (рис. 3.38).

Кроме того, для каждого профиля можно получить информацию, на какие ВМ он назначен и соответствует ли текущее хранилище ВМ этому профилю. Для этого пройдите **Home** ⇒ **VM Storage Profile** ⇒ выберите профиль ⇒ вкладка **Virtual Machines** (рис. 3.39).

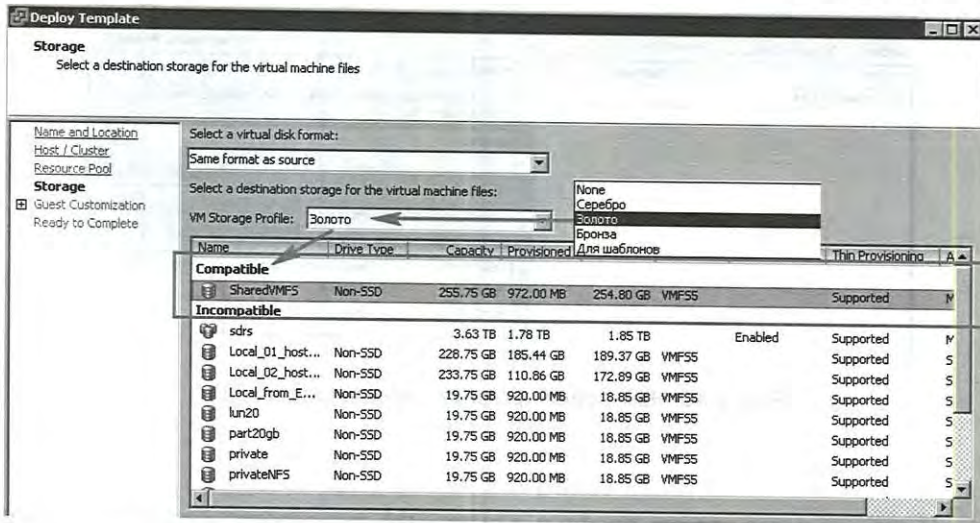


Рис. 3.37. Выбор хранилища в соответствии с профилем

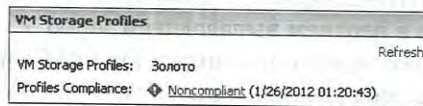


Рис. 3.38. Оповещения о несоответствии используемого хранилища профилю хранилища VM

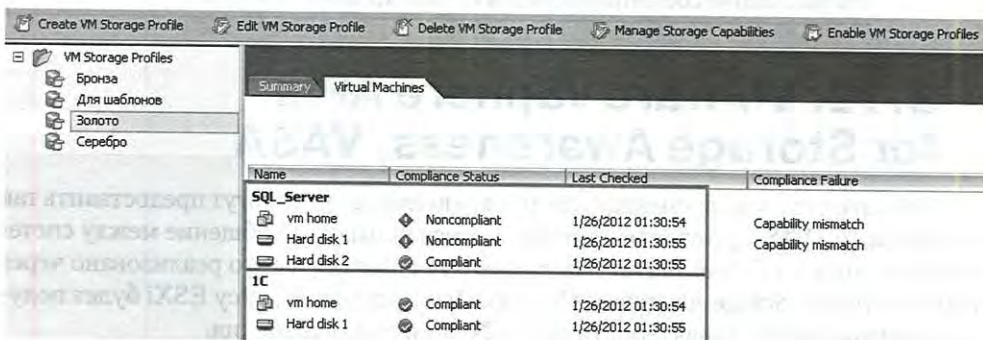


Рис. 3.39. Данные про все VM с определенным профилем хранилища

Обратите внимание на то, что для VM отдельно проверяется каждый виртуальный диск и отдельно – рабочий каталог. И действительно, зайдя в свойства VM ⇒ вкладка **Profiles**, мы можем указать для этих объектов профиль хранилища независимо (рис. 3.40).

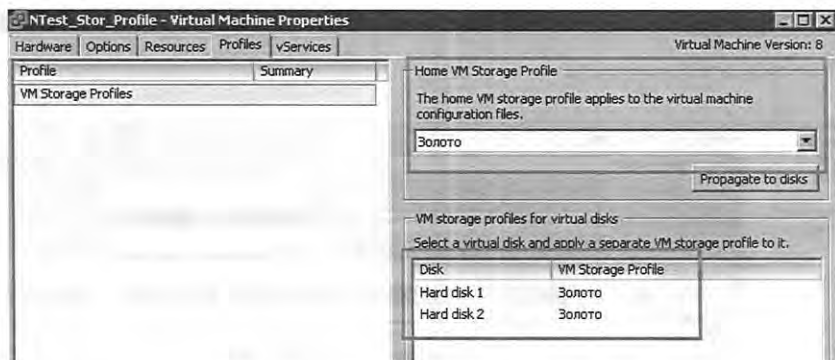


Рис. 3.40. Настройки профиля хранилища для VM

Здесь же можно указать профиль для тех VM, для которых он не был указан ранее. Обратите внимание, что в верхнем выпадающем меню **Home VM Storage Profile** мы выбираем профиль для рабочего каталога VM, внизу для каждого диска мы можем указать другие профили, и, важно(!), мы должны это сделать. Если указать профиль только в верхнем выпадающем меню – на диски он не распространяется. Кстати, тут нам может пригодиться кнопка **Propagate to disks** – нажав ее, мы распространим тот же профиль и для дисков этой VM.

Обратите внимание. Если вы планируете использовать и описываемые здесь профили хранилищ, и Storage DRS, то в Storage DRS-кластер следует объединять только хранилища с одинаковыми метками, чтобы профилям хранилищ одинаково соответствовали или не соответствовали все хранилища одного кластера.

3.12. VMware vSphere APIs for Storage Awareness, VASA

Производители современных систем хранения данных могут предоставить так называемый «VASA provider» – продукт, обеспечивающий общение между системой хранения и vCenter Server. На стороне vCenter Server это реализовано через vSphere API for Storage Awareness (VASA). По этому интерфейсу ESXi будет получать информацию о характеристиках LUN от системы хранения.

В принципе, такой информацией может быть что-то вроде:

- ☐ дедупликация (есть/нет);
- ☐ репликация (есть/нет);
- ☐ возможность снапшотов на уровне СХД (есть/нет);
- ☐ уровень RAID;
- ☐ тип дисков (SATA/SAS/FC/SSD);
- ☐ производительность в целом (IOps/MBs).

И смысл этого присвоения – в том, что для виртуальных машин можно будет выбирать хранилище не по названиям, а по этим характеристикам, которые автоматически сообщаются системой хранения.

Однако, к сожалению, на момент написания одному хранилищу можно было присвоить только одну характеристику. Поэтому де-факто если этим механизмом пользоваться, то сообщаемые системой хранения метки хранилищ должны содержать в себе описание всех важных для нас свойств, например:

- ☐ «RAID 5; Тонкие (Thin) LUN; Дедупликация; диски SAS»;
- ☐ «Быстрый LUN»;
- ☐ что-то вроде «Первый класс» или «Второй класс».

Система хранения с поддержкой VASA может сообщить хранилищу одну «метку», и еще одну может задать администратор – см. предыдущий раздел. Таким образом, одно хранилище как максимум может иметь две метки: одна – с системы хранения и одна – указанная администратором vSphere.

Для использования функционала интерфейсов VASA вам потребуется установить и настроить «провайдер VASA», предоставленный производителем вашей системы хранения. Так как это много разных продуктов (свой у каждого производителя), я вынужден за подробностями установки отправить в документацию производителя вашей системы хранения.

На момент написания вас интересуют следующие продукты:

- ☐ Dell – EqualLogic vSphere Plugin, являющийся частью Dell EqualLogic Host Integration Tools for VMware;
- ☐ EMC – Solutions Enabler (поставляется как в виде дистрибутива, так и в виде virtual appliance);
- ☐ NetApp – на момент написания соответствующий продукт NetApp находится на стадии бета-тестирования. Его название на этой стадии – NetApp VASA Provider;
- ☐ HP – vSphere management plug-in's, Insight Control Storage Module for vCenter Server и HP 3PAR Recovery Manager for VMware Software Suite.

3.13. Virtual Storage Appliance

Virtual Storage Appliance (VSA) можно перевести как «виртуальная система хранения». Это продукт, позволяющий взять два или три сервера ESXi, настроить зеркалирование между их локальными дисками и эти зазеркаленные диски сделать доступными всем этим серверам (и другим, при необходимости) как NFS-ресурсы. Таким образом, мы получаем разделяемую систему хранения, но программную. Подобная конструкция позволяет нам использовать функции живой миграции и высокой доступности, имея только 2–3 сервера и все, без аппаратной системы хранения.

На момент написания это решение обладает довольно большими ограничениями – их следует изучить перед внедрением. Однако продукт потенциально интересный.

Условия:

- ☐ 2 или 3 сервера ESXi, на которых будут работать виртуальные машины VSA. Эти сервера и эти ВМ образуют VSA-кластер. Этот кластер предоставляет NFS-ресурсы, а подключаться к ним могут как хосты – участники кластера, так и любые хосты ESXi;
- ☐ если vCenter запущен в ВМ, то эта ВМ не должна использовать VSA-хранилище или работать на сервере ESXi, который участвует в VSA-кластере;
- ☐ нельзя использовать Linux-версию vCenter – для работы VSA потребуется установить специальный VSA-сервис на vCenter, этот продукт существует только в Windows-варианте;
- ☐ для ВМ на VSA-хранилище рекомендуется выравнивание по границе 4 Кб, и размер одной операции IO должен быть равным или кратным 4 Кб;
- ☐ в серверах ESXi должно быть минимум 6 Гб ОЗУ, 4/6/8 дисков в RAID 5/6/10, 4 гигабитных порта Ethernet;
- ☐ сервера должны быть с настройками сети по умолчанию, как сразу после установки.

Суть довольно простая – поднимаем ВМ VSA на каждом сервере VSA-кластера. Каждая VSA ВМ использует локальные диски того сервера, где работает, и зеркалирует эти диски с другими VSA-ВМ этого кластера. За счет зеркалирования мы имеем отказоустойчивость – наши данные не теряются при плановом или неплановом выключении одного сервера ESXi.

Конфигурация кластера VSA на двух и на трех серверах ESXi различается. В случае двухузлового кластера на сервер vCenter необходимо будет установить вспомогательную службу VSA cluster service, которая будет выступать арбитром при обработке спорных ситуаций между двумя VSA-ВМ. Когда VSA-ВМ три – они сами разберутся.

Перед началом установки вам потребуется:

- ☐ 2 или 3 сервера ESXi (плюс еще на одном должен работать vCenter);
- ☐ ESXi, по сути, должны быть свежепоставленные. В частности, конфигурация сети должна быть по умолчанию. Не должно быть ни одной ВМ;
- ☐ потребуется по два IP-адреса на VSA-ВМ и по два IP-адреса для каждого сервера ESXi. Один IP на каждый VSA – для приватной сети, какая-нибудь отдельная подсеть, остальные – из подсети управления.

3.13.1. Ввод в VSA в эксплуатацию

Первый шаг – установка службы «VSA manager» на vCenter. Ничего сложного: setup.exe ⇒ next, next, finish.

После установки в клиенте vSphere появляется вкладка **VSA** для объекта Datacenter. В первый момент времени на ней предложат начать мастер установки VSA-кластера. Выберем **New Installation** ⇒ **Next**.

На шаге **Select Hosts** нам потребуется выбрать два или три сервера – будущих участника кластера VSA.

Следующий шаг – настройка сети (рис. 3.41). Нам потребуется указать:

- ❑ **VSA Cluster IP** – адрес ведущего узла кластера VSA. Используется в сервисных целях;
- ❑ если узлов в кластере VSA только два, то в спорных ситуациях будет необходим **VSA Cluster service** на vCenter. Ему потребуется **VSA Cluster Service IP**;
- ❑ **VSA management IP** – для каждой BM-VSA следует указать IP-адрес для управления;
- ❑ **VSA Datastore IP** – с этого адреса будет подключен NFS-ресурс. Указывается для каждой BM в кластере VSA;
- ❑ **VSA Featured IP** – будет создан интерфейс vmkernel с этим адресом. Через этот интерфейс будет производиться vMotion обычных BM (то есть напрямую к работе кластера VSA этот интерфейс не относится). Указывается для каждого сервера ESXi – участника кластера VSA.

В документации указано не использовать для этих адресов диапазон 192.168.xxx.xxx (откровенно говоря, я не смог найти причину такой рекомендации).

Reconfigure VSA Cluster Network

Configure Network

Enter static IP addresses for the VSA cluster network on the same subnet as the vCenter Server.

Configure Network

Verify Configuration

Reconfigure Network

► VSA Management ✓

▼ esxi-01.vm4ru.local ✓

Management IP Address: 192 . 168 . 22 . 228

Datastore IP Address: 192 . 168 . 22 . 227

vSphere Feature IP Address: 192 . 168 . 22 . 226 ☐ Use DHCP

Subnet Mask: 255.255.255.0

Gateway: 192.168.22.253

VLAN ID: 0

Back-end IP Address: 192 . 168 . 0 . 1

Back-end Subnet Mask: 255.255.255.0

Back-end VLAN ID: 0

► esxi-02.vm4ru.local ✓

► esxi-03.vm4ru.local ✓

Help < Back Next > Cancel

Рис. 3.41. Настройка сети для кластера VSA

В принципе, все. Ввод VSA в эксплуатацию весьма прост. Мастер сам развернет VSA-ВМ на хосты, сам создаст дополнительный vКоммутатор и группы портов, интерфейсы VMkernel, подключит ресурсы NFS.

После завершения работы мы увидим три хранилища NFS (рис. 3.42). Или два, если в VSA-кластере у вас только два сервера.

Identification	Status	Device	Drive Type
datastore1	Normal	Local VMware Disk (mpx.vmhba1:C0:T0:L0):3	Non-SSD
VSADs-0	Normal	192.168.22.227:/exports/bc4205fc-ce84-4f9c-b34b-e8ad9628b1fc	Unknown
VSADs-1	Normal	192.168.22.221:/exports/4f6fb257-7413-4038-97fd-20ae6913be5	Unknown
VSADs-2	Normal	192.168.22.224:/exports/8e259477-8622-4ac4-b5b1-9e972372a356	Unknown

Рис. 3.42. Хранилища VSA

На вкладке **VSA Manager** для Datacenter будет отображаться статус кластера.

The screenshot shows the vSphere VSA Manager interface. The left sidebar displays a tree view with 'VSA HA Cluster' and its components: 'vsa-01.vsan.local', 'vsa-02.vsan.local', 'vsa-03.vsan.local', 'VSA-0', 'VSA-1', and 'VSA-2'. The main panel is titled 'VSA Manager' and contains the following sections:

- VSA Cluster Properties:**
 - VSA Cluster Status:** Name: vStorage Cluster, Status: Online.
 - VSA Cluster Network:** IP Address: 192.168.22.229, Netmask: 255.255.255.0, Gateway: 192.168.22.253.
 - Capacity:** Physical Capacity: 42.00 GB, Storage Capacity: 21.00 GB, Time since last data update: 00:18.
- View:** Datastores, Appliances, Map.
- Datastore Table:**

Name	Status	Capacity	Free	Used	Exported By	Datastore Address	Datastore Netmask
VSADs-0	Online	6.99 GB	6.85 GB	143.79 MB	VSA-1	192.168.22.227	255.255.255.0
VSADs-1	Online	6.99 GB	6.85 GB	143.79 MB	VSA-2	192.168.22.221	255.255.255.0
VSADs-2	Online	6.99 GB	6.85 GB	143.79 MB	VSA-0	192.168.22.224	255.255.255.0
- Datastore Properties:**
 - Name:** VSADs-0
 - Status:** Online
 - Exported By:** VSA-1
 - Datastore Path:** //192.168.22.227/exports/bc4205fc-ce84-4f9c-b34b-e8ad9628b1fc
 - Datastore Network:** IP Address: 192.168.22.227, Netmask: 255.255.255.0, Gateway: 192.168.22.253, VLAN ID: 0.
 - Capacity:** Total: 6.99 GB, Used: 143.79 MB, Free: 6.85 GB.
 - Replication:** VSADs-0 (Online), VSA-1 (Online), VSA-0 (Online).

Рис. 3.43. Статус кластера VSA

Необходимые элементы сети будут созданы автоматически (рис. 3.44).

Через верхний вКоммутатор (Front End) идут управление и NFS-трафик, через нижний (Back End) – узлы VSA-кластера обмениваются сигналами пульса и реплицируют диски. Также на vSwitch1 создаются интерфейсы VMkernel для vMotion.

3.13.2. Эксплуатация VSA

С точки зрения использования, VSA – это просто два или три хранилища, на которых можно создавать ВМ.

А что с обслуживанием?

Если ломается Front End сеть одного из хостов, то одно из NFS-хранилищ отваливается. ВМ, расположенные на нем, становятся недоступными. И, в общем-то, все.

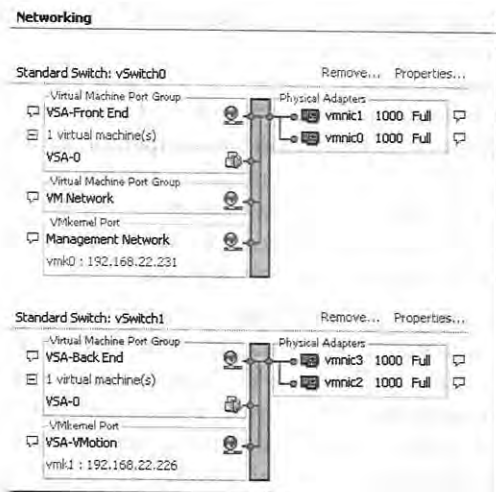


Рис. 3.44. Сетевые объекты, добавленные при настройке VSA

А если ломается Back End, то не происходит ничего: ВМ просто продолжают работать – так как хранилища зеркалируются между VSA-BM и в этом случае отработывает переход по отказу. Сюда же попадает ситуация, когда ломается один из серверов ESXi целиком.

В случае проблем один из узлов кластера VSA можно заменить штатными средствами, из GUI – на вкладке **VSA Manager** для Datacenter.

Однако добавить в кластер их двух узлов третий узел нельзя – следует сразу создавать VSA-кластер из требуемого количества узлов. Также, добавив дисковых ресурсов на сервера ESXi, нельзя добавить их в VSA.

3.13.3. Размышления про применимость

VSA обладает некоторыми особенностями, многие из которых попадают, к сожалению, в разряд минусов.

Видимая вендором схема использования VSA – инфраструктура с тремя-четырьмя серверами, которую развертывают и сразу начинают использовать с VSA.

Большая головная боль:

1. vCenter необходим.
2. Если vCenter запущен на ESXi-участнике кластера VSA, то его неудобно располагать на VSA-хранилище. При некоторых комбинациях проблем проблемы с VSA сделают недоступным такой vCenter, а без него нельзя будет починить VSA. Получается, vCenter должен или работать на отдельном сервере, где VSA компонентов нет, или использовать не VSA-хранилище. А раз мы используем VSA – вряд ли у нас есть еще какое-то хранилище, кроме локальных дисков. А если vCenter расположен на локальных дисках, то к нему не применяются VMware HA и vMotion.

Получается, если у нас два или три сервера ESXi в кластере VSA, то требуется еще один сервер. Этот «еще один»:

- ☐ или с физическим vCenter;
- ☐ или с ESXi, на котором работает vCenter. vCenter может быть расположен на VSA-хранилище с формальной точки зрения, но если сломается хранилище, чинить его будет затруднительно – так как недоступен vCenter, если он расположен на отказавшем хранилище. Этот сервер ESXi может использовать VSA-хранилище и быть объединенным в кластер HA с остальными серверами.

VSA использует локальное хранилище серверов ESXi, однако из-за зеркалирования только половина места доступна для размещения ВМ.

Однако в будущих версиях vSphere и VSA ситуация может меняться, так что не воспринимайте эти соображения как истину в последней инстанции.

Приведенная здесь информация достаточна для ввода VSA в эксплуатацию. Но некоторые детали остались за кадром. Искать их следует в документе vSphere Storage Appliance. Ссылка на него и некоторые другие полезные ссылки удобно найти здесь: <http://link.vm4.ru/vsa>.

Глава 4. Расширенные настройки, безопасность, профили настроек, решение проблем

4.1. Расширенные настройки (Advanced settings)

Если вы пройдете Configuration ⇒ Advanced Settings для Software, то увидите список разнообразных расширенных настроек ESXi (рис. 4.1).

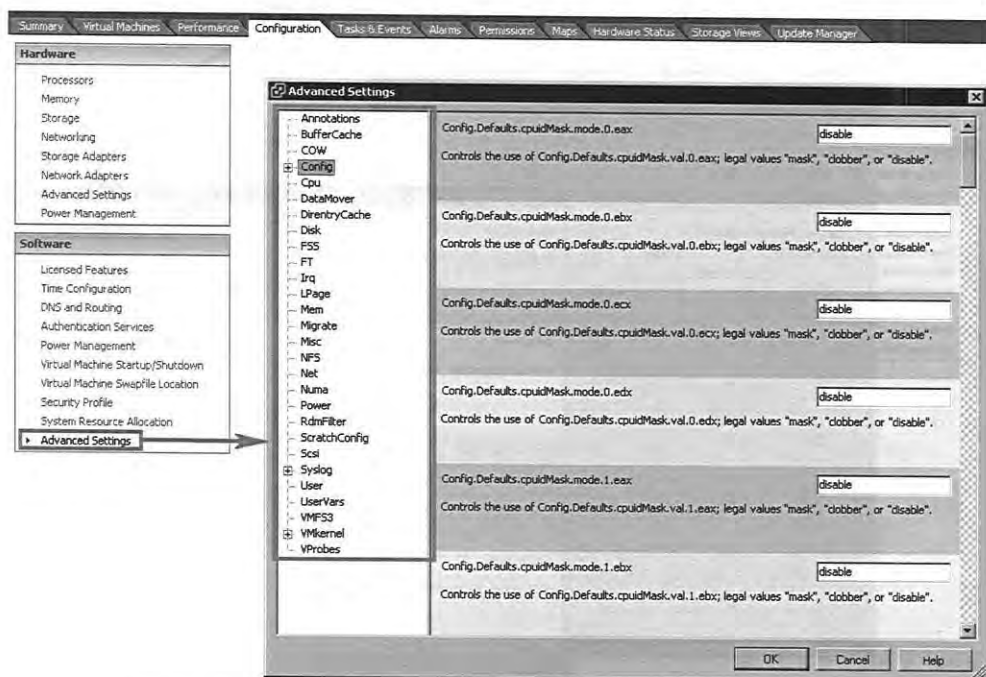


Рис. 4.1. Расширенные настройки (Advanced Settings) сервера ESXi

Изменять эти настройки нужно очень аккуратно, и только те из них, в назначении которых вы уверены. Как правило, рекомендации по их изменению вам может дать поддержка VMware как средство решения каких-то проблем. Или какая-то

статья в базе знаний VMware (<http://kb.vmware.com>). Небольшая часть этих настроек описана в документации.

Например:

Disk.MaxLun – в этом параметре мы указываем максимальный номер LUN, до которого ESXi опрашивает систему хранения при операции rescan. Если максимальный номер LUN, который мы используем, например, 15, то, указав Disk.MaxLun = 16, мы сократим время rescan.

В частности, в документации неплохо описаны расширенные настройки для iSCSI. Они, кстати, единственные, которые доступны в другом месте интерфейса (рис. 4.2).

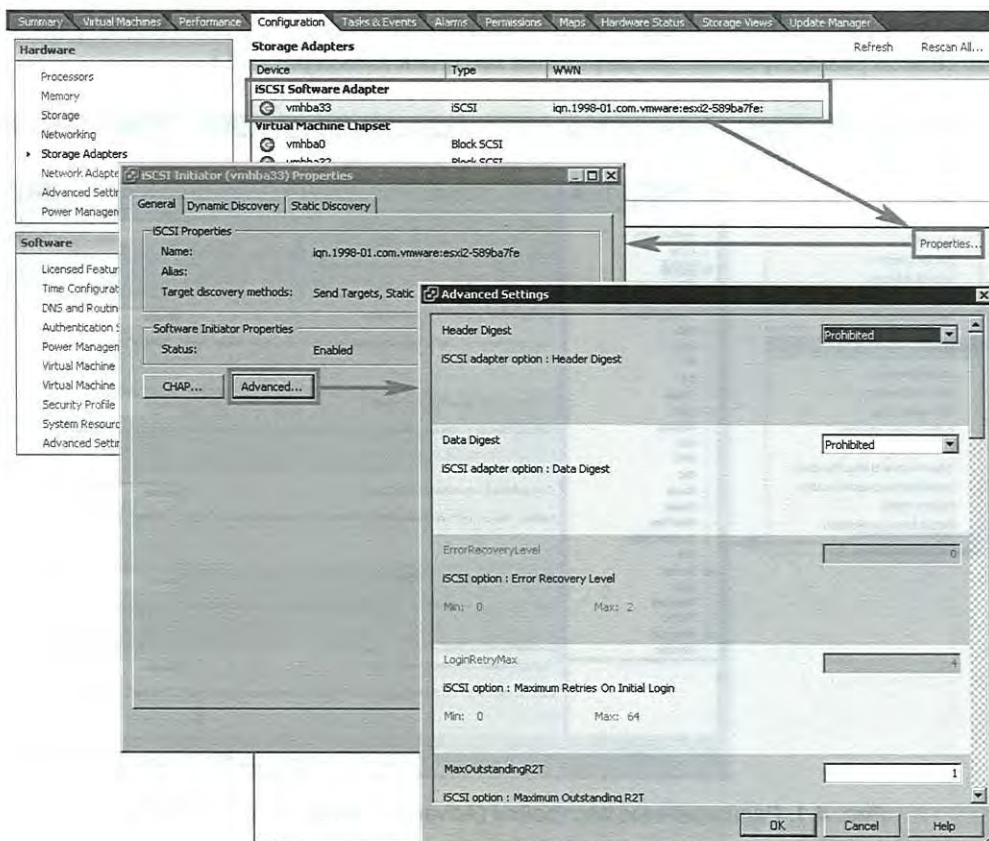


Рис. 4.2. Расширенные настройки (**Advanced Settings**) для программного инициатора iSCSI

4.2. Безопасность

Как и везде, в виртуальной инфраструктуре безопасность очень важна. В этом разделе поговорим про разные аспекты безопасности. И в общем – что нужно иметь в виду при разговоре о безопасности в контексте vSphere, и в частности – про брандмауэр, который появился в пятой версии ESXi; про системы раздачи прав, которые реализованы на ESXi и vCenter; про настройки там и здесь, которые имеют отношение к безопасности.

Если для вас стоит вопрос обеспечения безопасности по критериям, более жестким, чем это сделано по умолчанию, то первоисточником информации для вас станет документ VMware Security Hardening Guide (<http://communities.vmware.com/docs/DOC-15413>). На момент написания документ существовал для vSphere 4.1, но его актуальность для пятой версии я бы оценил как довольно высокую.

4.2.1. Общие соображения безопасности

Здесь мне кажется оправданным дать представление об отличиях виртуальной инфраструктуры с точки зрения безопасности.

Естественно будут выделить несколько уровней виртуальной инфраструктуры, подход к защите которых различается:

- ☐ уровень виртуализации (гипервизор, VMkernel);
- ☐ уровень виртуальных машин;
- ☐ виртуальная сеть на ESXi;
- ☐ системы хранения;
- ☐ vCenter Server.

Виртуальные машины. Подход к обеспечению безопасности ВМ (гостевых ОС и приложений) такой же, как и при развертывании тех же ОС и приложений на физической инфраструктуре, – антивирусы, сетевые экраны и прочее. Однако не все из привычных средств применимы – аппаратные межсетевые экраны малоэффективны, так как трафик между виртуальными машинами одного сервера не покидает пределов этого сервера и не доходит до физической сети.

Скомпрометированная виртуальная машина может перемещаться между серверами и компрометировать виртуальные машины на других серверах.

Также VMware предлагает набор программных интерфейсов (API) под названием VMSafe. С его помощью приложение из одной ВМ может получать доступ к другим ВМ на этом ESXi через гипервизор. Например, мы можем иметь одну ВМ с антивирусом, которая будет сканировать память всех прочих ВМ на этом сервере. В каких-то продуктах могут быть реализованы и другие полезные функции, например сканирование выключенных ВМ.

Обратите внимание. VMware предлагает свое решение межсетевого экрана для ВМ – vShield Zones. Кроме того, есть достаточное количество средств обеспечения безопасности от других производителей (см. <http://link.vm4.ru/sec>).

Несмотря на то что несколько ВМ работают на одном сервере, как-то перераспределяют его память, их совместная работа не ухудшает ситуацию с безопасностью. Изнутри одной ВМ нет способа получить доступ внутрь другой через гипервизор, кроме специализированных API VMsafe. Использование этих API требует четкого понимания механизмов работы гипервизора и целесообразности включения в эту работу. По умолчанию эти API не используются никем. Чтобы они начали работать, потребуется в явном виде предпринять ряд действий, как то: установить соответствующее ПО или Appliance, запустить его и настроить.

VMkernel. Гипервизор ESXi. Вопрос безопасности гипервизора достаточно важен, так как захват гипервизора позволит злоумышленнику перехватывать данные абсолютно незаметно для традиционных средств защиты, работающих внутри ВМ. Это и данные сетевых и дисковых контроллеров, и, теоретически, даже обращения к ОЗУ. Разумеется, под угрозой – доступность виртуальных машин.

Плюс к тому в силу повышающегося процента консолидации компрометация гипервизора влияет на все большее количество виртуальных машин, работающих на этом сервере.

Из средств внутренней защиты присутствуют только брандмауэр и система разграничения прав. Если вопрос безопасности стоит остро, то имеет смысл обратиться к сторонним производителям средств безопасности, в чьих активах уже есть специализированные программные продукты для ESXi.

Зато гипервизор является узкоспециализированной операционной системой, как следствие – поверхность атаки значительно меньше, чем для традиционных операционных систем. В простых случаях безопасность гипервизора обеспечивается прежде всего изоляцией его сети. Сеть для VMotion, для Fault Tolerance, сеть для трафика iSCSI и NFS желательно выделять в отдельные физические сети или VLAN.

Разумеется, интерфейсы управления также рекомендуется помещать в изолированную сеть. Еще в составе ESXi 5 есть брандмауэр, который имеет смысл настраивать на блокирование всех незадействованных портов (впрочем, так происходит по умолчанию). Про настройку этого брандмауэра я расскажу позднее.

VMware регулярно выпускает обновления, закрывающие уязвимости, найденные в VMkernel (впрочем, таковых было известно очень мало). Своевременная установка обновлений – важная часть обеспечения безопасности. VMware предлагает специальный компонент – VMware Update Manager, который дает возможность устанавливать обновления в автоматическом режиме на сервера ESXi. О нем буду рассказывать в последней главе.

С точки зрения безопасности, полезным шагом является настройка централизованного журналирования. Например, VMware предлагает Syslog Collector – предустановленный в Linux-версии vCenter и в виде отдельной службы для Windows. Эту службу чертовски легко настроить на сбор файлов журналов с серверов ESXi.

Обратите внимание. К сожалению, сегодня невозможно использовать большинство аппаратных средств обеспечения безопасности, так как применение их для гипервизоров требует реализации специфичных возможностей и специальных драйверов.

VMware vCenter – это приложение для Windows или, в пятой версии, для Linux. Каких-то специальных средств обеспечения безопасности для него не существует. Но стандартные для таких задач – это опять же установка обновлений, настройка межсетевого экрана и антивирус, помещение vCenter в изолированную сеть. Всем этим надо пользоваться.

Возможно, у вас доступ к управлению виртуальной инфраструктурой, то есть к vCenter через клиент vSphere, будут иметь несколько людей. В таком случае крайне желательным является жесткое разграничение прав. О механизме разграничения прав в vCenter вскоре расскажу.

Виртуальные сети. Виртуальные коммутаторы VMware имеют некоторые отличия от коммутаторов физических в силу своей природы. Например, в Коммутаторы не поддерживают Spanning tree, не устанавливают соответствия MAC-адресов и портов путем анализа трафика и т. д. Данные отличия вытекают из особенностей эксплуатации в Коммутаторов – они никогда не пересылают пришедших извне пакетов обратно «наружу», поэтому не могут стать причиной петли для физического коммутатора, следовательно, Spanning tree и не нужен. MAC-адреса VM им сообщает гипервизор, генерирующий эти MAC-адреса для VM, и анализировать трафик для обучения нет нужды и т. д.

Вывод: с сетевой безопасностью нет особо усложняющих работу нюансов. Разумеется, надо знать и понимать важные настройки в Коммутаторах, в первую очередь из группы настроек **Security**.

Хранилища. Безопасность хранилищ для VM в основном сводится к правильному зонированию и маскировке LUN для серверов ESXi. Так как сами VM не знают о физическом хранилище ничего (у VM нет доступа к HBA/iSCSI инициатору/клиенту NFS), с их стороны угроз безопасности хранилищ нет. Для полноты картины стоит упомянуть о том, что виртуальная машина может получить доступ к дисковому контроллеру с помощью функции VMDirectPath, и о том, что системы хранения iSCSI и NFS могут быть подключены по сети сразу к виртуальной машине, не через гипервизор. Но такие конфигурации уже не связаны с вопросом безопасности гипервизора и его служб.

Впрочем, теоретически компрометированная VM может вызвать повышенное потребление места на хранилище (если у нее тонкий диск с большим максимумом или есть снимки состояния). А также вызвать большую нагрузку на дисковую подсистему. Защититься от первого помогут организационные меры – разделять VM разного уровня важности по разным хранилищам, избегать снапшотов и тонких дисков. От второго – разделение по разным хранилищам и функция Storage IO Control.

4.2.2. Брандмауэр ESXi

В состав ESXi пятой версии входит брандмауэр. Он защищает интерфейсы VMkernel. Это означает, что к виртуальным машинам он отношения не имеет.

В отличие от ESX версий 3 и 4, брандмауэр которых был построен на базе iptables, брандмауэр ESXi построен с нуля.

По умолчанию он настроен на блокировку всех портов, кроме явно необходимых для работы ESXi. При активации стандартных служб и функций ESXi (таких как ssh, ntp, vMotion и прочих) необходимые порты открываются автоматически.

Если вдруг потребовалось поменять настройки, то некоторые вещи можно сделать из графического интерфейса.

Для настройки из графического интерфейса перейдите в **Configuration** ⇒ **Security Profile** ⇒ кнопка **Properties** (рис. 4.3).

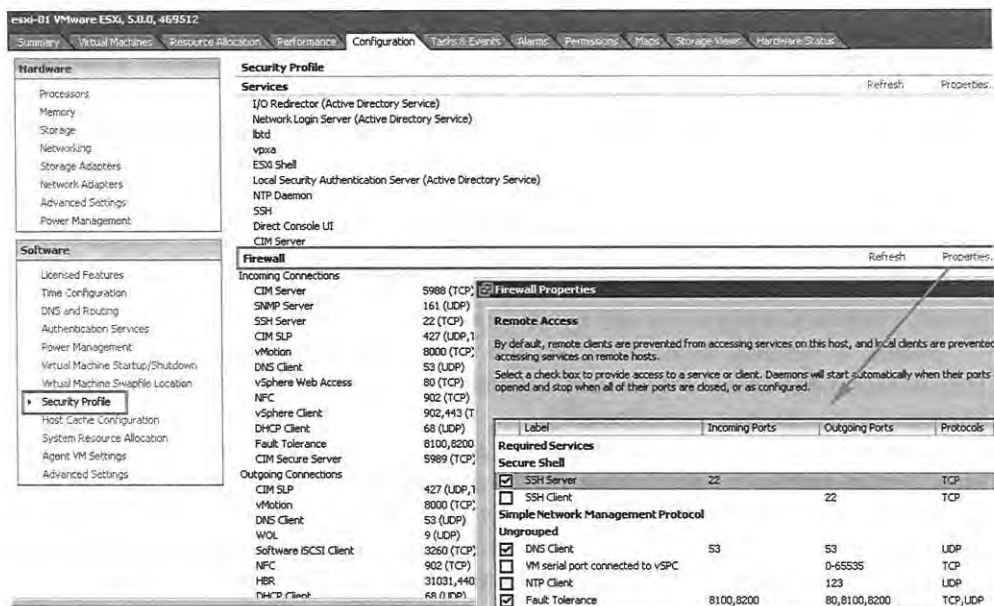


Рис. 4.3. Настройки брандмауэра ESXi из графического интерфейса

Здесь вы увидите описанные для брандмауэра службы, какие порты и направления им соответствуют. Если эти настройки вас устраивают, то разрешить работу в указанных направлениях по указанным портам можно, просто расставляя флажки.

Кнопка **Options**, активная при выборе некоторых служб в окне настройки брандмауэра, позволяет получить доступ к настройкам старта этих служб.

Кнопка **Firewall** в том же окне позволит ограничить доступ к ESXi, разрешив его только с явно указанных IP-адресов и/или подсетей.

Тиражировать конфигурацию брандмауэра ESXi 5 можно при помощи механизма **Host Profiles**.

Более глубокую настройку брандмауэра можно выполнить из командной строки. В этом нам поможет команда `esxcli network firewall`.

Для того чтобы описать для брандмауэра новый сервис (и получить возможность управлять его портами в графическом интерфейсе), нам следует описать его/их в файле конфигурации.

Воспользуйтесь командной строкой или графическим файловым менеджером (я предпочитаю WinSCP), для того чтобы создать xml-файл с произвольным именем в каталоге /etc/vmware/firewall.

В командной строке это будет выглядеть примерно так:

```
cd /etc/vmware/firewall
vi CustomManagementAgent.xml
```

В этом файле опишите необходимое количество сервисов по следующей схеме:

```
<ConfigRoot>
  <service>
    <id>CustomManagementAgent</id>
    <rule id='0000'>
      <direction>inbound</direction>
      <protocol>tcp</protocol>
      <porttype>dst</porttype>
      <port>1234</port>
    </rule>
    <enabled>>false</enabled>
    <required>>false</required>
  </service>
</ConfigRoot>
```

Какие секции здесь стоит выделить:

- ☐ секций `<rule>` `</rule>` может быть несколько – в каждой описан один порт;
- ☐ `Enabled` – открыты ли порты по умолчанию, сразу после загрузки этого правила;
- ☐ `required` – можно ли отключать этот сервис (то есть закрывать порты);
- ☐ `rule id` – порядковый номер. 0000, 0001, 0002 и т. д.;
- ☐ `protocol` – TCP или UDP. Если требуется открытый порт и там, и там – придется создавать отдельные правила;
- ☐ `direction` – `incoming` или `outgoing`. Если порт должен быть открыт в обоих направлениях – придется создавать отдельные правила;
- ☐ 1234 – номер порта.

После описания сервиса в этом файле выполните команду

```
esxcli network firewall refresh
```

Сервис с указанным именем теперь может быть активирован в графическом интерфейсе настроек брандмауэра, и его активация откроет соответствующие порты.

Изучите пространство имен `esxcli network firewall` – с его помощью можно взаимодействовать с брандмауэром из командной строки.

4.2.3. Аутентификация на серверах ESXi, в том числе через Active Directory

Первое, о чем стоит сказать, начиная разговор про раздачу прав, – это о том, куда мы обращаемся. Вариантов два – на ESXi или vCenter. Разберем первый вариант.

Я вижу следующие организационные варианты:

- 1) вам требуется работать напрямую с ESXi. Например, этого требует какое-то стороннее ПО;
- 2) вам не требуется работать напрямую с ESXi;
- 3) вам требуется жестко запретить работу напрямую.

Вариант 1 – вам требуется подключаться напрямую

Если у вас не окажется какого-то стороннего ПО, которому требуется прямое подключение к ESXi, то работать с ESXi напрямую (в командной строке или клиентом) придется, скорее всего, мало.

Клиентом напрямую следует подключаться только в том случае, если недоступен vCenter, – это настоятельная рекомендация VMware, да и просто это логично и удобно.

Из командной строки – в основном в случае возникновения каких-то проблем, не решаемых другими путями. Таких проблем, рискну предположить, у вас вряд ли будет много. Или для специфических настроек, которые выполняются разово. Притом для большинства манипуляций из командной строки VMware рекомендует применять vSphere CLI/Power CLI, нежели локальную командную строку. А эти средства позволяют авторизоваться на vCenter, а затем работать с сервером под его управлением без дополнительной авторизации (хотя бывают и исключения, например командлеты PowerCLI для настроек SNMP – они работают только при прямом обращении на хост).

Допустим, причина подключаться напрямую у вас или ваших коллег есть.

Итак, вы запускаете vSphere Client, вводите IP-адрес или имя сервера ESXi. Вам необходимо ввести имя пользователя и пароль. Или вы запустили клиент SSH типа PuTTY, и вам также необходимо авторизоваться. Без дополнительных телодвижений авторизовываться вы будете пользователем Linux.

Но тут возникают неудобства в том случае, если политики безопасности вашей компании предполагают такие вещи, как контроль сложности пароля, политики смены пароля и т. п., – ведь вам придется контролировать пользователей на каждом ESXi независимо. Это неудобно.

Однако, начиная с версии еще 4.1, в сервере ESXi очень легко настроить авторизацию учетными записями Active Directory.

Для реализации этой возможности необходимо пройти в настройки сервера – вкладка **Configuration** ⇒ **Authentication Services**.

В выпадающем меню выбрать метод аутентификации **Active Directory**, затем указать домен. Если вы хотите разместить записи серверов ESXi не в корневом контейнере AD, то домен укажите не в виде, например, «vm4.ru», а в виде, напри-

мер, «vm4.ru/vsphere/esxi». Тогда учетные записи серверов будут созданы в контейнере vsphere/esxi.

По умолчанию права администратора на ESXi имеют участники доменной группы «ESX Admins».

Если вариант по умолчанию вас не устраивает, то в контекстном меню сервера выберите пункт **Add Permissions** – и у вас будет возможность выдать произвольные привилегии произвольному доменному пользователю или группе.

Или вы можете пройти **Configuration ⇒ Advanced Settings ⇒ HostAgent ⇒ Config.HostAgent.plugins.hostsvc.esxAdminsGroup** и поменять группу администраторов по умолчанию (ESX Admins) на произвольную.

Обратите внимание на то, что данную настройку можно тиражировать между серверами при помощи механизма **Host Profiles**.

Использовать доменного пользователя с таким образом назначенными привилегиями можно при обращении на этот сервер ESXi при помощи клиента vSphere или по SSH.

Вариант 2 – вам не требуется работать напрямую с ESX

Самый простой вариант – вам не требуется работать напрямую и не требуется жестко запрещать это. Что ж, просто не подключайтесь и коллегам запретите.

Вариант 3 – вам требуется жестко запретить работу напрямую

А вот в ситуации, когда политики безопасности или собственные желания диктуют вам жестко запретить подключения напрямую к хостам, минуя vCenter, – как это сделать, не совсем понятно, на первый взгляд.

Частично может помочь брандмауэр, в пятой версии ESXi его штатная функция – явно ограничивать IP-подсети и IP-адреса, с которых можно подключаться.

Но есть возможность получить требуемое проще и даже в чем-то надежнее. Этот способ – настройка **Lockdown mode**. Пройдите **Configuration ⇒ Security Profile ⇒ Lockdown mode**. Если поставить флажок, то ESXi не позволит авторизоваться никаким пользователем, кроме пользователя vpxuser. vpxuser – это специальный пользователь, которого создает vCenter в момент добавления в него ESXi. Его пароль генерируется автоматически и неизвестен никому. Таким образом vCenter управляет сервером ESXi как обычно, а вот напрямую к этому серверу подключиться не получится никаким образом – потому что запрос на авторизацию будет заведомо отвергнут.

Здесь пытливый читатель насторожится – а что, если vCenter будет недоступен? Представим себе ситуацию, что администратор ошибочно выключил ВМ с vCenter. Как ее включить?

Соображений два. Во-первых – да, это проблема. Но если речь идет про инфраструктуру с повышенными требованиями к безопасности – удобством за это приходится платить. Ну и в такой инфраструктуре сервер управления уровня vCenter не должен быть подвергнут такого рода случайностям.

Во-вторых, хорошая новость. Если мы обратимся на локальную консоль ESXi (при помощи KVM/iLO/ножками к серверу), то в локальное БИОС-подобное

меню нас пустят. В нем мы обнаружим пункт **Configure Lockdown Mode**, который позволит возобновить доступ к серверу по сети, например клиентом vSphere.

Однако в инфраструктурах с максимально жесткими требованиями по безопасности это локальное БИОС-подобное меню также могут потребовать отключить. Это можно сделать в клиенте vSphere: **Configuration** ⇒ **Security Profile** ⇒ **Properties** в секции **Services** ⇒ **dcui**. DCUI = Direct Console User Interface, это официальное название упоминаемого мною БИОС-подобного меню. Кстати, его можно запустить в ssh-сессии командой `dcui`.

Так вот, если DCUI выключен, а Lockdown-режим включен, то vCenter – это абсолютно единственный способ управлять таким сервером ESXi. Если vCenter отказал без возможности восстановления, то единственный способ вернуть контроль над ESXi – это его переустановка.

4.2.4. Контроль доступа, раздача прав при работе через vCenter

vCenter предлагает вам достаточно гибкую систему раздачи прав. Что она собой представляет, см. рис. 4.4.

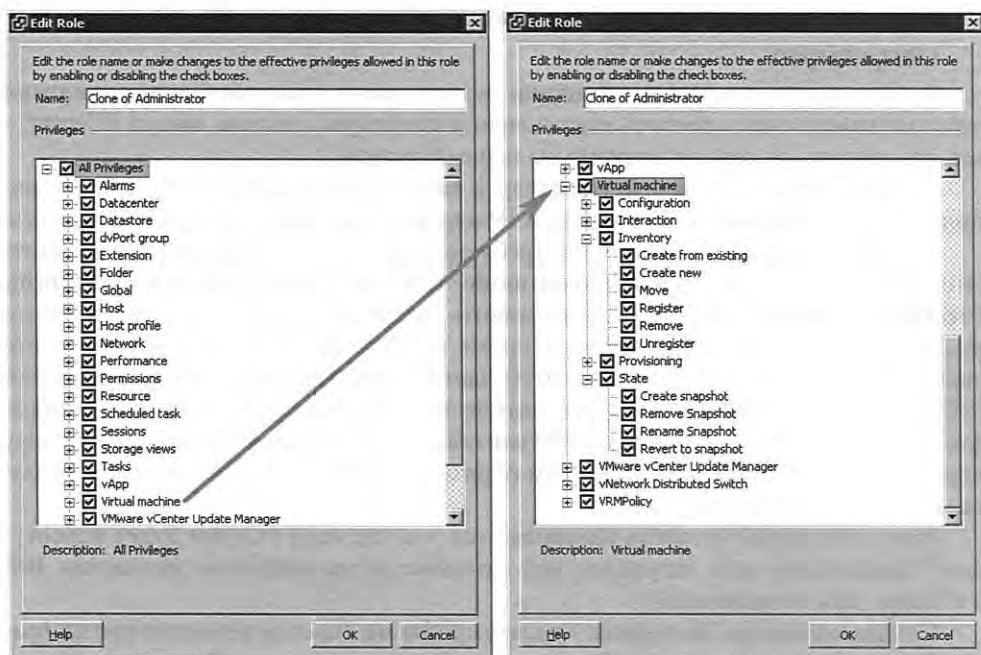


Рис. 4.4. Настройки привилегий на vCenter

Здесь вы видите список привилегий (privileges, прав), включенных в роль (role) Administrator.

Привилегия – это право на атомарное действие. На рисунке справа вы видите привилегии для виртуальных машин, такие как «Создать», «Удалить», «Создать снимок состояния (snapshot)» и др. Набор каких-то привилегий называется ролью.

Роль – это конкретный набор привилегий, то есть разрешенных действий. Роль можно дать пользователю или группе на какой-то уровень иерархии vCenter – см. рис. 4.5.

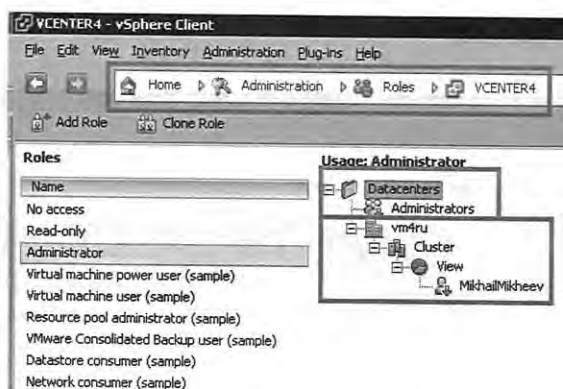


Рис. 4.5. Информация о том, на какой уровень иерархии и какому пользователю назначена выбранная роль

Здесь вы видите, что роль Administrator (слева) дана группе Administrators на уровне объекта Datacenters (самом верхнем уровне иерархии vCenter). Это, кстати, настройки по умолчанию – группа локальных администраторов на сервере vCenter имеет все права на всю иерархию.

Кроме того, здесь эта роль выдана пользователю MikhailMikheev на пул ресурсов под названием «View».

У vCenter нет собственной БД пользователей, он пользуется:

- ☐ локальными пользователями и группами Windows, созданными на сервере, на котором установлен vCenter (или пользователями Linux в случае vCSA);
- ☐ доменными пользователями и группами того домена, в который входит сервер vCenter (если он в домен входит).

Таким образом, порядок действий для выдачи каких-то прав пользователю или группе следующий:

1. Создаем этого пользователя/группу, если они еще не существуют. Они могут быть локальными в Windows/Linux, где установлен vCenter, или доменными, если он входит в домен.
2. Создаем роль. Для этого проходим **Home** ⇒ **Administration** ⇒ **Roles**. Затем:
 - в контекстном меню пустого места выбираем пункт **Add** для создания новой роли с нуля;
 - в контекстном меню существующей роли выбираем пункт **Clone** для создания копии существующей роли. Если хотим поменять созданную роль, вызываем для нее контекстное меню и выбираем **Edit**.

В любом случае флажками отмечаем все необходимые привилегии.

3. Идем **Home** ⇒ **Inventory** и выбираем:

- Hosts and Clusters для раздачи прав на видимые в этой иерархии объекты – кластеры, сервера, каталоги с серверами, пулы ресурсов и др.;
- VMs and Templates – на ВМ, шаблоны и каталоги с этими объектами;
- Datastore – для раздачи прав на хранилища;
- Networking – для раздачи прав на вКоммутаторы.

Посмотрите на рис. 4.6.

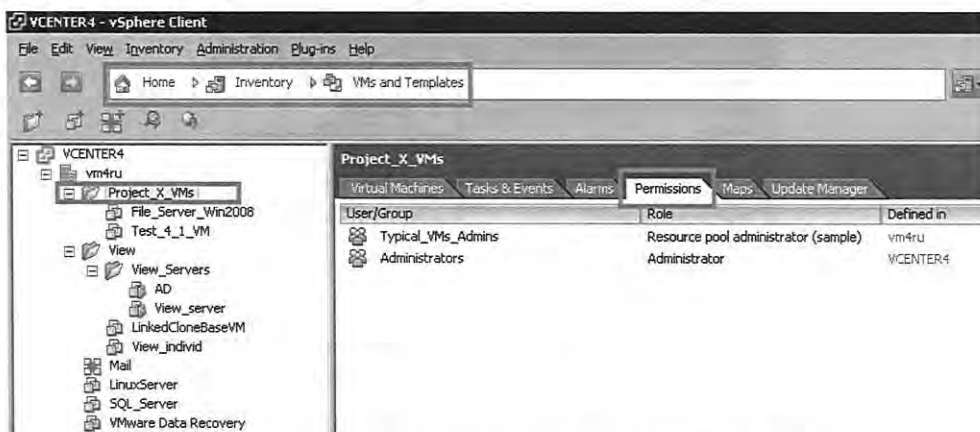


Рис. 4.6. Просмотр разрешений для выбранного объекта

Здесь вы видите каталоги для ВМ (они голубого цвета и видны только в режиме «VMs and Templates»). Если перейти на вкладку **Permissions**, то мы увидим информацию о том, кто и какие права имеет сейчас на выбранный объект.

В данном примере мы видим, что группа **Administrators** имеет права роли **Administrator**, притом эта роль назначена группе на уровне «vcenter4» (то есть в корне иерархии vCenter, у меня vcenter4 – это имя машины с установленным vCenter).

Кроме того, группе «**Typical_VMs_Admins**» назначена роль «**Resource pool administrator (sample)**» на уровне «vm4ru» (в данном примере это название объекта Datacenter, содержащего все сервера и ВМ).

Обратите внимание. Если вы вернетесь к рис. 4.5, то увидите, как посмотреть обратную связь – какая роль кому и на какой объект назначена.

Теперь вы хотите дать некие права группе или пользователю на объект в иерархии. Оставаясь на вкладке **Permissions** этого объекта, вызываем контекстное меню и выбираем **Add Permissions** (рис. 4.7).

В открывшемся окне нажатием кнопки **Add** вы можете выбрать пользователей и группы, затем в правой части из выпадающего меню выбрать роль, которую хо-

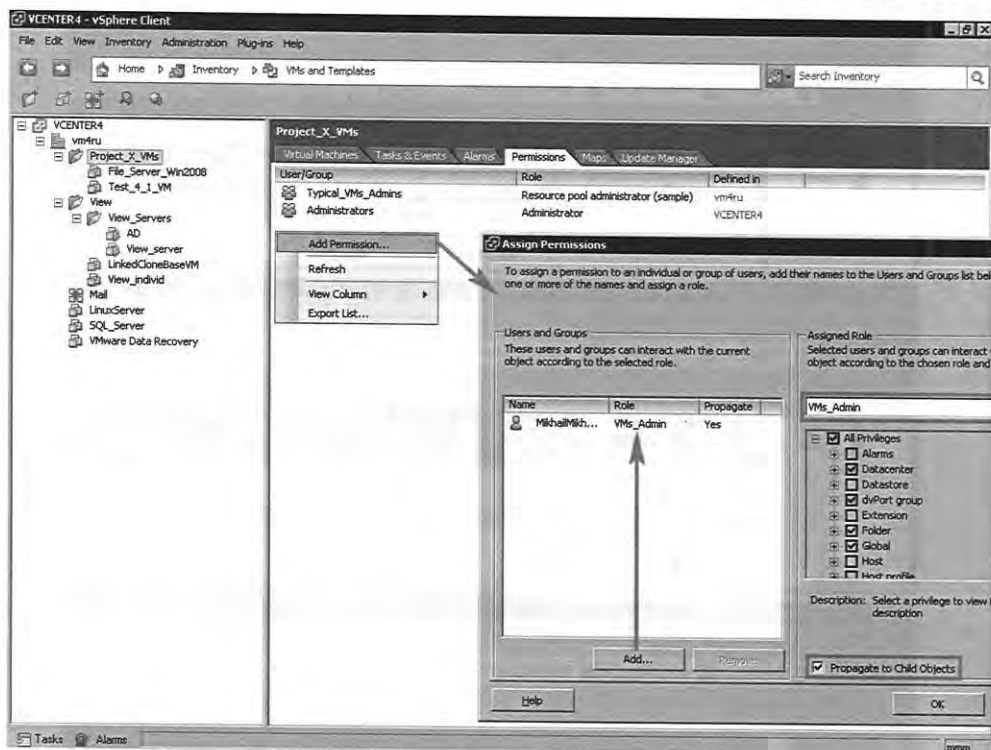


Рис. 4.7. Назначение роли на каталог с VM

тите им дать. Желаемая роль к этому моменту должна быть создана, делается это в меню **Home** ⇒ **Administration** ⇒ **Roles**.

Обратите внимание на флажок **Propagate to Child Objects** (Применять к дочерним объектам). Если он не стоит, то права выдаются только на объект (каталог Project_X_VMs в моем примере), но не на ветвь его подобъектов.

В своих примерах я дал пользователю «MikhailMikheev» (созданному мною в Windows, на которой установлен vCenter) роль VMs_Admin (которую я сам создал в vCenter) на каталог Project_X_VMs. Если теперь обратиться клиентом vSphere от имени этого пользователя, то увидим следующую картину – см. рис. 4.8.

Пользователь не видит других объектов, кроме тех, на которые имеет права, – то есть двух VM. Если он просматривает списки событий (**Events**), то ему доступны события только этих двух объектов. Ему недоступно назначение ролей, недоступно управление лицензиями, и далее, и далее.

Далее немного правил применения прав.

Самое важное: если пользователю выданы разные права на разных уровнях иерархии vCenter, то результирующими для какого-то объекта являются первые, встреченные снизу вверх, – см. рис. 4.9.

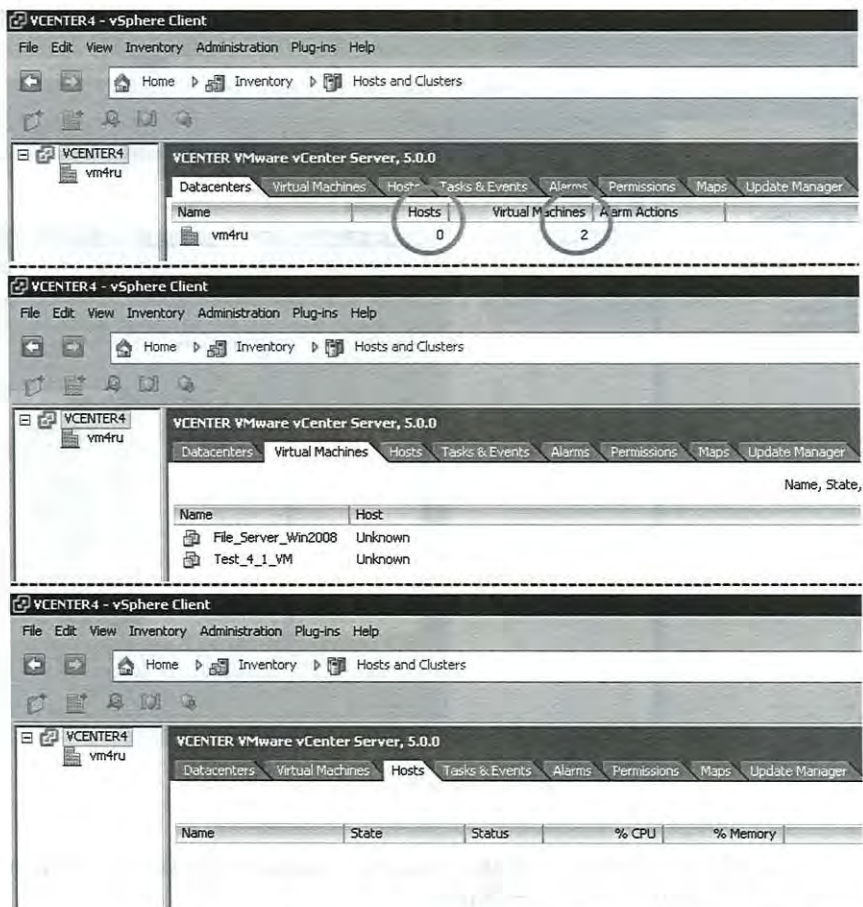


Рис. 4.8. Подключение от имени непривилегированного пользователя

Это достаточно типичная ситуация: пользователь VasilyPurkin имеет роль администратора (то есть все права) на всю иерархию. На пул ресурсов nonCritical_Production_VMs выданы ограниченные права группе пользователей, в которую входит и VasilyPurkin. По правилам распространения привилегий vCenter, для этого пула и для входящих в него ВМ пользователь не обладает правами администратора, только правами на чтение.

Почему я назвал ситуацию типичной: потому что не исключено, что администратор будет давать каким-то группам пользователей ограниченные права на группы ВМ (или сетей, или хранилищ. Впрочем, ВМ более вероятны). И бывает, что администратор сам входит, или начинает входить через какое-то время, в эту самую группу. И вследствие этого теряет административные привилегии на ветвь иерархии.

Другой пример – на рис. 4.10.

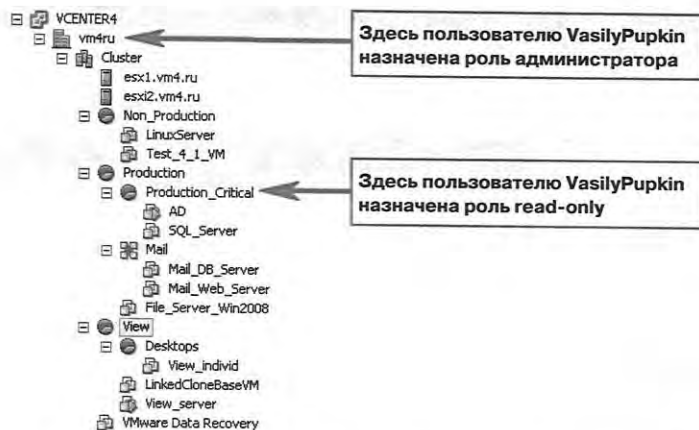


Рис. 4.9. Иллюстрация варианта назначения разных прав на разные уровни иерархии

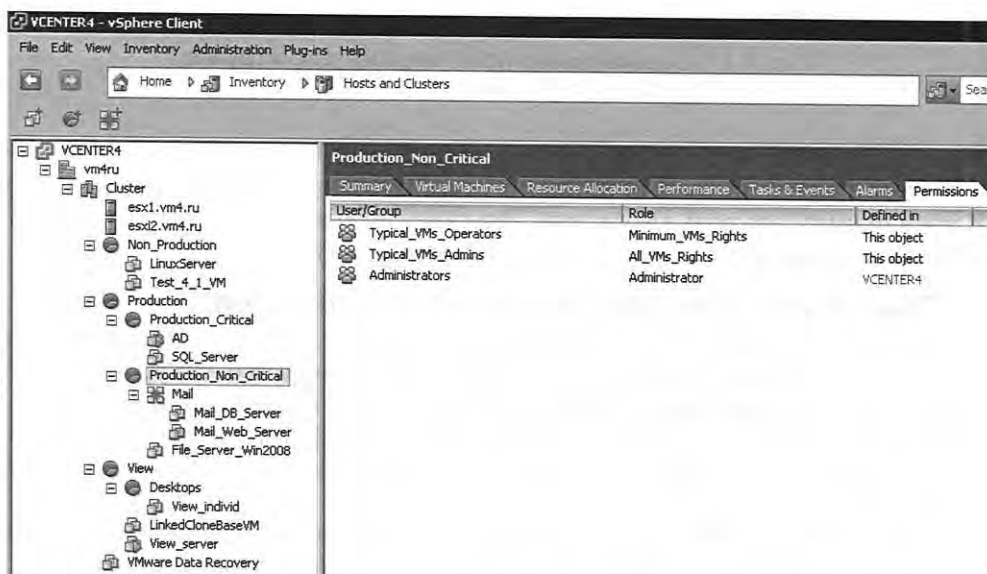


Рис. 4.10. Иллюстрация выдачи разных прав двум группам на один объект иерархии

Здесь вы видите, что на один и тот же объект в иерархии выданы разные права двум группам. Какие права для нас будут действовать в случае, если мы входим в обе группы? В данном случае происходит объединение привилегий – мы будем обладать теми, что входят хотя бы в одну роль.

Последний пример – на рис. 4.11.

Здесь на один объект в иерархии выданы права и непосредственно пользова-

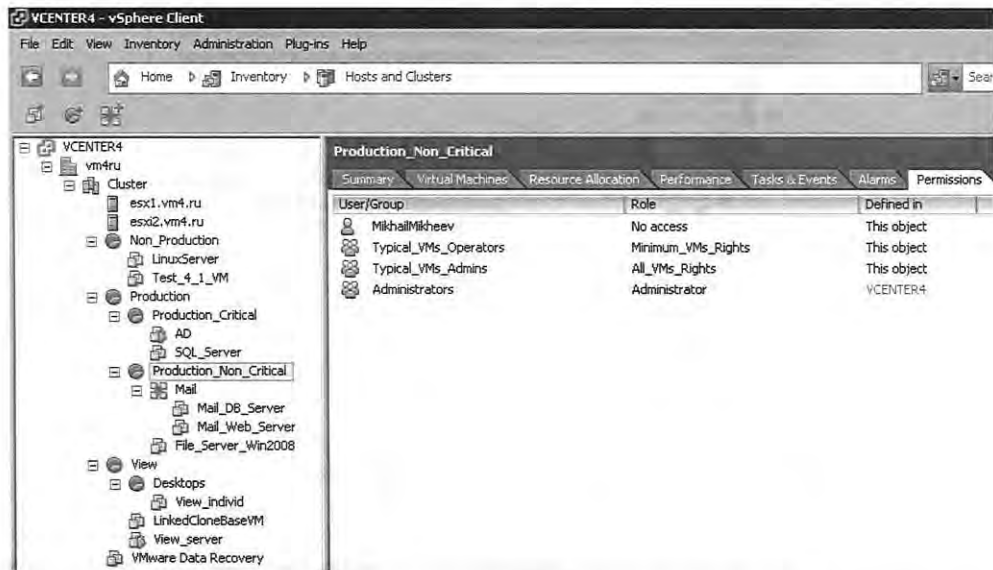


Рис. 4.11. Выдача разных прав пользователю и группе, в которую он входит, на один объект

телю MikhailMikheev, и группам, в которые он входит. В данном случае результирующими являются только те права, что выданы непосредственно пользователю. В моем примере это роль «No access». Эта существующая по умолчанию роль необходима, как вы понимаете, для явного лишения доступа.

Общие соображения по разграничению прав доступа

Если в компании один-два-три администратора, то, скорее всего, вам система раздачи прав не пригодится или пригодится очень ограниченно – на уровне предоставления прав на некоторые ВМ некоторым группам пользователей.

А вот если с разными областями виртуальной инфраструктуры должны работать разные группы администраторов, операторов и пользователей, то имеет смысл сделать примерно так:

1. Составьте перечень повседневных задач, которые придется решать. Если что-то забудете – ничего страшного, выполните несколько итераций.
2. Перечислите, какие объекты иерархии и элементы интерфейса используются для выполнения каждой задачи из списка, составленного выше. Про каждую, и подробно.
3. Распишите, кто и где будет это выполнять.

Полученный в результате документ называется «матрицей доступа к информации». Кто, куда, зачем. Фактически это – основа внутренней документации по безопасности виртуальной инфраструктуры, на ее основе будут создаваться роли и выдаваться на те или иные уровни иерархии, с ее помощью будут фиксироваться изменения. Наличие документации, куда вносятся изменения в конфигурации,

является обязательным – иначе вы рискуете в какой-то момент запутаться вплоть до потери доступа к инфраструктуре.

Не используйте роли, существующие в vCenter по умолчанию. За исключением роли **Administrator** (то есть все права), **Read-only** (просмотр информации об объекте) и **No Access** (явное отсутствие доступа). Прочие существующие по умолчанию роли использовать не рекомендуется (кроме роли **VMware Consolidated Backup user (sample)**).

Само собой, правильным будет создание ролей под конкретные нужды, а не использование одной роли, которая может все. Создавайте роли под конкретные задачи с минимально необходимым набором прав. Существующие по умолчанию роли могут не соответствовать этому условию.

Однозначно роли должны назначаться персонифицированно, то есть не должно быть учетной записи, из-под которой могут аутентифицироваться несколько человек. Это чрезвычайно помогает, в частности если необходимо восстановить последовательность событий и виновное лицо.

Не используйте локальных учетных записей, кроме исключительных случаев. Только доменные. Тем более это удобнее – для аутентификации в vSphere можно использовать ту учетную запись, от имени которой был выполнен вход в Windows, и не набирать пароль заново.

Не забывайте о настройках по умолчанию: локальная группа администраторов имеет все права на корень иерархии vCenter. Конечно же правильно будет:

1. Создать персонифицированную учетную запись (даже две).
2. Наделить их правами администратора на все в vCenter.
3. Убрать их данные в сейф, пользоваться ими лишь в исключительных случаях.
4. Лишить группу локальных администраторов прав в vCenter.

4.3. Настройка сертификатов SSL

VMware vSphere, а именно продукты ESXi 5, vCenter 5 и VMware Update Manager 5, поддерживают SSL v3 и TLS v1 (обычно употребляется просто «SSL»). Если SSL включен, то трафик между узлами виртуальной инфраструктуры зашифрован, подписан и не может быть незаметно изменен. ESXi, как и другие продукты VMware, использует сертификаты X.509 для шифрования передаваемого по SSL трафика.

В vSphere 5 проверка сертификатов включена по умолчанию, и они используются для шифрования трафика. Однако по умолчанию эти сертификаты генерируются автоматически во время установки ESXi. Данная процедура не требует от администратора каких-то действий. Но они не выданы центром сертификации (certificate authority, CA. Также иногда упоминается в русскоязычной документации как «удостоверяющий центр», УЦ). Такие самоподписанные сертификаты потенциально уязвимы для атак «человек в середине». Потому что для них до того, как начинается шифрование трафика, не производится проверка подлинности самих сертификатов.

При попытке подключения к ESXi или vCenter (с помощью клиента vSphere или браузера) пользователю выдается соответствующее предупреждение (рис. 4.12), сообщающее ему о том, что удаленная система может не являться доверенной и установить ее подлинность не представляется возможным. Конечно, это предупреждение можно отклонить, можно занести сертификаты всех управляемых систем в список доверенных, но это может ослабить безопасность инфраструктуры. К тому же, возможно, в вашей организации запрещено использование систем с недоверенными сертификатами (административно или технически).

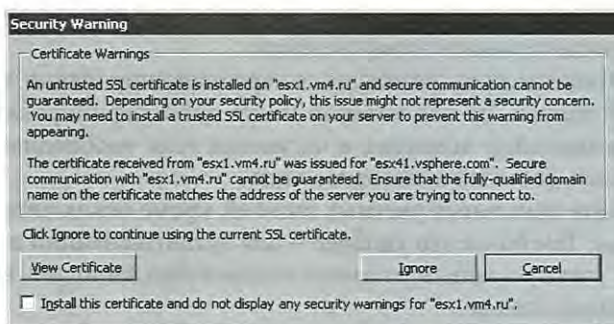


Рис. 4.12. Предупреждение о недоверенном сертификате

Для решения этих проблем вам потребуется запросить у доверенного центра сертификации подходящий сертификат и заменить им сгенерированные автоматически. Доверенный центр сертификации может быть коммерческим (например, VeriSign, Thawte или GeoTrust) или установленным в вашей сети (например, Microsoft Windows Server Active Directory Certificate Services, AD CS или OpenSSL).

Примерный план этого действия выглядит следующим образом.

1. Получение сертификата для ESXi, замена ими сгенерированных по умолчанию сертификатов.
2. Получение сертификатов для vCenter и Update Manager. Замена сгенерированных по умолчанию сертификатов.

Здесь не приводятся конкретные инструкции, так как они сильно зависят от вашей инфраструктуры, да и далеко не всем администраторам vSphere придется заниматься этим вопросом. Однако если для вас данная проблема актуальна, обратитесь к документации. Подборка ссылок доступна тут: <http://link.vm4.ru/ssl>.

4.4. Host Profiles

Механизм Host Profiles, появившийся в четвертой версии виртуальной инфраструктуры VMware, предоставляет возможность осуществлять настройку серверов ESXi шаблонами. В пятой версии vSphere было увеличено количество на-

строек, тиражируемое при помощи этого механизма. В первую очередь это было вызвано появлением Auto Deploy – механизма PXE загрузки серверов ESXi. При его использовании ESXi стартует по сети с типового образа, и уникальные настройки для свежезагруженного сервера применяются именно при помощи Host Profiles. Именно для этой функции к Host Profiles была добавлена функция файлов ответа – у нас есть типовый профиль настроек для многих серверов и для каждого сервера набор уникальных параметров в рамках этого профиля (в первую очередь это IP-адреса для интерфейсов VMkernel).

Что это за настройки:

- ☐ **Storage Configuration** – настройки подключения хранилищ NFS, программного инициатора iSCSI, программного контроллера FCoE, модулей multipathing;
- ☐ **Networking** – настройки сети. Здесь мы можем указать, какие коммутаторы, распределенные коммутаторы, группы портов и интерфейсы VMkernel надо создавать. С какими настройками IP, VLAN, security, NIC teaming. То есть даже сложную и разветвленную конфигурацию сети мы можем создать один раз, а затем через профиль настроек перенести на другие сервера;
- ☐ **Date and Time** – сервера NTP;
- ☐ **Firewall** – настройки брандмауэра;
- ☐ **Security** – пароль root;
- ☐ **Service** – настройка служб, таких как ssh, ntp и др. Настройки на уровне «должны ли они запускаться при старте сервера»;
- ☐ **Advanced** – некоторые из расширенных настроек;
- ☐ порядок именования PCI-устройств (если нам важно, например, какая именно сетевая карта имеет имя vmnic1);
- ☐ настройки SNMP;
- ☐ интеграция с AD;
- ☐ и др.

Последовательность действий для работы с этим механизмом следующая.

1. У вас должен быть полностью настроенный сервер, с которого шаблон и снимается.
2. Шаблон назначается на другой сервер, кластер или датацентр.
3. Выполняется проверка соответствия настроек серверов шаблону (host's compliance).
4. Настройки из шаблона применяются к серверам с отличающимися настройками.

Для создания шаблона проще всего перейти **Home** ⇒ **Management** ⇒ **Host Profiles** (рис. 4.13).

Запустится мастер, где вы сначала выберете, хотите ли импортировать из файла шаблон (например, это может быть резервная копия, понадобившаяся вам после переустановки vCenter) или создать новый (рис. 4.14).

Вам предложат выбрать сервер, настройки которого будут использоваться в шаблоне. Затем вы укажете имя и описание шаблона. После завершения работы мастера шаблон настроек появится в списке (рис. 4.15).

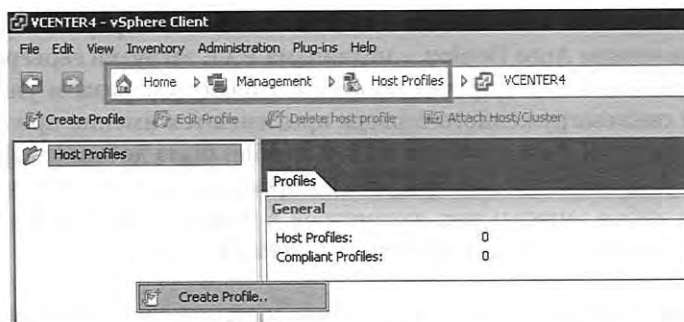


Рис. 4.13. Создание профиля настроек

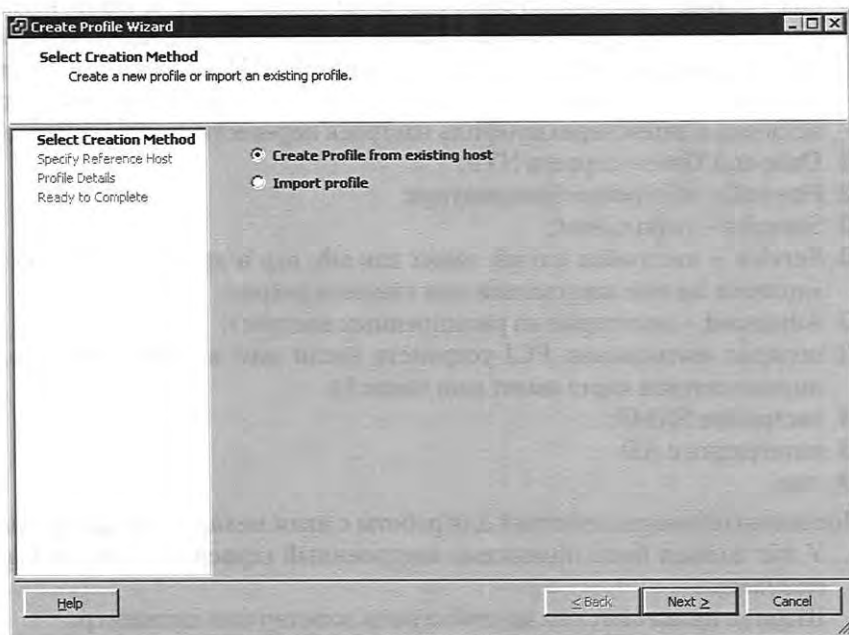


Рис. 4.14. Мастер создания профиля настроек

На рисунке выделено контекстное меню профиля настроек. Как вы видите, прямо отсюда можно:

- ☐ изменять содержащиеся в профиле настройки;
- ☐ удалить этот профиль;
- ☐ указать сервера и кластеры, на которые этот шаблон назначен;
- ☐ сменить эталонный сервер;
- ☐ обновить профиль – то есть заново считать настройки с эталонного сервера и занести их в профиль;
- ☐ экспортировать этот профиль настроек в файл.

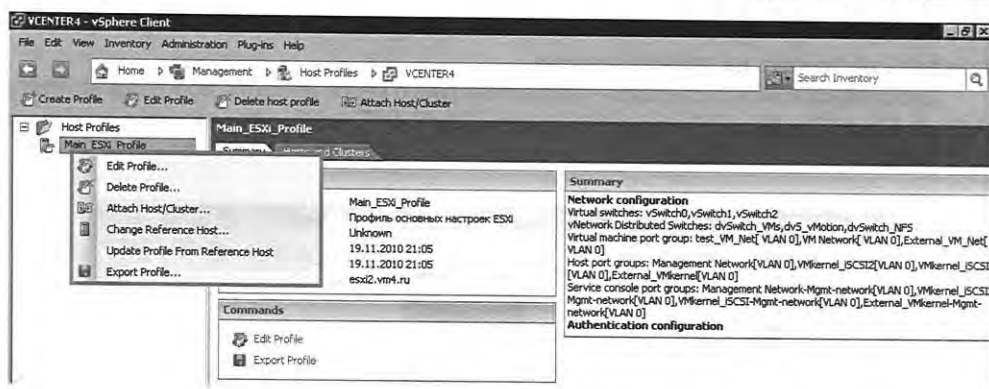


Рис. 4.15. Доступные операции с профилем настроек

Вполне вероятно, что сразу после создания шаблона настроек вам захочется его отредактировать перед назначением на другие сервера. Нажмите **Edit Profile**. Вы увидите список настроек, которыми можно манипулировать через механизм редактирования профилей (рис. 4.16).

Итак, вы создали профиль настроек сервера. Вы изменили его – например, удалив из профиля создание каких-то элементов виртуальной сети, которые были

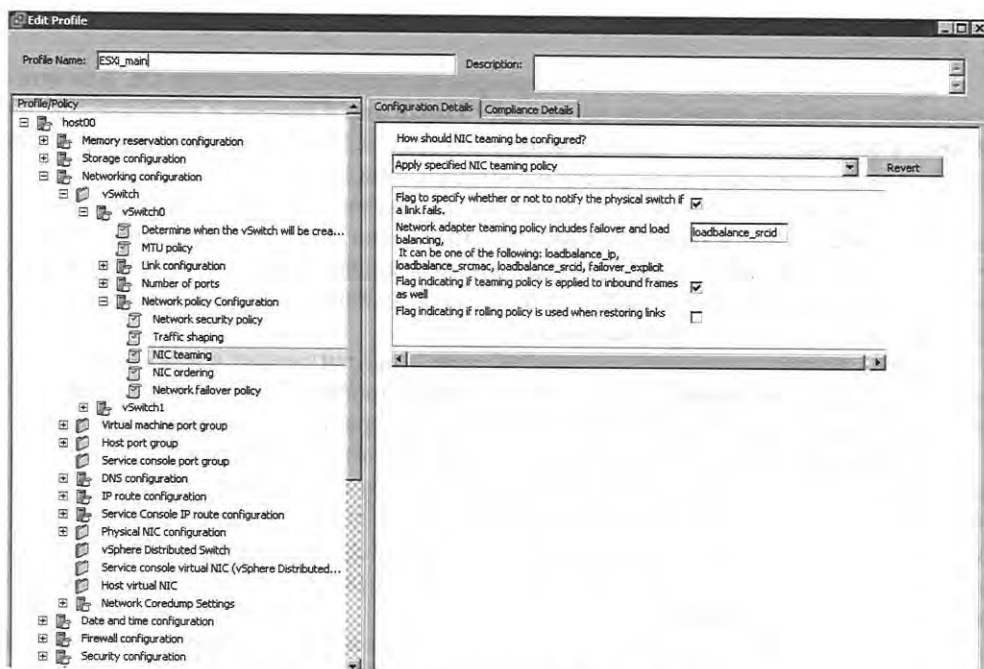


Рис. 4.16. Редактирование профиля настроек

на эталонном сервере, но которые не нужны на прочих серверах. Теперь надо этот профиль назначить на прочие сервера.

Простой способ это сделать – из контекстного меню данного профиля выбрать **Attach Host/Cluster**. В появившемся окне (рис. 4.17) выберите нужные сервера и кластеры и нажмите кнопку **Attach**.

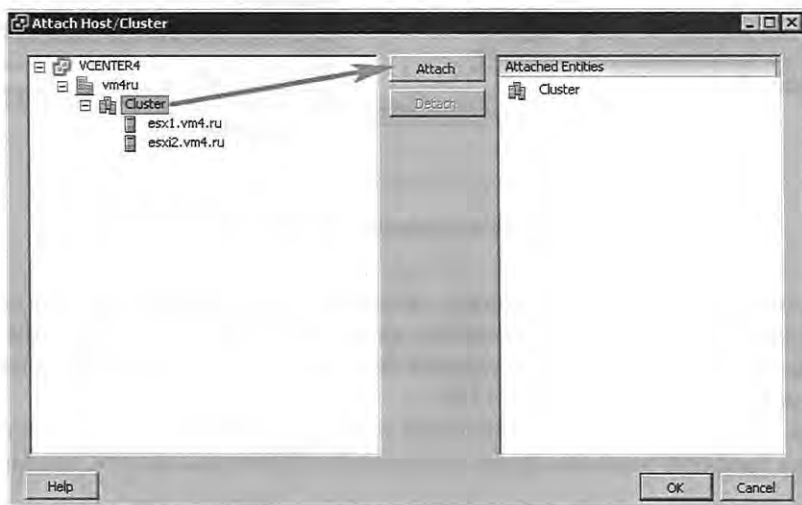


Рис. 4.17. Назначение профиля на сервер или кластер

Предпоследний шаг – проверка серверов на соответствие профилю. Для этого выделите профиль, перейдите на вкладку **Hosts and Clusters** и нажмите ссылку **Check Compliance Now** в правой части окна (рис. 4.18). Через короткое время на этой вкладке отобразится ситуация с соответствием настроек.

Здесь вы видите, что профиль назначен на кластер и два сервера (сервера, очевидно, принадлежат кластеру). **Cluster** отмечен как **Noncompliant**, это означает, что в кластере хотя бы один сервер не удовлетворяет профилю настроек. Сразу видим, что это сервер `esxi1.vm4.ru`, и, выделив его, в нижней части видим расхождение его настроек с настройками из профиля.

Теперь можно выполнить последний шаг – ввести сервер в режим обслуживания (**maintenance mode**) и применить (**Apply**) профиль настроек. Оба действия можно выполнить из контекстного меню для сервера на том же окне (рис. 4.19).

Maintenance Mode. Напомню, что сервер может войти в режим обслуживания только тогда, когда на нем не остается ни одной работающей ВМ – все они мигрированы или выключены. Также вас спросят, хотите ли вы переместить выключенные и приостановленные (**suspend**) ВМ на другие сервера, – это полезно для подстраховки на случай, если сервер потеряет работоспособность после применения профиля настроек.

Итак, перевели сервер в режим обслуживания и нажали **Apply** для профиля настроек. Откроется мастер, который спросит о значениях уникальных настроек, таких как IP-адреса для интерфейсов VMkernel и подобных (рис. 4.20).

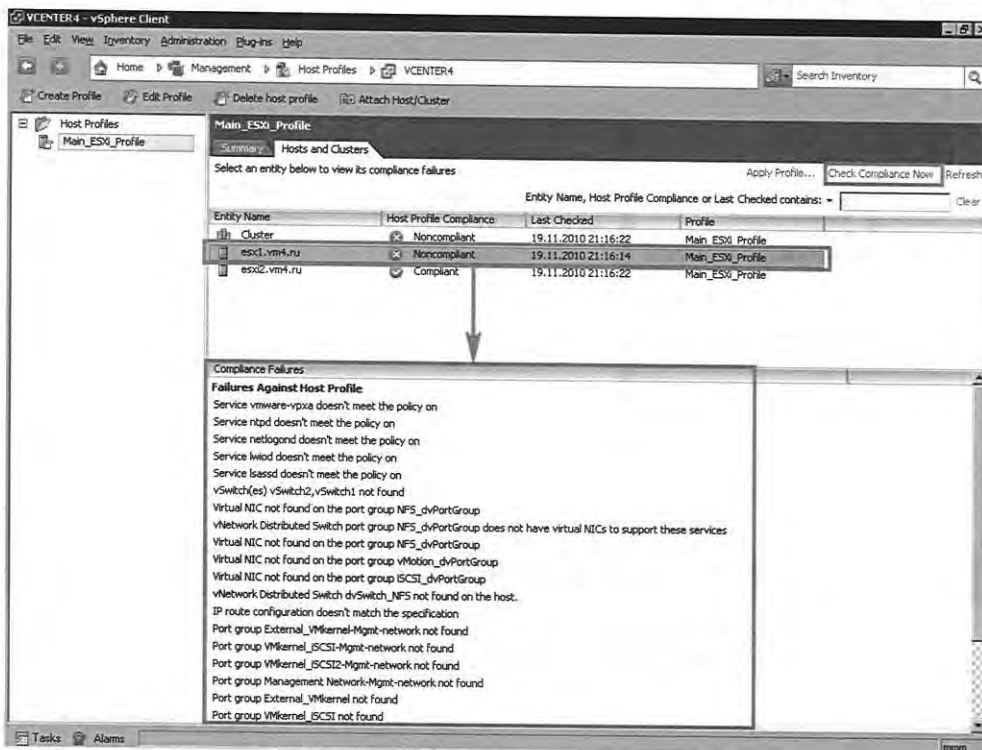


Рис. 4.18. Проверка на соответствие серверов назначенному профилю настроек

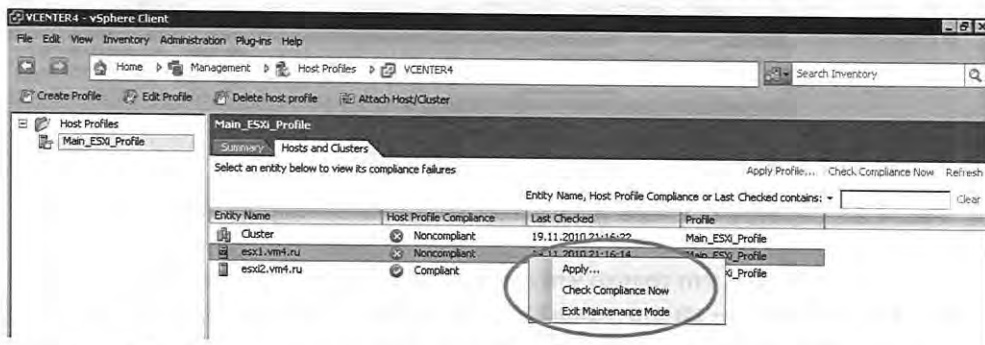


Рис. 4.19. Приведение настроек сервера в соответствие профилю настроек

В нижней части указано, сколько таких настроек нам нужно будет указывать. Здесь – одну.

После завершения мастера и выполнения операции настройки вы должны увидеть примерно такую картину, как на рис. 4.21.

Здесь вы видите, что все сервера соответствуют (**Compliant**) профилю настроек (иногда необходимо заново запустить проверку соответствия (**Compliance**)).

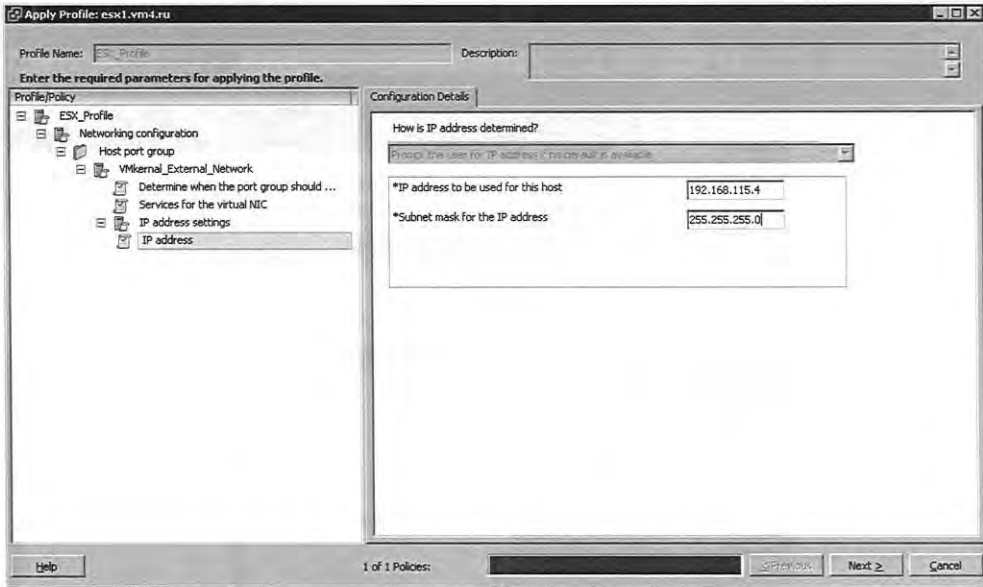


Рис. 4.20. Запрос о значении уникальных настроек при применении профиля настроек к серверу

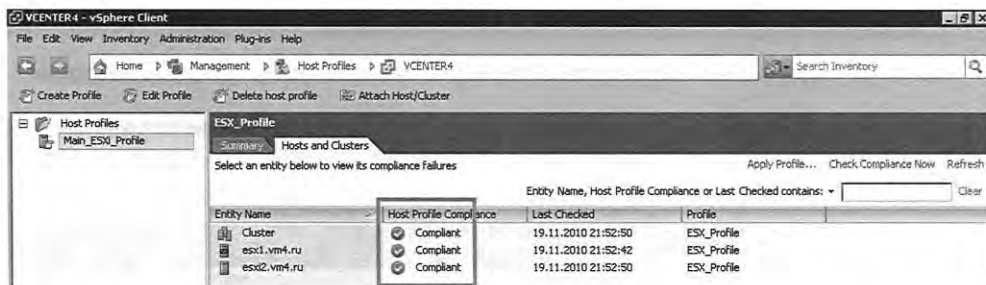


Рис. 4.21. Серверы кластера удовлетворяют назначенному профилю настроек

Check)). Не забудьте, что только что настроенный сервер все еще находится в режиме обслуживания – на это указывает его иконка. Пока он не выйдет из этого режима, на нем нельзя запускать VM. Так что вызовите для него контекстное меню и выберите пункт **Exit Maintenance Mode**.

Все.

Hosts Profiles удобно применять:

- ☐ для первоначальной настройки инфраструктуры. Установили ферму ESXi, один из них настроили, сняли шаблон – с его помощью настроили остальные;
- ☐ для добавления сервера в инфраструктуру. Установили на него ESXi, назначили профиль – новый сервер настроен;

- для автоматической проверки корректности настроек. При создании профиля настроек автоматически создается задача в планировщике (**Home** ⇒ **Management** ⇒ **Scheduled tasks**), которая ежедневно запускает проверку соответствия каждому профилю настроек для каждого из серверов, на который он назначен. Таким образом, если на одном из серверов по ошибке или случайно изменилась настройка (из управляемых профилями) – вы легко сможете это отследить. К сожалению, нет возможности настроить автоматическое оповещение по электронной почте или SNMP. Для просмотра текущей ситуации вам необходимо будет зайти на вкладку **Hosts and Clusters** нужного профиля настроек;
- наконец, для резервного копирования. К примеру, в силу каких-то причин вы приняли решение переустановить ESXi на каком-то из серверов. Вы сохраняете его текущие настройки в профиль и применяете этот профиль к свежеустановленному ESXi на этом же сервере.

Ну и я уже упоминал про Auto Deploy – работа этого сервера PXE-загрузки невозможна без Host Profiles.

Кстати, работать с профилями настроек можно не только из раздела **Home** ⇒ **Management** ⇒ **Host Profiles**. Пройдите в **Home** ⇒ **Inventory** ⇒ **Hosts and Clusters**, выделите сервер, кластер или датацентр и перейдите на вкладку **Profile Compliance** (рис. 4.22).

Здесь или в контекстном меню кластера и серверов можно назначать профили настроек, убирать назначения, запускать проверку на соответствие, запускать приведение к соответствию.

Профили настроек не реплицируются между серверами vCenter в режиме Linked Mode.

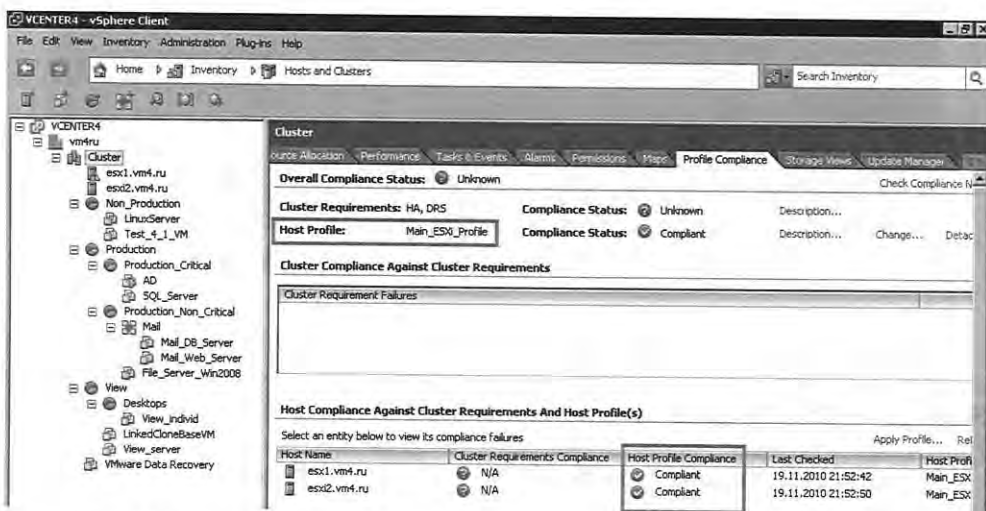


Рис. 4.22. Работа с профилями настроек из раздела **Hosts and Clusters**

4.5. Использование SNMP

Для мониторинга, происходящего с инфраструктурой, полезной или незаменимой может быть система мониторинга по SNMP. Использовать ее с vSphere можно и нужно, и даже немного разными способами.

Программа-минимум – настроить оповещение по SNMP от vCenter, по факту срабатывания важных нам alarm.

Кроме того, есть возможность активировать агента SNMP на серверах ESXi.

4.5.1. Настройка SNMP для vCenter

Выберите alarm, который отслеживает интересующее вас условие или несколько. На вкладке **Action** укажите оповещение по SNMP – **Send a notification trap**. Для этого, добавив реакцию кнопкой **Add**, вызовите выпадающее меню в столбце **Action** для появившейся строки (рис. 4.23).

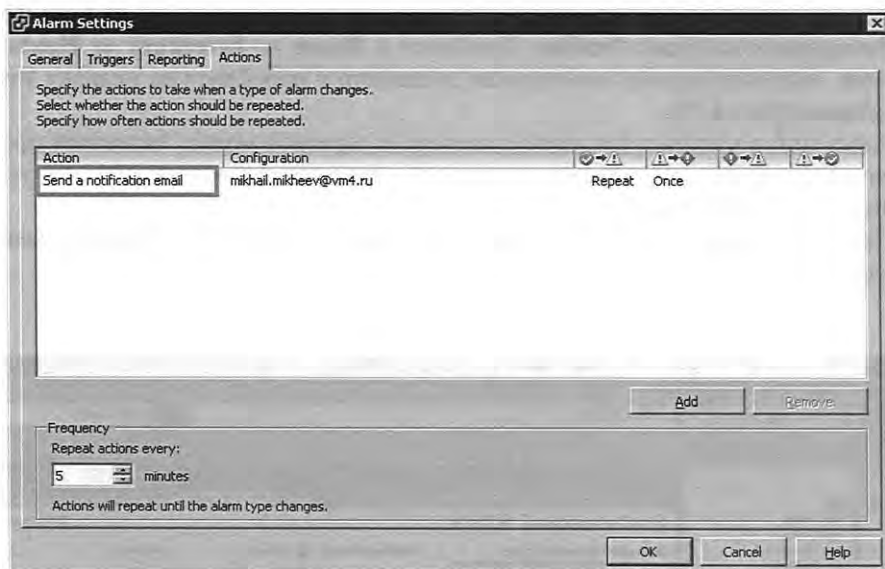


Рис. 4.23. Настройка оповещения по SNMP для alarm

Все эти оповещения будут отсылаться сервером vCenter, поэтому необходимо сделать настройки SNMP для него. Для этого пройдите **Home** ⇒ **vCenter Server Settings** ⇒ **SNMP**. Здесь вы можете указать получателей и строки community (рис. 4.24).

В результате – подобное оповещение, что ВМ потребляет аж больше 77% памяти (рис. 4.25).

Или что сервер отвалился от vCenter (рис. 4.26).

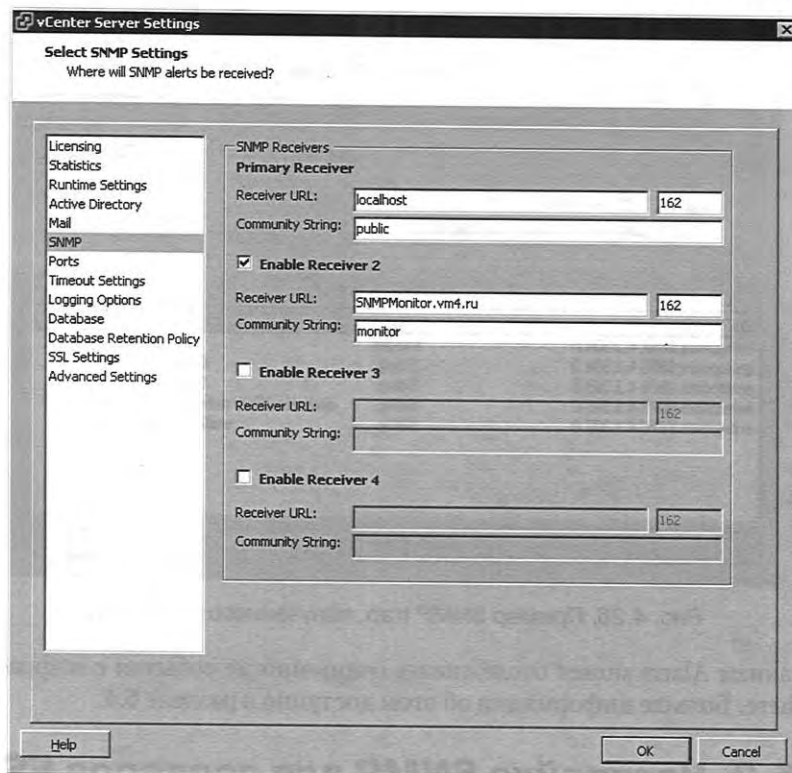


Рис. 4.24. Настройки получателей SNMP для vCenter

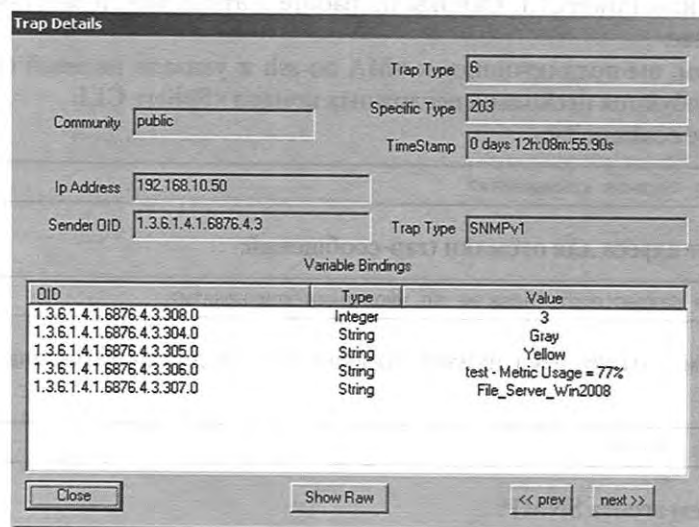


Рис. 4.25. Пример SNMP trap, полученного от vCenter

OID	Type	Value
enterprises.6876.4.3.308.0	Integer	2
enterprises.6876.4.3.304.0	String	Gray
enterprises.6876.4.3.305.0	String	Gray
enterprises.6876.4.3.306.0	String	alarm.HostConnectivityAlarm - Event: Hos...
enterprises.6876.4.3.307.0	String	esxi2.vm4.ru

Рис. 4.26. Пример SNMP trap, полученного от vCenter

Механизм Alarm может отслеживать очень многие события с инфраструктурой vSphere. Больше информации об этом доступно в разделе 6.4.

4.5.2. Настройка SNMP для серверов ESXi

Настройку SNMP для серверов ESXi правильнее всего осуществлять при помощи vMA или PowerCLI. Основы по работе с этими инструментами я описал в первой главе.

Допустим, вы подключились к vMA по ssh и указали целевой сервер ESXi. Теперь потребуются несколько несложных команд vSphere CLI:

Указание community:

```
vicfg-snmp -c <нужное коммьюнити>
```

Указание адреса для отсылки trap-сообщений:

```
vicfg-snmp -e <адрес:порт, если не по умолчанию/коммьюнити>
```

Если команды get и set нужны, то укажем порт для прослушивания агентом SNMP:

```
vicfg-snmp -p <порт>
```

Включаем агента SNMP:

```
vicfg-snmp --enable
```


Пробуем отправить тестовый trap:

```
vicfg-snmp --test
```

Если все сделано правильно, то тестовый trap мы увидим на системе мониторинга.

Там же будут отображаться все события, которые отслеживает SNMP-агент на серверах ESXi. Например, включение виртуальной машины или проблемы с аппаратной частью.

Если хочется настроить snmp при помощи PowerCLI, то пригодится примерно следующий код.

Получим список всех серверов esxi. К сожалению, эти командлеты будут работать только при прямом подключении, не через vCenter, поэтому к vCenter подключимся для получения списка серверов, затем будем подключаться к каждому по очереди.

Создадим цикл – будем подключаться к каждому ESXi из ранее созданного списка и выполнять для него настройку. В конце – отключаться от него.

```
# подключаемся к vCenter
Connect-VIServer vcenter -User <юзер> -Password <пароль>
# заносим в переменную все наши сервера ESXi
$esxis = Get-VMHost
# отключаемся от vCenter
Disconnect-VIServer vcenter -Confirm:$false
# начинаем цикл – перебираем по одному серверу из списка
# дело в том, что эти команды вроде как работают только
# при прямом подключении, без vCenter
foreach ($esxi in $esxis) {
    # подключаемся к текущему серверу
    Connect-VIServer $esxi -user root -Password <пароль рута>
    # заносим в переменную его настройки SNMP
    $hostsnmp = Get-VMHostSnmp
    # включаем snmp
    Set-VMHostSnmp -HostSnmp $hostsnmp -Enabled:$true
    # указываем community, с которым ESXi будет получать команды snmp
    Set-VMHostSnmp -HostSnmp $hostsnmp -ReadOnlyCommunity 'vssphererocommunity'
    # указываем, на какой сервер и с каким community слать trap
    Set-VMHostSnmp -HostSnmp $hostsnmp -AddTarget -TargetHost "192.168.22.250"
    -TargetCommunity "monitoring"
    # отключаемся от текущего сервера
    Disconnect-VIServer $esxi -Confirm:$false
}
```

Если захотим не изменить, а узнать настройки SNMP, то в этом цикле можно выполнить следующую команду:

```
Test-VMHostSnmp -HostSnmp $Hostsnmp
```

Библиотеки MIB доступны на сайте VMware: <http://downloads.vmware.com> ⇒ VMware vSphere ⇒ вкладка **Drivers & Tools** ⇒ **VMware vSphere 5.0 SNMP MIBs**.

4.6. Рекомендации по решению проблем

Здесь я постараюсь высказать некоторые общие соображения. Итак, вы столкнулись с проблемой. Это может быть что угодно – жалобы на тормоза виртуальной машины, самопроизвольные отключения ESXi от vCenter, периодическая недоступность конкретной ВМ по сети, неработоспособность какого-либо компонента vCenter и многое-многое другое. Не подумайте, что я вас запугиваю – продукты VMware довольно качественны, но ошибки возможны и в них. А еще больше ошибок возможно на стыке нескольких сервисов/инфраструктур – ведь vSphere зависит от сетевой инфраструктуры, от систем хранения, иногда от Active Directory и т. д. Так что ждать проблем не надо, а вот знать, как быть, если вдруг что, стоит.

Здесь я выскажу свои собственные, личные соображения по решению проблем. Рассчитаны они на читателя с не очень большим опытом – но проглядеть рекомендую в любом случае.

4.6.1. Статусные сообщения и файлы журналов (Logs&Events)

Проблема может быть диагностирована по сообщениям в системе статусных сообщений (events) и записям в файлах журналов (log-файлы). Нам надо извлечь потенциально интересные сообщения и воспользоваться ими. Здесь поговорим, как извлечь, а дальше – как воспользоваться.

Events

Выделив в клиенте vSphere объект, вы можете перейти на вкладку **Tasks&Events** – и, нажав кнопку **Events**, увидеть статусные сообщения для этого объекта и для его дочерних (если они есть). Таким образом, выделив ВМ, мы видим ее сообщения, выделив vCenter (корень иерархии) – видим вообще все сообщения.

Events – самый близко расположенный для нас источник информации. Обязательно пользуемся.

Обратите внимание: в пункте меню **Edit** ⇒ **Client Settings** ⇒ **List** мы можем указать, как много сообщений Events будет отображать клиент vSphere. Сами сообщения хранятся в базе данных vCenter.

Минусом этого инструмента можно назвать тот факт, что в заметном проценте сообщений может быть написано: «Еггог». А «что Еггог?», «где Еггог?», что делать и кто виноват – не написано. Поэтому нам могут потребовать файлы журналов – более детальные источники информации, но менее удобные.

Журналы

Файлы журналов («логи») есть у многих компонентов виртуальной инфраструктуры. В первую очередь следует выделить vCenter, ESXi, каждую отдельную VM.

Для работы с журналами необходимо знать две вещи: как до них добраться и в каком файле какая информация хранится.

Описание файлов журналов доступно в базе знаний: <http://kb.vmware.com/kb/1021806>. Кроме того, см. еще <http://kb.vmware.com/kb/1008524>.

А о том, как до них добраться, мы говорим более подробно.

Журналы работы конкретной VM расположены в ее каталоге на VMFS или NFS-хранилище. Если есть подозрения на проблемы этой единичной VM – следует ознакомиться с ее собственными журналами.

Кроме того, не забываем о том, что некоторые проблемы могут быть вызваны гостевой ОС и приложениями – их источниками информации также не стоит пренебрегать, хотя мы как администраторы vSphere не всегда имеем доступ к информации «изнутри» VM.

Файлы журналов vCenter расположены на той ОС, где установлен vCenter. Однако часто нет необходимости их искать – доступ к ним возможен прямо со страницы **Home** ⇒ **System Logs**.

Обычно наибольшую ценность представляют файлы журналов ESXi. Увидеть их мы можем сразу несколькими способами:

- подключившись к ESXi при помощи ssh или WinSCP. Файлы журналов ESXi по умолчанию сохраняются в каталоге /var/log;
- обратившись браузером по адресу <https://<адрес ESXi>/host>;
- подключившись клиентом vSphere напрямую к ESXi – тогда на домашней странице клиента будет пиктограмма **System Logs**;
- работая в клиенте vSphere через vCenter – если выделить сервер (кластер, датацентр) и выбрать в меню **File** ⇒ **Export** ⇒ **Export System Logs**. Впрочем, для отдельной VM это тоже работает;
- перенаправив файлы журналов на внешний сервер syslog или на VMFS/NFS-хранилище.

Экспорт журналов

Важно! Как бы вы ни получили доступ к журналам – для изучения сделайте их копию! Дело в том, что ESXi генерирует довольно много сообщений, и есть вероятность того, что сообщения за интересующий вас период времени будут удалены. Это случится рано или поздно, в зависимости от настроек логирования – количества лог-файлов в ротации и размера каждого файла. Например, если использовать Syslog-сервер от VMware, то при настройках по умолчанию глубина логирования может составлять только несколько часов.

Если вы обратились на ESXi при помощи клиента vSphere, то доступ файлам журналов обладает как плюсами, так и минусами. Из плюсов – не надо активировать никакого ssh, стандартный привычный инструмент. Из минусов – не все файлы журналов отображаются, скорее даже меньшая часть – пусть и самые ос-

новные. Обратите внимание на кнопку **Export System Logs** – с ее помощью вы легко выполните мою рекомендацию по созданию копии файлов журналов на текущий момент.

Кроме того, выполнить экспорт данных можно при помощи браузера, обратившись по ссылке <https://<пользователь на ESXi>:<пароль пользователя>@<адрес ESXi>/cgi-bin/vm-support.cgi>.

Еще более удобно сделать это через vCenter, пункт **Export System Logs** вы обнаружите в меню **File** ⇒ **Export** (только должен быть выделен хотя бы один сервер ESXi, если интересуют журналы хоста). Что важно – данный пункт позволяет экспортировать массу интересной информации, в том числе с нескольких. Когда вы запускаете мастер экспорта, то в нем вы отвечаете на вопросы о том, данные какого/каких объектов надо выгружать, какие именно данные, куда сохранять. В указанном месте обнаружится архив с указанием даты и времени экспорта. Распаковав его, вы увидите много каталогов. Некоторые из них будут полезны только для службы поддержки VMware, а некоторыми сможем воспользоваться и мы:

- ☐ в каталоге `/var` вы найдете непосредственно файлы журналов;
- ☐ в каталоге `/vmfs` – файлы виртуальных машин, кроме файлов журналов, мы найдем еще и конфигурационные файлы;
- ☐ в каталоге `/etc` – конфигурационные файлы с настройками ESXi на момент экспорта;
- ☐ в каталоге `/commands` – результаты выполнения различных команд. С их помощью мы узнаем очень много о конфигурации и состоянии сервера на момент экспорта. Это и таблица маршрутизации, и данные по сетевым объектам гипервизора в разных аспектах, и многое-многое другое.

Наконец, можно произвести экспорт диагностической информации при помощи PowerCLI, см. <http://kb.vmware.com/1027932>.

Syslog

В первой главе я описывал установку VMware Syslog Collector – сервера сбора журналов. На Linux-версии vCenter такой предустановлен. Можно использовать любой syslog-сервер. Плюс такого сервера централизованного сбора журналов:

- ☐ даже если сервер стал недоступен в силу проблем, у нас остаются доступными его журналы до последнего мгновения;
- ☐ все журналы всех серверов доступны в одном месте;
- ☐ некоторые syslog-сервера обладают дополнительными функциями, облегчающими анализ записей. Иногда даже автоматической реакцией на некоторые события (к сожалению, в реализации syslog-сервера от VMware ничего такого нет);
- ☐ ESXi создает для своей работы гат-диск. В некоторых случаях (особенно если он установлен на маленькую флэшку и LUN) все журналы остаются в этом гат-диске, не сохраняясь на системный диск. В таком случае каждая перезагрузка сервера гарантированно удаляет все журналы. При использо-

вании PXE-загрузки с VMware Auto Deploy это даже «не баг а фича», в том смысле что это нормальное и единственно возможное поведение системы в таких условиях. А вот перенаправление журналов на внешний сервер решает эту проблему.

Напомним, что, пройдя в расширенные настройки, **Configuration** ⇒ **Advanced Settings** ⇒ **Syslog**, мы найдем два интересующих нас сейчас параметра:

- ❑ **Syslog.global.loghost** – здесь мы укажем адрес syslog-сервера в формате `udp://<адрес IP>: порт` (см. всплывающую подсказку при наведении курсора на это поле);
- ❑ **Syslog.global.logdir** – здесь мы можем указать путь на VMFS/NFS-хранилище, куда будут сохраняться локальные файлы журналов. В каком-то смысле это альтернатива syslog-серверу, ведь если мы выберем для всех серверов какое-то одно хранилище с системы хранения, то в этом одном месте будут доступны журналы всех хостов, и доступны они будут, даже если недоступны сами сервера.

Обычно бывает удобно указать один и тот же путь для всех серверов ESXi и на всех поставить флажок **Syslog.global.logDirUnique** – в этом случае каждый сервер создаст по указанному пути подкаталог со своим IP-адресом в качестве имени и свои файлы журналов разместит уже в личном подкаталоге.

Я думаю, что будет очень удобно изменить эту настройку при помощи PowerCLI – сразу для всех хостов:

```
Connect-VIServer <vcenter>  
Get-VMHost | Set-VMHostAdvancedConfiguration -Name Syslog.global.logHost -Value  
udp://<адрес сервера syslog>:<порт>
```

4.6.2. Онлайн-источники информации

Самый главный наш помощник в онлайн – это база знаний VMware, <http://kb.vmware.com>. Часто в статусных сообщениях или журналах мы можем обнаружить сообщение об ошибке – вот поиск по тексту этого сообщения стоит выполнить обязательно, и начать следует именно с базы знаний. Информация там, как и везде, представлена на английском языке.

База знаний – с нее стоит начать. Но даже если там не удалось найти искомого – есть еще места, где стоит поискать (имеется в виду «перед тем как искать просто в google»). Подборка основных ссылок доступна тут – <http://link.vm4.ru/docs>.

4.6.3. Поддержка VMware

Любая компания, приобретая коммерческую лицензию vSphere, приобрела поддержку (Support&Subscription). Эта поддержка приобретается на какой-то срок, по истечении данного срока ее следует продлить (в частности, по той причине, что для компаний, у которых актуальна эта подписка, бесплатно обновление

на новые версии vSphere). Это означает, что у большинства читателей этих строк есть право обращаться в поддержку VMware в случае проблем.

Мой опыт подсказывает, что очень большой процент специалистов пренебрегает этим.

Мой совет: не стоит этого делать. В заметном проценте случаев обращение в поддержку обеспечит решение проблемы, и часто это решение будет найдено быстрее, чем если вы будете заниматься поиском решения сами, в свободное от остальной работы время.

Разумеется, не стоит полностью полагаться на поддержку. Если вы хотя бы в базе знаний по тексту ошибки поищите, то стандартные проблемы могут быть решены очень быстро. Но существует заметное количество проблем, решить которые самостоятельно сложно, или это займет очень много времени.

Просто для иллюстрации упомяну о некоторых проблемах из своего опыта (большинство уже неактуально сегодня, но для иллюстрации сойдет):

- ❑ был создан шаблон Windows 2003 x32, с него создано порядка 20 VM. Когда VM попадали на некоторые хосты (последние в кластере), начинались дикие тормоза мыши, видео и, в общем-то, всего. Притом VM на эти хосты мигрировала секунд за 20, а с них – все минут 10. Оказалось, что к шаблону остался подключен установочный образ iso. Следовательно, он считался подключенным к каждой развернутой из этого шаблона VM. Когда их число перевалило за второй десяток – начались вышеописанные проблемы;
- ❑ в сервере Dell установлен контроллер удаленного управления drac. Если ему IP-адрес выдан по DHCP – все отлично. Если статика – на ESX переставала работать сеть управления после каждой перезагрузки;
- ❑ клиент vSphere не подключался к vCenter, в сообщении об ошибке шла ругань на сертификаты SSL. Оказалось – мешал установленный на этой же машине «Криптопровайдер Avest CSP»;
- ❑ не получалось просмотреть содержимое каталога VM на VMFS-хранилище. Сама VM работоспособна, включается, выключается, мигрирует и прочее. А вот файлы не отображаются. Оказалось – мешал пробел в конце имени VM (и, следовательно, каталога этой VM);
- ❑ на серверах определенной модели возникали проблемы с сетью. Оказалось, в драйвере (или прошивке) той модели сетевого контроллера, что был установлен в этих серверах, была ошибка. Поддержка прислала совершенно неочевидную команду, выполнение которой отключало проблемную функцию этого контроллера, после чего все стало ОК.

Как видите, причина и следствие далеко не всегда очевидны, как и решение.

Официальная информация по обращению в поддержку VMware доступна по ссылкам

- ❑ <http://www.vmware.com/support/policies/howto.html>;
- ❑ <http://www.vmware.com/support/russia.html>;
- ❑ <http://www.vmware.com/support/policies/language.html>.

4.6.4. Core Dump, дампы

Если произошел критический отказ ESXi и хост упал в «пурпурный экран смерти», PSOD, то гипервизор создает так называемый «дамп» (dump) – архив с диагностической информацией. Будет хорошей идеей заранее настроить сбор этой информации по сети – см. раздел 1.3.3.

Зачем эти дампы нам пригодятся?

Самое главное – их может запросить поддержка VMware в случае инцидента.

А также их можно попробовать проанализировать самостоятельно. Единственный известный мне способ их анализа – это распаковка файла дампа командой `vmkdump_extract`, доступной в локальной командной строке ESXi 5. Вам может пригодиться статья базы знаний VMware № 1004250 (<http://kb.vmware.com/kb/1004250>).

4.7. Время на сервере ESXi

Еще пару слов хочу сказать про время серверов ESXi. Могут сбить с толку следующие факты:

1. ESXi не позволяет указывать часовой пояс (time zone). Его локальное время всегда UTC.
2. При отображении времени на сервере ESXi в клиенте vSphere время отображается в часовом времени *машины клиента*.

Таким образом, если вы посмотрите время на сервере ESXi разными способами:

- ☐ при помощи клиента vSphere (**Configuration** ⇒ **Time Configuration** или время событий, events);
- ☐ и из командной строки (вам поможет команда `date` или `hwclock`), –

то это время будет отличаться на смещение часового пояса той машины, где запущен клиент vSphere.

Таким образом, для беглой проверки правильности времени вам следует проверить, что время на сервере ESXi совпадает со временем вашей рабочей станции, если вы проверяете время ESXi через клиент vSphere, запущенный с вашей рабочей станции.

Но при просмотре одного и того же события в клиенте vSphere и в командной строке (или в экспортированных файлах журналов) вы обнаружите, что время, когда произошло это событие, отличается (на смещение часового пояса клиентской машины). Это нормально.

В случае каких-либо проблем рекомендую ознакомиться с документами:

- ☐ Timekeeping in VMware Virtual Machines (<http://www.vmware.com/vmtn/resources/238>);
- ☐ Timekeeping best practices for Windows, including NTP (<http://kb.vmware.com/kb/1318>);
- ☐ Timekeeping best practices for Linux guests (<http://kb.vmware.com/kb/1006427>);
- ☐ Troubleshooting timekeeping issues in Linux guest operating systems (<http://kb.vmware.com/kb/1011771>).



Глава 5. Виртуальные машины

Здесь поговорим про виртуальные машины (ВМ), что они из себя представляют, какими обладают возможностями, как мы их можем создать и какие манипуляции с ними производить.

В этой главе будут рассмотрены моменты создания ВМ, но не планирования. То есть на вопросы «сколько процессоров выдать ВМ», «диски какого размера создавать» здесь ответов не будет.

5.1. Создание ВМ. Начало работы с ней

Процесс создания ВМ с нуля прост и понятен. В клиенте vSphere нужно вызвать контекстное меню для сервера, кластера, пула ресурсов или каталога с ВМ и выбрать пункт **New Virtual Machine**. В любом случае запустится один и тот же мастер со следующими шагами.

1. **Configuration** – здесь мы можем выбрать, типичную или нет ВМ мы хотим создать. В случае выбора **Custom** мастер будет содержать больше вопросов. Те из шагов мастера, которые будут предложены только в случае выбора варианта Custom, я этим словом и буду помечать.
2. **Name and Location** – имя ВМ и в каком каталоге иерархии vCenter она будет расположена. В ваших интересах сделать имя понятным, уникальным и не содержащим пробелов и спецсимволов. Достаточно удобно делать имя виртуальной машины совпадающим с именем DNS гостевой операционной системы.
3. **Resource Pool** – в каком пуле ресурсов расположена ВМ. Эта страница мастера не выводится, если ВМ создается в кластере без функции DRS или пулов ресурсов просто нет.
4. **Storage** – на каком хранилище будут располагаться файлы ВМ. Обратите внимание на выпадающее меню **VM Storage Profile** – выбрав в нем соответствующий профиль для создаваемой ВМ, вы получите подсказку – на каких хранилищах эту ВМ лучше создавать. Разумеется, функция **Profile Driven Storage** должна быть настроена предварительно.
5. **Virtual Machine Version (Custom)** – здесь мы можем выбрать версию виртуального оборудования. Версия 8 – новая. Версия 7 – старая, совместимая с ESX(i) 4.x. Старую имеет смысл выбирать, лишь если у вас есть сервера ESX(i) 4.x и эта ВМ может оказаться на них. По умолчанию (в варианте Typical) выбирается версия 8.

6. **Guest Operating System** – тип гостевой ОС. Он указывается для того, чтобы ESXi правильно выбрал дистрибутив VMware tools, предложил больше или меньше памяти по умолчанию, тот или иной тип виртуальных SCSI и сетевого контроллеров. Изменять значение этого поля можно и после создания ВМ. А после установки VMware Tools оно будет изменяться автоматически, согласно полученной от VMware Tools информации.
7. **CPUs (Custom)** – выбираем количество виртуальных процессоров. В пятой версии ESXi виртуальные процессоры могут быть многоядерными.
8. **Memory (Custom)** – можем указать объем памяти для ВМ. По умолчанию выбирается небольшой объем в зависимости от типа гостевой ОС. Значения по умолчанию – константы, прописанные в ESXi.
9. **Network** – можем указать количество виртуальных сетевых контроллеров, их тип и в какие группы портов они подключены. По умолчанию создается один виртуальный сетевой контроллер оптимального для выбранной гостевой ОС типа. Он подключается к первой в алфавитном порядке группе портов. «Оптимальный» в данном случае – не всегда самый производительный или функциональный, а некий баланс между совместимостью и функциональностью. Например, если в ОС есть драйвер для e1000, то будет выбран именно он, несмотря на то что vmxnet2/3 тоже будет работать (и возможно, работать лучше), но для этого нужны VMware Tools. Про типы виртуальных сетевых контроллеров читайте в разделе о виртуальном оборудовании.
10. **SCSI Controller (Custom)** – тип виртуального контроллера, SCSI к которому будут подключены виртуальные диски ВМ. О разнице расскажу позже, в разделе о виртуальном оборудовании. Выбираемый по умолчанию зависит от типа гостевой ОС.
11. **Select a Disk (Custom)** – можно выбрать создание нового виртуального диска, подключение существующего виртуального диска, подключение LUN как Raw Device или создание ВМ без дисков вообще. По умолчанию создается новый виртуальный диск.
12. **Create a disk** – здесь вы можете выбрать размер создаваемого диска для ВМ и его тип (рис. 5.1).

Размер диска по умолчанию зависит от системных требований выбранной ранее гостевой ОС.

Тип диска по умолчанию – **Thick Provisioning Lazy Zeroed**. Это означает, что файл виртуального диска займет сразу 100% места на хранилище (40 Гб в моем примере).

Нижний вариант, **Thin Provisioning**, создаст диск в thin, «тонком» режиме. Это означает, что место под этот файл-диск ВМ не выделится сразу, а будет выделяться лишь по мере необходимости.

Тип диска **Thick Provisioning Eager Zeroed**. Он необходим, если создаваемая ВМ будет узлом кластера Microsoft (MSCS/MFC) или VMware Fault Tolerance. Тогда не только все место, отведенное под виртуальный диск, будет размечено сразу (как происходит по умолчанию), но еще и каждый блок соз-

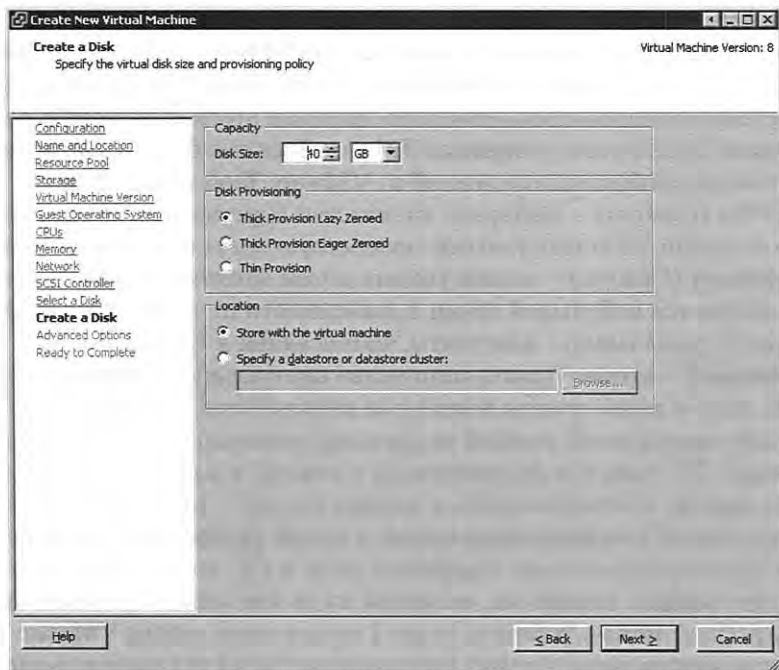


Рис. 5.1. Этап настроек создаваемого виртуального диска в мастере создания VM

даваемого виртуального диска будет перезаписан нулями в момент создания (а не в момент первого использования, как в остальных случаях). Побочным эффектом обнуления является увеличенное время создания такого виртуального диска. Более подробно о типах дисков расскажу позже, в разделе о виртуальном оборудовании. В случае варианта мастера Custom можно отдельно указать хранилище для создаваемого виртуального диска. Тогда на выбранном ранее хранилище будут храниться только файл настроек, журналы и другие файлы VM. Большинство из них текстовые, небольшого размера. Единственное исключение – файл подкачки, который для VM создает гипервизор. По умолчанию его размер равен объему выделенной для VM памяти.

13. **Advanced Options (Custom)** – здесь можно указать, что диск VM будет на контроллере IDE (пригодится для тех ОС, которые не поддерживают интерфейсы SCSI), и установить флажок **Independent** (о том, что это такое, см. далее).
14. **Ready to Complete** – здесь можно поставить флажок **Edit the virtual machine settings before completion**, что дает возможность удалить или добавить какое-то оборудование в эту VM непосредственно перед ее созданием. Впрочем, это замечательно делается и потом.

Созданная сейчас VM – это виртуальный сервер. На нем еще нет операционной системы, но его уже можно включить. Нам надо загрузить в него операцион-

ную систему и приложения. Кстати, операционная система, установленная внутри виртуальной машины, называется **Гостевая ОС**.

В большинстве случаев мы поступаем с ВМ так же, как с физическим сервером, – устанавливаем ОС на диски. Самый очевидный способ это сделать – подключить к приводу CD-ROM этой ВМ физический диск или образ в формате ISO с дистрибутивом ОС и установить ее с этого диска. Установка операционных систем в ВМ ничем не отличается от установки их на обычные сервера. Разве что у ВМ нет физического монитора, но его небезуспешно заменяет консоль клиента vSphere (рис. 5.2).

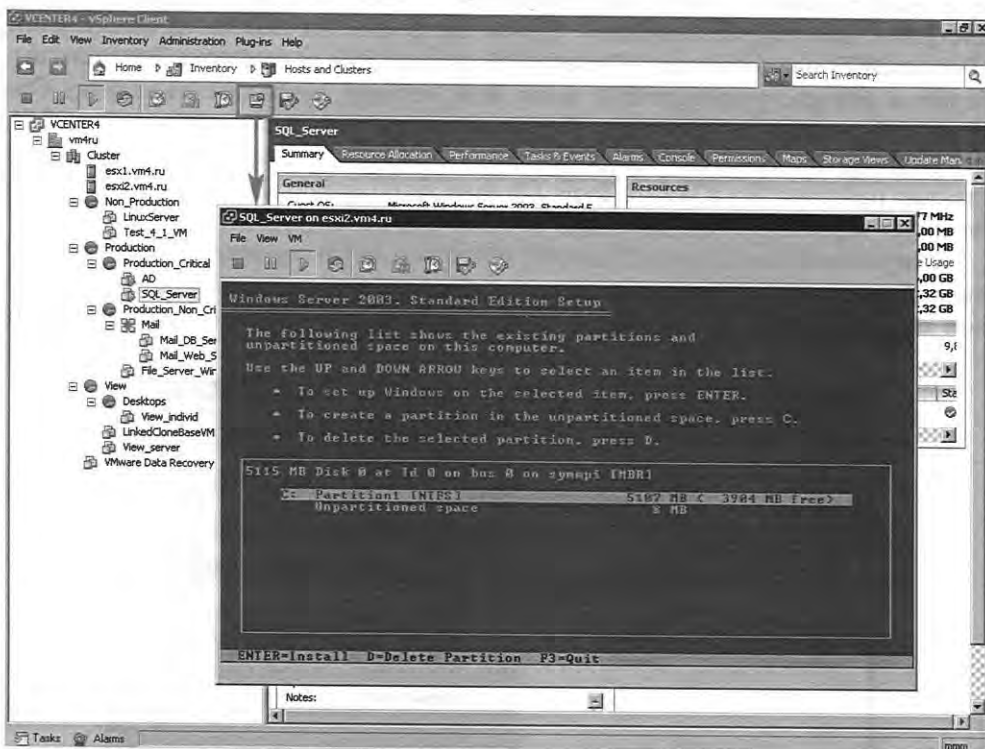


Рис. 5.2. Пиктограмма открытия консоли ВМ в отдельном окне и сама консоль

Консоль можно запустить из контекстного меню ВМ, кнопкой на панели инструментов клиента vSphere (отмечена на рисунке) или на вкладке **Console** для ВМ. В первых двух случаях консоль открывается в отдельном окне, в верхней части которого присутствуют дополнительные элементы управления. Одни из самых часто используемых вынесены на панель инструментов (рис. 5.3).

Первая группа из четырех кнопок – управление питанием. Выключить ВМ, поставить на паузу (состояние «suspend»), включить и перезагрузить. Обратите внимание на то, что в ESXi 5 эти кнопки по умолчанию настроены на корректное



Рис. 5.3. Панель инструментов консоли VM

выключение (**Shutdown guest**) и корректную перезагрузку (**Restart Guest**) гостевой ОС. Эти операции могут быть выполнены, лишь если в гостевой ОС установлены и запущены VMware tools. В некоторых ситуациях, например сейчас, когда даже ОС еще не установлена, такая настройка может оказаться не очень удобной – при нажатии кнопок **Power Off** и **Reset** мы получаем ошибку «VMware tools недоступны, действие невозможно». Чтобы настроить эти иконки на выключение (**Power OFF**) и перезагрузку (**Reset**) VM, надо зайти в ее свойства. Для этого выберите в контекстном меню VM пункт **Edit Settings**, перейдите на вкладку **Options**, выберите пункт **VMware tools** и измените настройку на желаемую (рис. 5.4).

Впрочем, изменять эти настройки имеет смысл лишь для тех VM, которые постоянно работают без VMware Tools. Обычно такое происходит тогда, когда не существует версии VMware Tools для гостевой ОС, используемой в VM.

Операция приостановки (**Suspend**) позволяет зафиксировать текущее состояние работающей виртуальной машины путем выгрузки содержимого ее оперативной памяти в файл. Таким образом, виртуальная машина останавливается и не

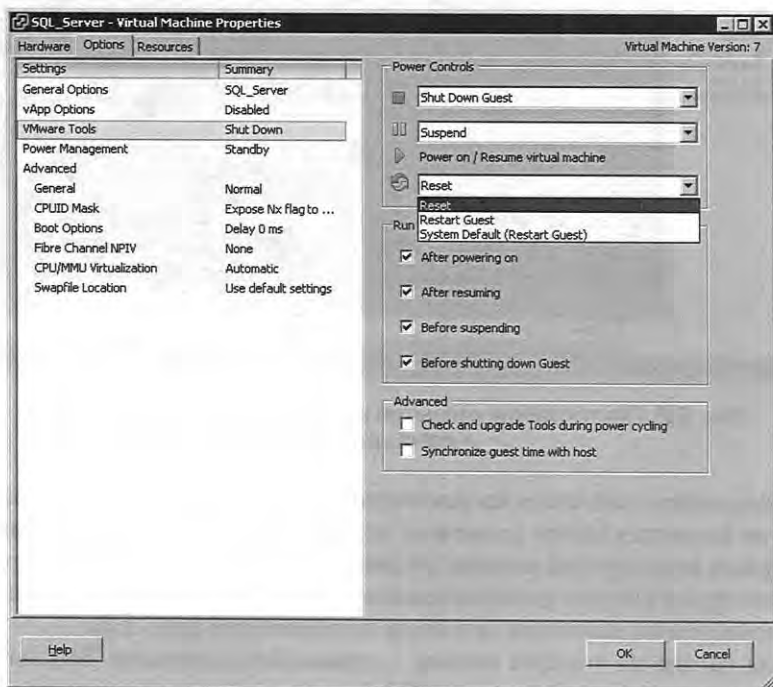


Рис. 5.4. Выбор действия для кнопок управления питанием VM

потребляет ресурсов сервера, но при возобновлении ее работы мы возвращаемся в состояние на момент приостановки.

Затем идут кнопки управления снимками состояния (snapshot).

Последние три – настройки виртуальных CD/DVD-ROM, FDD и USB. Обратите внимание на то, что с помощью этих кнопок можно подключать к CD/DVD (FDD) виртуальной машины как CD/DVD (FDD) клиентского компьютера, так и образы (ISO или Flp) с диска клиентского компьютера. Подключение устройств и образов с клиентского компьютера выполняется только через этот элемент интерфейса. Подключения прочих вариантов (образов, доступных с сервера ESXi и физических устройств сервера) доступны просто из окна настроек ВМ.

Кнопка подключения USB появилась лишь в пятой версии ESXi. Если в конфигурацию виртуальной машины добавлен контроллер USB, то при помощи описываемой пиктограммы мы можем подключить к ВМ устройство USB с нашего рабочего места.

Из пункта меню **VM** ⇒ **Guest** доступны пункты для установки и обновления VMware tools, изменения настроек этой ВМ, запуска миграции этой ВМ, клонирования, снятия шаблона и включения Fault Tolerance.

Если в ВМ не установлены VMware tools, то, щелкнув мышкой внутрь окна консоли, вы передаете туда фокус ввода. Чтобы вернуть его в ОС своего компьютера (в которой запущена консоль), нажмите **Ctrl+Alt**.

Чтобы передать в ВМ комбинацию **Ctrl+Alt+Del**, нажмите **Ctrl+Alt+Ins** или воспользуйтесь меню: **VM** ⇒ **Guest** ⇒ **Send Ctrl+Alt+Del**.

С помощью консоли можно производить все необходимые действия с ВМ. За работу этой консоли отвечает сам ESXi, а не какое-то ПО в виртуальной машине, и трафик данной консоли идет через управляющие интерфейсы ESXi. Также доступ к такой консоли ВМ можно получить через веб-интерфейс vSphere (рис. 5.5). До-



Рис. 5.5. Веб-интерфейс vCenter

ступ через веб-интерфейс часто удобен операторам ВМ, для выполнения работы которых вы посчитаете нецелесообразным устанавливать консоль.

Для работы веб-интерфейса в vSphere 5 необходим компонент vSphere Web Client (Server). См. раздел, посвященный его установке.

5.2. Клонирование и шаблоны ВМ (Clone и Template)

Устанавливать и настраивать ОС в виртуальную машину можно точно так же, как и на машину физическую. Однако есть способы лучше. Первый из таких способов – сделать копию, клон существующей ВМ, второй – механизм шаблонов (Templates). Обратите внимание на то, что оба этих механизма доступны только при работе через vCenter.

5.2.1. Клонирование виртуальных машин

Для выполнения клонирования в контекстном меню ВМ, копию которой вы хотите сделать, выберите пункт **Clone**. Запустится мастер.

1. **Name and Location** – укажите имя для создаваемой ВМ, в каком Datacenter (датацентре) и каталоге она будет расположена.
2. **Host/Cluster** – укажите, в каком кластере и на каком сервере будет зарегистрирована создаваемая ВМ.
3. **Resource pool** – в каком пуле ресурсов ВМ будет находиться.
4. **Storage** – на каком хранилище будут располагаться ее файлы. По кнопке **Advanced** можно указать разные хранилища для ее файла настроек и виртуальных дисков.

Если настроена функция **Profile Driven Storage**, то, выбрав в выпадающем меню **VM Storage Profile** желаемый профиль, мы получим подсказку о том, какие хранилища предпочтительнее использовать.

Выпадающее меню **Select a virtual disk format** позволит выбрать, в каком формате будут созданы виртуальные диски новой ВМ.

5. **Guest Customization** – надо ли обезличивать гостевую ОС, если надо – то с какими параметрами. Подробности по поводу обезличивания см. чуть далее.
6. Все.

Обратите внимание на то, что операция клонирования может проводиться и для работающей ВМ (правда, без гарантии целостности данных).

Операция клонирования может помочь вам еще в следующих ситуациях.

- ☐ Иногда бывает, что ВМ создается с одним именем, а затем ее переименовывают. Но переименование виртуальной машины в иерархии vCenter не меняет имен ее файлов, и получается, что каталог и файлы этой ВМ имеют имя, отличное от видимого нам в клиенте vSphere. Это неудобно. Один из удобных способов привести имена в соответствие – клонировать ВМ и удалить исходную. У клона название каталога и файлов будет совпадать с име-

нем ВМ в клиенте vSphere (здесь следует сделать замечание: клонировать надо не после переименования, а вместо). Еще один способ – выполнить холодную миграцию на другое хранилище, или Storage VMotion. Мигрировать следует после переименования.

На самом деле для переименования Storage vMotion или миграция выключенной ВМ на другое хранилище лучше, чем клонирование, но в vSphere 5 на момент написания при этих операциях переименоывается только каталог, но не файлы ВМ в нем.

- Если вы хотите избавиться от снимков состояния (snapshot). Про них еще поговорим, но забегаю вперед: иногда процесс удаления снимков может оказаться нетривиальным и требовать много свободного места на хранилище. В таких случаях проще клонировать эту ВМ – у клона снимков состояния уже не будет.

В пятой версии vSphere для решения такой беды появилась специальная функция – в контекстном меню ВМ пункт **Snapshot ⇒ Consolidate**. Но я допускаю, что клонирование в этих целях все равно сможет иногда быть более удобным.

5.2.2. Шаблоны виртуальных машин (template)

Еще интереснее, чем клонирование, создание шаблонов (Template) для последующего развертывания типовых ВМ. По сути, механизм Templates – это альтернатива созданию образов дисков с помощью ПО типа Windows Deployment Services (WDS) Image Capture Wizard (WDSCapture), ImageX, Acronis Snap Deploy или Symantec Ghost. Файлы ВМ, особенно виртуальные диски, – что это, если не образ? Образ и есть. Так вот, в vCenter есть механизм, который позволит вам сделать эталонную копию ВМ – шаблон.

В контекстном меню ВМ есть пункт **Templates**, в котором находятся две операции – **Clone to Template** и **Convert to Template**. Соответственно, первый нужен, когда вы хотите и сохранить ВМ, и получить из нее шаблон. А второй – когда ВМ вам не нужна и вы хотите *превратить* ее в шаблон. Клонировать в шаблон при необходимости можно и работающую ВМ. Конвертация происходит практически мгновенно, так как копировать файлы ВМ не нужно – достаточно пометить их как файлы шаблона.

В иерархии **Host and Clusters** вы не видите шаблоны в иерархии объектов, однако здесь можно выделить сервер, кластер или датацентр и, перейдя на вкладку **Virtual Machines**, увидеть и ВМ, и шаблоны (рис. 5.6). Шаблоны можно отличить по пиктограмме.

Или пройдите **Home ⇒ Inventory ⇒ VMs and Templates**, тогда шаблоны будут отображаться и в иерархии объектов. Этот вариант при работе с шаблонами удобнее.

Обратите внимание на то, что шаблон, как и ВМ, числится на каком-то из серверов. Увидеть, на каком именно, можно на вкладке **Summary** для шаблона. Это

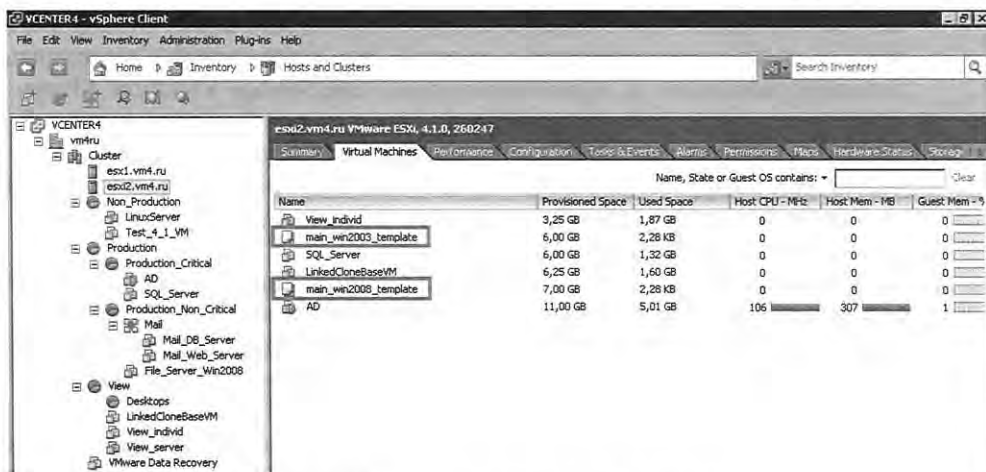


Рис. 5.6. Шаблоны виртуальных машин

создает небольшие проблемы, потому что шаблоном невозможно воспользоваться, когда сервер, на котором он числится, недоступен. Такое возникает в нештатных ситуациях, когда сервер выходит из строя, или во время обслуживания сервера. Еще подобная ситуация возможна при штатной работе DPM, в таком случае шаблоны имеют смысл настроить числящимися на не выключаемом DPM сервере. Если такая ситуация произошла, самым простым выходом является следующий:

1. Удалить недоступный шаблон из иерархии vCenter, выбрав в его контекстном меню пункт **Remove from Inventory**.
2. Пройти **Home** ⇒ **Inventory** ⇒ **Datastores** и вызвать контекстное меню для того хранилища, где шаблон расположен. Само собой, предполагается, что он расположен не на локальных дисках недоступного сейчас сервера. Выберите пункт **Browse Datastore**.
3. В диспетчере файлов хранилища найдите в каталоге шаблона файл с расширением **.vmtx**. В его контекстном меню есть пункт **Add to Inventory** (рис. 5.7).

Файл шаблона с расширением **.vmtx** – это не что иное, как файл настроек, в случае виртуальной машины имеющий расширение **.vmx**. В общем-то, одной буквой в расширении файла настроек шаблон и отличается от ВМ. Это изменение сделано для того, чтобы шаблон был незапускаемой ВМ, дабы уберечь эталонную виртуальную машину от изменений вследствие случайного включения.

Чтобы воспользоваться шаблоном, найдите его. Проще всего это сделать, пройдя **Home** ⇒ **Inventory** ⇒ **VMs and Templates**. Выберите в его контекстном меню пункт **Deploy Virtual Machine from this Template**, запустится мастер:

1. **Name and Location** – укажите имя для создаваемой ВМ, в каком Datacenter (датацентре) и каталоге она будет расположена.
2. **Host / Cluster** – укажите, в каком кластере и на каком сервере будет зарегистрирована создаваемая ВМ.

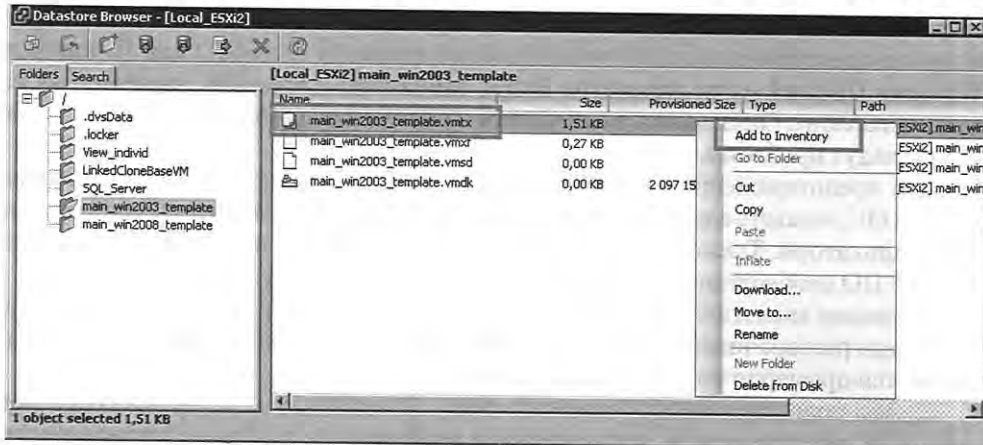


Рис. 5.7. Регистрация шаблона

3. **Resource pool** – в каком пуле ресурсов VM будет находиться.
4. **Storage** – на каком хранилище будут располагаться ее файлы. Нажав кнопку **Advanced**, можно указать разные хранилища для ее файла настроек и виртуальных дисков.

Если настроена функция **Profile Driven Storage**, то, выбрав в выпадающем меню **VM Storage Profile** желаемый профиль, мы получим подсказку о том, какие хранилища предпочтительнее использовать.

Выпадающее меню **Select a virtual disk format** позволит выбрать, в каком формате будут созданы виртуальные диски новой VM.

5. **Guest Customization** – надо ли обезличивать гостевую ОС, если надо – то с какими параметрами. Подробности по поводу обезличивать см. чуть далее.
6. На последнем шаге есть возможность поставить флажок **Edit virtual hardware (Experimental)**. Если он стоит, то по нажатии **Finish** откроется окно настроек VM, и мы сможем изменить настройки и сам набор виртуального оборудования. Это может быть полезно для указания группы портов для сети новой VM – иначе она окажется подключенной к той группе портов, куда был подключен шаблон. Статус **Experimental** для этого флажка говорит о том, что VMware не гарантирует стабильной работы этой функции.
7. Все.

Пока что я описал только доступные нам действия с шаблонами. Однако даже более важным является то, какой должна быть эталонная VM. См. раздел 5.2.4 «Рекомендации для эталонных VM».

5.2.3. Обезличивание гостевых ОС, SysPrep

Когда мы клонируем уже установленную ОС, мы можем получить проблемы из-за полной идентичности старой и новой операционных систем. Притом не важно, пользуемся ли мы шаблонами виртуальных машин в vCenter или разворачива-

ем ранее снятые образы на физические сервера. Имя машины, статичный IP-адрес и другие идентификаторы (в случае Windows это могут быть SID или ID клиента Windows Update) – эти параметры будут одинаковы у всех ВМ, развернутых из одного шаблона. Оказавшись в одной сети и подключаясь к одним серверам, такие ОС создадут проблемы.

Для предотвращения этих проблем нам необходимо давать уникальное имя каждой ОС, задавать адрес IP, если он назначается вручную, и генерировать прочие идентификаторы. Также в некоторых случаях для каждого экземпляра Windows и другого ПО необходимо указывать собственный ключ продукта.

Проведем аналогию с неvirtуальной средой. Например, в случае, когда у нас на сервера разворачиваются образы, также необходимо обезличивание. И оно выполняется примерно по следующему плану:

1. Мы подготавливаем эталонный образ. Устанавливаем на один из серверов ОС, настраиваем ее, устанавливаем необходимые драйверы и приложения.
2. Удаляем из ОС уникальную информацию (обычно это имя, сетевые настройки, ключ продукта и т. д.). Microsoft требует, чтобы при клонировании Windows для этой операции использовалась утилита System Preparation Tool (SysPrep).
3. Снимаем образ такой «обезличенной» системы.
4. При развертывании ОС из подобного образа удаленная информация заменяется данными, уникальными для каждой установки. Это выполняется вручную или автоматически.

В виртуальной среде мы можем (и иногда вынуждены) поступать точно так же. Только этап № 3 выглядит так: «преобразуем ВМ с обезличенной ОС в шаблон».

Однако подобный подход не лишен недостатков. Самый неприятный – сложность обновления образа. Чтобы добавить в образ (или шаблон) что-то новое, например обновления, необходимо:

1. Развернуть этот образ.
2. Задать уникальную информацию (имя, сетевые настройки и т. д.).
3. Произвести необходимые изменения (установить обновления, добавить ПО).
4. Удалить уникальные параметры (имя, сетевые настройки и т. д.).
5. Заново снять образ.

Получается, мы должны пройти полный цикл, хотя хотим всего лишь образ обновить.

Для некоторых гостевых операционных систем vCenter реализует более удобный подход к обезличиванию, который лишен этого недостатка. Суть подхода в следующем: инструмент для обезличивания устанавливается не в гостевую ОС, а на сервер vCenter. И само удаление уникальных параметров происходит не на эталонной системе перед снятием с нее образа (преобразованием в шаблон), а уже на развернутой копии. В таком случае последовательность действий выглядит следующим образом.

1. Мы создаем эталонный образ. Устанавливаем гостевую ОС, настраиваем ее, устанавливаем необходимые приложения.

2. Преобразуем эту VM в шаблон. Обратите внимание: обезличивания не происходит.
3. Разворачиваем VM из этого шаблона. Мастер развертывания описан в предыдущем разделе.
На шаге **Guest Customization** нас спрашивают, надо ли обезличивать разворачиваемую ОС. Если надо, то нас просят указать, на что заменить уникальную информацию.
4. Файлы шаблона копируются в новую VM, ее диск монтируется к серверу vCenter. vCenter помещает ПО для обезличивания на диск новой VM и настраивает автоматический запуск этого ПО при первом старте системы. Кроме того, в образ помещается файл ответов для мастера мини-установки (Mini-Setup). В файле ответов содержатся данные, указанные нами в мастере развертывания VM из шаблона. Благодаря файлу ответов мини-установка ОС проводится в автоматическом режиме, избавляя нас от необходимости указывать параметры в мастере при первом запуске VM.

Такой подход хорош тем, что для обновления шаблона достаточно конвертировать его в VM, включить ее и, никак не меняя уникальную информацию, внести необходимые изменения, после чего конвертировать обратно в шаблон, не удаляя уникальную информацию снова.

Для операционных систем Linux vCenter поддерживает обезличивание ОС из следующего списка:

- ☐ Red Hat Enterprise Linux AS версий от 2 до 5 (включая 64-битные версии);
- ☐ Red Hat Application Server версий от 2 до 5 (включая 64-битные версии);
- ☐ SUSE LINUX Enterprise Server 8, 9, 10.

Впрочем, список может обновляться, и актуальный стоит смотреть в документации (<http://pubs.vmware.com>).

Притом на vCenter не нужно чего-то доустанавливать или настраивать. Он умеет делать обезличивание этих ОС «из коробки».

Если у вас другие версии *nix, то для их обезличивания VMware не предлагает средств, отличных от стандартных для этих ОС. То есть обезличивать их придется точно так же, как это делается в случае физических серверов.

В случае Windows vCenter нам поможет со следующими ОС:

- ☐ Windows 2000 Server, Advanced Server, или Professional;
- ☐ Windows XP Professional (включая 64-битные версии);
- ☐ Windows Server 2003, Web, Standard, или Enterprise Editions (включая 64-битные версии);
- ☐ Windows Vista (включая 64-битные версии);
- ☐ Windows Server 2008;
- ☐ Windows 7.

Для обезличивания Windows Vista, Windows 7 и Windows Server 2008 vCenter использует средства, встроенные в сами эти ОС.

Для того чтобы vCenter помог нам с обезличиванием ОС более ранних версий, чем Windows Vista, нам необходимо поместить утилиту sysprep на сервер vCenter. Порядок действий следующий.

1. Находим эту утилиту для нужной версии ОС. Взять ее можно из архива deploy.cab, который располагается на диске с дистрибутивом этой ОС (\SUPPORT\TOOLS\deploy.cab), или загрузить с сайта Microsoft (версия SysPrep должна совпадать с версией ОС вплоть до пакетов обновления, поэтому если мы устанавливали Service Pack на операционную систему, то версия с диска может и не подойти). Рекомендую ознакомиться со статьей базы знаний VMware <http://kb.vmware.com/kb/1005593>. В ней вы найдете прямые ссылки на sysprep разных версий ОС и памятку о путях, по которым их следует расположить.
2. Скопировать файлы Sysprep по необходимому пути. vCenter может быть установлен на разные ОС, путь будет отличаться:
 - Windows 2008 – %AllUsersProfile%\VMware\VMware VirtualCenter\sysprep\<каталог с нужной версией Windows в названии>;
 - Windows 2003 – %ALLUSERSPROFILE%\Application Data\VMware\VMware VirtualCenter\Sysprep\<каталог с нужной версией Windows в названии>;
 - vCenter Virtual Appliance – /etc/vmware-vpx/sysprep/<каталог с нужной версией Windows в названии>.

Распишу мастер обезличивания на примере Windows Server 2003.

1. Sysprep скопирован по нужному пути на сервер vCenter. Вы запустили мастер клонирования или разворачивания ВМ из шаблона. На шаге **Guest Customization** вы выбрали **Customize using the Customization Wizard**. Запустился отдельный мастер, в котором надо указать уникальную для разворачиваемой копии ОС информацию.
2. **Registration Information** – на кого зарегистрирована эта копия гостевой ОС.
3. **Computer Name** – имя ОС в ВМ. Часто удобно бывает поставить переключатель **Use the virtual machine name**. Тогда в качестве имени ОС будет использоваться название ВМ. Однако предварительно стоит убедиться в том, что в имени ВМ нет запрещенных символов. Например, когда из прочих соображений используются для названия ВМ FQDN (computer.domain.com). Если установить описываемый флажок, мастер возражать не будет, но вот мини-установка закончится ошибкой, потому что в имени компьютера в Windows нельзя использовать точки.
Кроме того, можно указать **Enter a name in the Deploy wizard** – если мы сохраним наши ответы данному мастеру в файл ответов, то имя не сохранится и каждый раз будет спрашиваться. Удобно, если DNS-имя гостевой ОС не совпадает с именем ВМ.
4. **Windows License** – надо указать ключ продукта и тип лицензирования.
5. **Administrator Password** – пароль администратора и количество автоматических аутентификаций под его учетной записью.
Обратите внимание: если пароль администратора в гостевой Windows пустой, то Sysprep не сможет его поменять.
6. **Time zone** – часовой пояс.

7. **Run once** – произвольные команды, которые будут выполнены в гостевой ОС после обезличивания. Могут пригодиться для выполнения каких-то специфических настроек. Например, добавив команду

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update" /v AUOptions /t REG_DWORD /d
```

мы отключим автообновление в этой гостевой ОС.
8. **Network** – настройки сети.
9. **Workgroup or Domain** – членство в рабочей группе или домене. Имя домена стоит указывать в FQDN, имя учетной записи для ввода в домен – в виде user@domain.com.
10. **Operating System Options** – скорее всего, вам не надо снимать флажок **Generate New Security ID (SID)**.
11. **Save Specification** – мы можем сохранить эти ответы на вопросы мастера, с тем чтобы не отвечать на те же вопросы при разворачивании однотипных ВМ. Вводим имя и описание файла ответов для обезличивания. Однажды созданный, этот файл ответов можно выбирать в пункте № 0 данного списка, выбрав там настройку **Customize using an existing customization specification**.

Обратите внимание на то, что в интерфейсе клиента vSphere есть пункт **Home** ⇒ **Management** ⇒ **Customization Specification Manager**. Пройдя туда, вы увидите все существующие файлы ответов для мастера обезличивания, сможете их изменять, удалять и создавать новые. Также есть возможность их импорта и экспорта. Хранятся эти файлы в БД vCenter, пароли администратора хранятся в зашифрованном виде. Обратите внимание: для шифрования используется сертификат vCenter. Это означает, что при переустановке vCenter (но использовании той же базы данных) доступ к зашифрованным паролям теряется.

Для обезличивания гостевых ОС с помощью vCenter есть некоторые условия:

- ☐ обезличивание с помощью vCenter невозможно для контроллера домена;
- ☐ загрузочный диск ВМ должен быть первым диском на первом контроллере, то есть быть подключенным к узлу SCSI 0:0;
- ☐ в ВМ должны быть установлены VMware tools;
- ☐ Sysprep должен быть скопирован в правильный каталог на сервере vCenter (для обезличивания Windows версий более ранних, чем Windows Vista).

Обратите внимание. В эталонной ВМ может быть какое-то ПО, требующее отдельного обезличивания, обычно это какие-то агенты, например антивируса или системы мониторинга. Для них обезличивание придется производить отдельно, их собственными средствами. VMware не дает каких-то специальных инструментов для решения подобных задач.

5.2.4. Рекомендации для эталонных ВМ

В общем случае вот что можно порекомендовать делать для шаблонов ВМ с упором на Windows:

- ❑ выполнить выравнивание диска (disk alligment, о нем чуть ниже, в разделе про виртуальные диски). Загрузочный диск следует выравнивать до установки ОС;
- ❑ настройте BIOS при необходимости. Например, пароль, порядок загрузки;
- ❑ внесите типовые для вашей инфраструктуры изменения в файл настроек (*.vmx), если таковые есть. Например, настройки
vlsance.noOpram = "true"
vmxnet.noOpram = "true"
запретят загрузку по pxe для контроллеров типа flexible и vmxnet. Это может быть нужно из соображений безопасности.
Или вы захотите, чтобы в консоли ВМ работал буфер обмена между ОС клиента и гостевой ОС – для удобства.
Информацию о подобных параметрах файла *.vmx ищите в первую очередь в документе vSphere Hardening Guide, раздел Virtual Machines;
- ❑ само собой, устанавливать последние обновления (даже если вы создаете шаблон в тестовых целях, хотя бы обновление отмены перехода на зимнее время имеет смысл поставить, <http://support.microsoft.com/kb/2570791>);
- ❑ установить VMware tools;
- ❑ поменяйте тип SCSI-контроллера на наиболее оптимальный из поддерживаемых. Скорее всего, это VMware Paravirtual SCSI, но ознакомьтесь с документацией – поддерживается ли он для используемой гостевой ОС;
- ❑ удалить файлы для отката обновлений из %systemroot%. Обычно это каталоги \$NTUninstallxxxxxx\$ и \$NTServicePackUninstall\$ (для ОС Vista и старше используйте compcln.exe);
- ❑ дефрагментировать жесткие диски ВМ (для ВМ с тонкими дисками не стоит);
- ❑ в документации вашей системы хранения вам наверняка встретятся рекомендации повысить время ожидания отклика от диска в гостевой ОС. Для этого в ключе реестра **HKEY_LOCAL_MACHINE ⇒ System ⇒ CurrentControlSet ⇒ Services ⇒ Disk** укажите значение в 60 (для FC/iSCSI) или в 125 (NFS);
- ❑ некоторые источники рекомендуют отключать скринсейвер, в том числе тот, что работает при отсутствии залогиненных пользователей;
- ❑ установить и настроить все необходимые службы ОС. Например, Remote Desktop, IIS, SNMP и т. п.;
- ❑ установить типовое ПО. Обратите внимание, что это ПО должно нормально относиться к смене имени ОС;
- ❑ не пренебрегайте полем Описание (**Description**). Хорошей привычкой является занесение туда полной информации о шаблоне (зачем был сделан этот шаблон и как предполагается его использование. Например: «типовой узел кластера для промышленных нагрузок») и дате последнего изменения;
- ❑ бывает удобно в имени шаблонов использовать префикс, который однозначно их отличает от виртуальных машин. Например, Template_Win2008r2, Template_SUSE и т. д. Кроме того, будет хорошей идеей создать

отдельный каталог в иерархии **VMs & Templates** именно для шаблонов и все их туда помещать;

- ❑ шаблон и разворачиваемые из него VM числятся подключенными к той же группе портов на виртуальном коммутаторе, что и та VM, из которой был сделан шаблон. Если это была группа портов на стандартном коммутаторе и ее нет на том сервере, куда вы развернули новую VM, то VM нормально развернется, но ее виртуальные сетевые контроллеры не будут никуда подключены. Если это была группа портов на распределенном виртуальном коммутаторе и ее нет на том сервере, куда вы разворачиваете новую VM, процесс разворачивания остановится с ошибкой;
- ❑ обратите внимание на то, что нет простой возможности увеличить размер диска для разворачиваемой из шаблона VM. Поэтому для вас может иметь смысл создавать небольшой загрузочный диск в шаблоне, а для данных использовать дополнительные виртуальные диски, индивидуального размера для каждой VM. Также нет простой возможности уменьшить размер диска VM, поэтому заведомо большее, чем необходимо, количество гигабайт выдавать не стоит.

Обратите внимание. При конвертации шаблона в VM лучше всего использовать то же хранилище – тогда при конвертации не будет копирования файлов шаблона, что значительно ускорит процедуру. VMware рекомендует выделять отдельное хранилище (LUN) под шаблоны (и iso-образы). Эта рекомендация дается из соображения упрощения администрирования СХД, упрощения расчета необходимого места, презентирования этого LUN всем серверам.

5.3. Виртуальное оборудование VM

Виртуальная машина – это не что иное, как набор виртуального оборудования. Притом набор, достаточно ограниченный, практически весь представленный на рис. 5.8.

В этом списке не отображаются слоты PCI. В VM с версией виртуального оборудования 7 их порядка 16, что означает: в ограничение по количеству PCI-слотов мы не упрямся. Они могут быть заняты следующими устройствами:

- ❑ один всегда занят под видеоконтроллер;
- ❑ SCSI-контроллерами (до 4 на VM);
- ❑ сетевыми контроллерами (до 10 на VM);
- ❑ если мы импортируем VM, созданную в VMware Workstation, то там еще может быть аудиоконтроллер.

Пойдем по порядку с прочими устройствами и компонентами виртуальной машины.

5.3.1. Memory

Для оперативной памяти мы можем указать размер. Притом здесь мы указываем размер максимальный. В реальности гипервизор может выделять этой VM

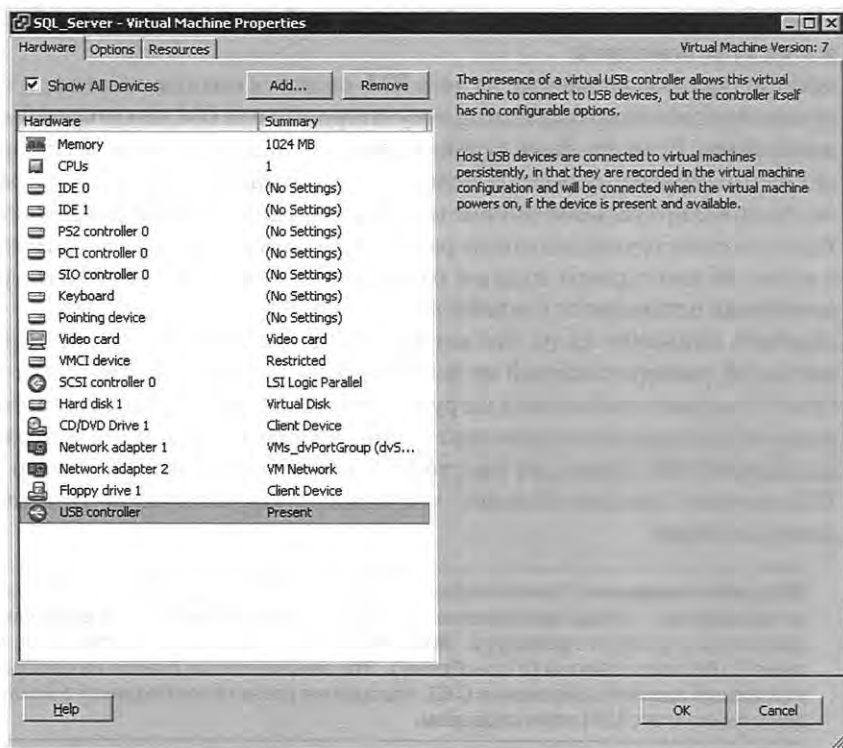


Рис. 5.8. Список комплектующих виртуальной машины

меньше оперативной памяти. Это происходит в ситуациях, когда ВМ просто не использует весь выданный ей объем, и в ситуациях, когда памяти на все ВМ не хватает. Как управлять распределением памяти в таких ситуациях, поговорим позже, в разделе про распределение ресурсов.

Само собой, бессмысленна выдача объема памяти больше, чем поддерживает гостевая ОС. Пользуйтесь подсказками в виде цветных треугольников на экране настройки оперативной памяти для ВМ.

5.3.2. CPUs

В пятой версии ESXi мы можем создавать для ВМ многоядерные виртуальные процессоры. Зайдя в настройки и выделив строку CPUs, мы увидим три величины:

- ❑ **Number of virtual sockets** – столько процессоров увидит гостевая ОС;
- ❑ **Number of cores per socket** – столько ядер в каждом из процессоров увидит гостевая ОС;
- ❑ **Total number of cores** – эта цифра является произведением первых двух.

Важно – производительность процессорной подсистемы виртуальной машины зависит от последнего значения, от **Total number of cores**. Каждое одно вирту-

альное ядро позволяет этой VM использовать одно ядро физическое. Таким образом, сколько всего виртуальных ядер у VM есть – столько физических ядер она сможет задействовать как максимум.

Еще раз – если вы указали для VM четыре процессора с одним ядром каждый, вы получили **Total number of cores** = $4 = 4 \times 1$. Если вы указали использовать один четырехъядерный виртуальный процессор, то **Total number of cores** = $4 = 1 \times 4$. Производительность этих двух случаев абсолютно одинакова.

Если производительность одинакова, зачем вообще эта настройка? Ответ прост – для преодоления технических или лицензионных ограничений гостевых ОС и приложений.

Допустим, вы используете Windows Server Standard edition. У этой версии Windows есть ограничение – она не работает более чем с четырьмя процессорами. Получается, хотя vSphere может дать VM до 32 vCPU, только 4 из них будут задействованы самой гостевой ОС.

А вот если мы этой же VM дадим не 32 одноядерных виртуальных процессора, а один 32-ядерный (или два 16-ядерных, или четыре 8-ядерных) – гостевая ОС сможет использовать все 32 потока.

Общее количество виртуальных ядер у одной VM может быть до 8 в любой лицензии vSphere и до 32 в Enterprise Plus. Но число виртуальных ядер у одной VM не может превышать числа физических ядер того сервера, где она работает, – поэтому если у вас сервер с двумя 4-ядерными процессорами (всего 8 ядер), вы не сможете отдать одной VM больше 8 виртуальных ядер вне зависимости от лицензии.

И наоборот – даже если в сервере у вас четыре 12-ядерных процессора, VM с одним виртуальным одноядерным процессором получит производительность только одного физического ядра.

Еще данная настройка может оказать влияние на лицензирование гостевых ОС или ПО. Подробности следует уточнять по документации поставщика ПО.

Еще подробности про работу процессора и перераспределение ресурсов читайте в разделе про распределение ресурсов.

5.3.3. IDE, PS2 controller, PCI controller, SIO controller, Keyboard, Pointing device

Эти устройства присутствуют всегда, никакая настройка для них невозможна. Обратите внимание: контроллеров IDE два, на каждом может висеть по два устройства. Таким образом, CD-ROM плюс HDD на IDE-контроллерах вместе – может быть не более четырех на одну VM.

5.3.4. Video card

Для видеоконтроллера мы можем настраивать максимальное количество дисплеев, подключаемое к VM, и объем зарезервированной памяти под видео. Видеоконтроллер для VM на ESXi – это всегда простой SVGA-видеоконтроллер. След-

ствием является то, что требовательные к видеокарте приложения (например, AutoCad) – не самые лучшие кандидаты на виртуализацию сегодня.

Однако базовые генераторы 3D-нагрузки, такие как интерфейс Aero (актуально при внедрении инфраструктуры виртуальных рабочих мест, VDI) уже нормально работают на виртуальном видеоконтроллере.

Возможность задать несколько мониторов (а для этого увеличить размер видеопамяти) может пригодиться в основном в VDI-решениях.

5.3.5. VMCI device, VM Communication Interface

Контроллер VMCI – это, по сути, специализированный сетевой контроллер, осуществляющий чрезвычайно быструю связь между ВМ на одном сервере и между гипервизором и ВМ. Виртуальный сетевой контроллер в силу своей виртуальности способен дать порядка 2 или 10 Гбит/сек скорости обмена трафика между ВМ на одном сервере. Интерфейс VMCI – порядка до 10 Гбит/сек, а по некоторым данным до 40 Гбит.

Задействование этого интерфейса должно быть реализовано на уровне ПО. VMware предоставляет соответствующие VMCI Socket API и документацию.

Де-факто предыдущий абзац означает, что это устройство вам не пригодится, – мне не встречалось ни единого ПО, для которого была бы реализована поддержка VMCI.

5.3.6. Floppy drive

К ВМ может быть подключено до двух флоппи-дисководов. Из настроек мы можем указать, на что ссылается этот виртуальный FDD. Варианты следующие:

- ☐ **Client Device** – то есть физический FDD на машине, с которой вы подключились к этой ВМ с помощью клиента vSphere или веб-консоли;
- ☐ **Host Device** – физический FDD на сервере;
- ☐ **Existing floppy image** – образ flp, доступный на каком-то из хранилищ или в каталоге /vmimages на файловой системе ESXi;
- ☐ **New floppy image** – на указанном хранилище создается пустой образ flp;
- ☐ также кликнув на иконку в верхней панели клиента vSphere (или открытой в отдельном окне консоли ВМ), вы можете подключить к FDD виртуальной машины образ с локального диска компьютера, откуда запущен клиент vSphere.

Не забывайте ставить флажки **Connected** и **Connect at power on**, когда хотите воспользоваться виртуальной дискетой. Если **Connected** не стоит, то виртуальный дисковод не работает. В случае подключения **Client device** флажок **Connected** можно ставить только после включения ВМ.

Обратите внимание, что после живой миграции виртуальной машины с FDD снимается флажок **Connected**.

5.3.7. CD/DVD Drive

К ВМ на ESXi может быть подключено до 4 виртуальных CD/DVD-ROM. Ссылаться они могут на:

- ☐ **Client Device** – то есть физический CD/DVD-ROM на машине, с которой вы подключились к этой ВМ с помощью клиента vSphere или веб-консоли;
- ☐ **Host Device** – физический CD/DVD-ROM на сервере;
- ☐ **Datastore ISO File** – образ iso, доступный на каком-то из хранилищ;
- ☐ также кликнув на иконку в верхней панели клиента vSphere (или открытой в отдельном окне консоли ВМ), вы можете подключить к DVD виртуальной машины образ с локального диска компьютера, откуда запущен клиент vSphere.

Не забывайте ставить флажки **Connected** и **Connect at power on**, когда хотите воспользоваться виртуальным DVD-ROM. Если **Connected** не стоит, он не работает. В случае подключения **Client device** флажок **Connected** можно ставить только после включения ВМ.

5.3.8. Network Adapter

Один из самых многовариантных компонентов виртуальной машины – это сетевой контроллер. При использовании последней, 8-ой версии виртуального оборудования их может быть до 10 на одну ВМ. Эти контроллеры могут быть разных типов:

- ☐ **vlance** – виртуальный сетевой контроллер этого типа эмулирует контроллер AMD 79C970 PCnet32 LANCE, старый 10 Мбит/с сетевой контроллер. Его плюсом является наличие драйверов для него в разнообразных, в том числе старых, ОС;
- ☐ **VMXNET** – виртуальный сетевой контроллер этого типа является более производительным гигабитным сетевым контроллером. Его использование возможно после установки драйверов для него в составе VMware tools;
- ☐ **Flexible** – при создании ВМ на ESXi 5 вы увидите скорее этот тип виртуального сетевого контроллера, нежели vlance или vmxnet. Это обусловлено тем, что контроллер типа Flexible как раз и эмулирует или vlance, или vmxnet, в зависимости от того, какой драйвер активен в гостевой ОС;
- ☐ **E1000/E1000e** – виртуальный сетевой контроллер этого типа эмулирует сетевой контроллер Intel 82545EM/82574L Gigabit Ethernet. Драйверы для него доступны под большинство современных ОС, и это основное его преимущество. При использовании этого типа виртуального сетевого контроллера для некоторых операционных систем мы можем воспользоваться сетевыми драйверами от Intel и задействовать их стандартные возможности по настройке NIC Teaming и VLAN изнутри гостевой ОС. Некоторые подробности об их настройке приведены в разделе, посвященном сетям;
- ☐ **VMXNET 2 (Enhanced)** – виртуальный сетевой контроллер этого типа является эволюцией контроллера типа VMXNET. Драйверы для него есть

для многих современных ОС (в составе VMware Tools). Он поддерживает VLAN, TCP Segmentation offload. Некоторые подробности об их настройке приведены в разделе, посвященном сетям;

- ❑ **VMXNET 3** – виртуальный сетевой контроллер этого типа является самым новым на сегодня поколением паравиртуализованных виртуальных сетевых контроллеров. Это означает, что он поддерживает все функции, доступные VMXNET 2, а также некоторые новые. Например, это поддержка multiqueue (также известная в Windows как Receive Side Scaling), IPv6 offloads, VLAN off-loading и MSI/MSI-X interrupt delivery. Говоря проще, VMXNET 3 работает быстрее, с меньшими накладными расходами и поддерживает многие актуальные на сегодня сетевые функции. Однако для ВМ с этим типом сетевого контроллера не работает VMware Fault Tolerance.

Драйвер для VMXNET 3 доступен для ОС:

- ❑ 32 и 64-битных версий Microsoft Windows XP и более поздних;
- ❑ 32 и 64-битных версий Red Hat Enterprise Linux 5.0 и более поздних;
- ❑ 32 и 64-битных версий SUSE Linux Enterprise Server 10 и более поздних;
- ❑ 32 и 64-битных версий Asianux 3 и более поздних;
- ❑ 32 и 64-битных версий Debian 4/Ubuntu 7.04 и более поздних;
- ❑ 32 и 64-битных версий Sun Solaris 10 U4 и более поздних.

Прочие различия виртуальных сетевых контроллеров перечислены в табл. 5.1.

Таблица 5.1. Функции виртуальных сетевых контроллеров разных типов

	Flexible	Enhanced vmxnet (vmxnet2)	E1000	VMXNET 3
IPv4 TSO	Нет	Да	Да	Да
IPv6 TSO	Нет	Нет	Нет	Да
Jumbo Frames	Нет	Да	Нет	Да
Large Ring Sizes	Нет	Нет	Да	Да
RSS	Нет	Нет	Нет	Да
MSI-X	Нет	Нет	Нет	Да

Выбрать тип контроллера можно при создании ВМ (в мастере **Custom**) или при добавлении в нее контроллера позднее. Если необходимо поменять тип существующего контроллера, то нужно или удалить старый и добавить новый, или напрямую править файл настроек (*.vmx).

В файле настроек ВМ могут быть строки следующего вида:

- ❑ ethernetX.virtualDev = "e1000" для сетевого контроллера типа e1000;
- ❑ ethernetX.virtualDev = "vmxnet" для сетевого контроллера типа VMXNET 2 (Enhanced);
- ❑ ethernetX.virtualDev = "vmxnet3" для сетевого контроллера типа VMXNET 3.

В строке ethernetX «X» – это порядковый номер сетевого контроллера в данной ВМ.

Если поменять тип сетевого контроллера для ВМ с уже установленной ОС, то с точки зрения этой ОС поменяется сетевой контроллер. Это повлечет за собой

сброс настроек IP и, иногда, невозможность выставления настроек, аналогичных предыдущим, – так как они числятся за старым, отключенным, но не удаленным с точки зрения гостевой ОС сетевым контроллером.

Для сохранения настроек IP в Windows можно сделать так:

```
netsh interface ip dump > c:\ipconfig.txt
```

С точки зрения Windows, новый контроллер будет виден под новым именем, вида «Local Area Connection 2» или подобным. В таком случае в полученном текстовом файле следует поменять название подключения на это новое. Или поменять имя сетевого адаптера на старое.

Для импорта настроек воспользуйтесь командой

```
netsh -c interface -f c:\ipconfig.txt
```

Для удаления упоминаний о старых сетевых контроллерах воспользуйтесь менеджером устройств (**Device Manager**), поставив в меню **View** флажок **Show hidden devices**. Кроме того, выполните команду

```
set devmgr_show_nonpresent_devices=1
```

Подробности см. в статье базы знаний Майкрософт № 269155 (<http://support.microsoft.com/?kbid=269155>).

TSO

TCP Segmentation Offloading – функция физического сетевого контроллера, то есть для использования TSO вам нужны поддерживающие это сетевушки. Суть функции заключается в том, что работа по формированию IP-пакетов выполняется сетевым контроллером. То есть операционная система (или процессор) посылает на сетевушку большой блок данных. А контроллер сам формирует из него IP-пакеты, пригодные для передачи по сети. Контроллер выполняет эту работу без нагрузки на процессоры сервера.

Для задействования TCP Segmentation Offloading убедитесь, что физические сетевые контроллеры его поддерживают, а для ВМ выберите тип виртуального сетевого контроллера с поддержкой TSO. Затем в свойствах драйвера включите его использование (рис. 5.9).

Jumbo Frames

Jumbo Frames позволяет увеличить размер поля для данных в IP-пакете. Получается, мы тем же числом пакетов передаем больше килобит. То есть мы с тем же количеством накладных расходов передаем больше полезной информации. Если в стандартном IP-пакете поле для данных размером 1500 байт, то при использовании Jumbo Frame – обычно максимальные 9000 байт.

Jumbo Frames должен поддерживаться всей сетью end-to-end, то есть должны быть включены на физических коммутаторах, виртуальных коммутаторах и в ВМ. Jumbo Frames может использоваться с сетевыми контроллерами 1 Гбит и 10 Гбит.

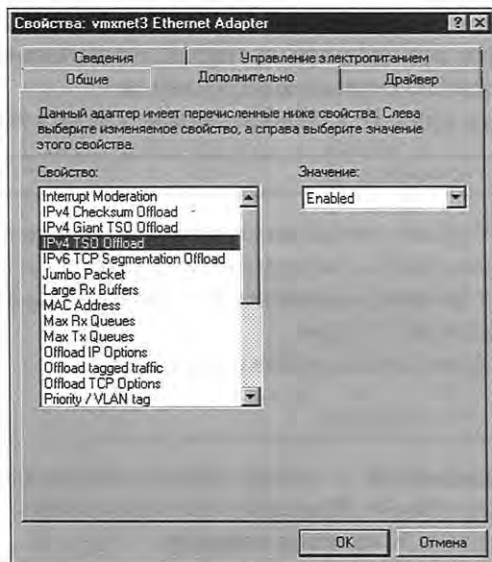


Рис. 5.9. Включение TSO в Windows

Jumbo Frames может использоваться виртуальными машинами и портами VMkernel для трафика NFS, iSCSI и VMotion. Для начала использования Jumbo Frames нам необходимо включить их поддержку на vSwitch/dvSwitch, а затем настроить их использование внутри ВМ или для интерфейса VMkernel.

Для того чтобы включить Jumbo Frames для ВМ, выберите поддерживающий их тип виртуального сетевого контроллера (см. табл. 5.1). После этого на примере Windows (рис. 5.10):

1. Зайдите в настройки драйвера сетевого контроллера.
2. Найдите настройку Jumbo Frames и укажите значение 9000.

Для теста Jumbo Frames выполните ping:

```
ping -f -l 8972 <IP какой-нибудь другой машины>
```

Ключ -f запрещает фрагментацию пакетов. Если с этим ключом ответа не будет – значит, какое-то устройство в сети не пропускает большие пакеты без фрагментации.

Large Ring Sizes

Большой буфер на сетевом контроллере, который позволяет обрабатывать большие всплески трафика без отбрасывания пакетов. Rx буфер равен 150 для VMXNET2 и 256 для VMXNET3. Это позволяет VMXNET3 меньше загружать процессоры сервера в случае гигабитного Ethernet и демонстрировать лучшую производительность для 10-гигабитного Ethernet.

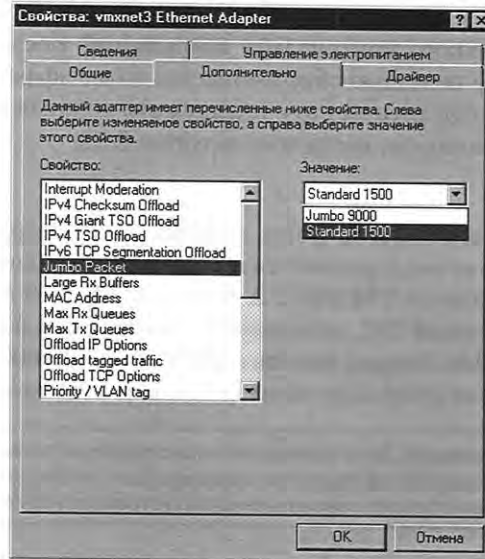


Рис. 5.10. Пример настройки Jumbo Frames для Windows

RSS

Технология Receive-Side Scaling реализует многопоточность обработки стека TCP/IP. Пакеты даже одного сетевого контроллера могут обрабатываться одновременно на нескольких процессорах сервера или виртуальной машины. RSS поддерживается Windows Server 2003 SP2, Windows Server 2008. Включение этой функции должно быть выполнено и в свойствах драйвера виртуального сетевого контроллера, и в операционной системе.

Первое, что стоит проверить, – включена ли эта функция для ОС. В случае Windows Server 2008 выполните команду

```
netsh int tcp show global
```

В выводе вас интересует строка

```
Receive-Side Scaling State is enabled
```

Значение enabled и означает, что RSS включен.

Следующий шаг – зайти в Диспетчер устройств (**Device Manager**) ⇒ зайти в свойства сетевого контроллера ⇒ вкладка **Advanced** ⇒ настройку **RSS** в значение **enabled**.

MSI-X

Message Signaled Interrupts – альтернативная форма прерываний: вместо присваивания номера запроса на прерывание устройству разрешается записывать

сообщение по определенному адресу системной памяти, на деле отображенному на аппаратуру локального контроллера прерываний (local APIC) процессора. Попросту говоря, с ее помощью обеспечивается более эффективная работа сети. ОС от Майкрософт поддерживает эту технологию, начиная с Windows Vista. Основная выгода будет в случае, когда используется RSS.

Резюме

Подводя итоги разговора о типах сетевых контроллеров ВМ: используйте VMXNET 3 там, где это поддерживается гостевыми ОС. Используйте VMXNET 2 там, где не поддерживается VMXNET 3. Если VMXNET 3 и VMXNET 2 не поддерживаются для гостевой ОС, используйте E1000. Если и он не поддерживается, используйте Flexible. Однако иногда с Flexible или E1000 стоит начать, чтобы у ВМ был доступ к сети сразу, а не после установки VMware tools.

Обратите внимание. После смены типа контроллера гостевая ОС будет считать, что ей поменяли контроллер. Зачастую это приводит к необходимости удалить упоминание о старом контроллере через менеджер устройств (Device manager).

В 8-ой версии виртуального оборудования виртуальные сетевые контроллеры числятся USB-устройствами. В частности, это означает, что их можно отключить от ВМ через стандартный механизм отключения USB-устройств или запустив оснастку VMware tools (рис. 5.11).

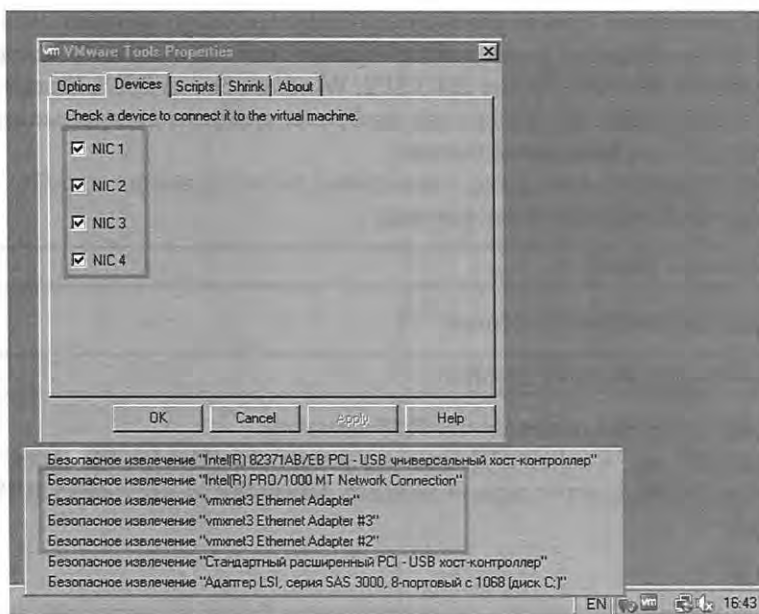


Рис. 5.11. Возможность отключения сетевых контроллеров

В некоторых ситуациях это недопустимо, например на терминальном сервере. Для отключения данной возможности добавьте в файл настроек (*.vmx) этой ВМ строку

```
devices.hotplug = "false"
isolation.device.connectable.disable = "true"
isolation.device.edit.disable = "true"
```

Напомню, что сделать это возможно, пройдя в свойства выключенной ВМ: **Edit Settings** ⇒ вкладка **Options** ⇒ **Advanced** ⇒ **General** ⇒ **Configuration Parameters** ⇒ кнопка **Add Row**.

Первая из этих настроек запретит горячее удаление или добавление любых устройств в эту ВМ, а две другие запретят отключение устройств через VMware tools.

MAC-адреса виртуальных машин

Они генерируются автоматически. Сгенерироваться заново MAC-адрес для той же ВМ может после перемещения ее файлов. При генерации MAC-адреса проверяется несовпадение его с MAC-адресами работающих ВМ и ВМ в состоянии паузы (suspend). Также делается проверка на уникальность MAC-адреса при включении ВМ.

Из этого вытекает следующее: при включении ВМ она может поменять свой MAC, если он совпадает с MAC-адресом какой-то из работающих ВМ. Последовательность такая: создали ВМ1, ей сгенерировался MAC-адрес. Выключили ВМ1. Создали ВМ2 – ей сгенерировался такой же MAC-адрес (вероятность этого крайне мала, но все же). При его генерации проверки на совпадение с ВМ1 не происходит, так как та выключена. При включении ВМ1 меняет свой MAC-адрес.

Как правило, все хорошо. Однако механизм генерации имеет ограничение – до 256 уникальных MAC-адресов (то есть виртуальных сетевых контроллеров) на сервер. Если это количество превышено, может потребоваться ручная настройка MAC-адреса для ВМ.

Эта проблема потенциально актуальна при отсутствии vCenter. Если вы работаете через vCenter, то за генерацию и проверку уникальности MAC-адресов отвечает уже он. Формат MAC-адреса выглядит следующим образом: 00:50:56:vCenterID:xx:xx, где **vCenterID** можно посмотреть через клиент vSphere в меню **Home** ⇒ **vCenter Server Settings** ⇒ **Runtime Settings**. Таким образом, vCenter способен сгенерировать до 65 535 уникальных MAC-адресов. При старте ВМ vCenter обязательно убеждается в отсутствии совпадения MAC-адресов у разных ВМ.

За VMware закреплены два диапазона MAC-адресов – один для автоматически генерируемых и один для вручную присваиваемых. Для вручную задаваемых MAC-адресов VMware использует диапазон 00:50:56. Таким образом, если вы задаете MAC-адрес вручную, допустимым является оный из диапазона 00:50:56:00:00:00-00:50:56:3F:FF:FF.

Поле для указания MAC-адреса вручную вы увидите в свойствах виртуальной машины, выделив ее сетевой контроллер. В этом поле может быть указан MAC-адрес только из диапазона MAC-адресов VMware.

Можно задать и совсем произвольный MAC-адрес, уже внутри файла настроек (*.vmtx), вручную. А также изнутри гостевой ОС. Однако в случае не VMware-диапазона проблема уникальности MAC-адреса в вашей сети – ваша головная боль.

Для задания произвольного MAC-адреса в файле настроек сделайте следующее.

Каким-либо способом откройте файл настроек VM в текстовом редакторе. Например, подключившись к ESXi по SSH и выполнив команду

```
vi /vmfs/volumes/<название хранилища>/<каталог с VM>/<файл настроек VM.vmtx>
```

Найдите строки вида

```
ethernet0.generatedAddress = "00:50:56:be:2c:21"
ethernet0.addressType = "vpx"
```

(на примере первого из сетевых контроллеров этой VM).

Замените их строками

```
ethernet0.checkMACAddress = "false"
ethernet0.addressType = "static"
ethernet0.Address = "00:0C:29:B0:27:E1"
```

Разумеется, MAC-адрес укажите желаемый.

5.3.9. SCSI controller

Виртуальный контроллер SCSI тоже бывает разных типов, и для него доступны кое-какие настройки. Сначала про типы. Если вы зайдете в свойства VM и на вкладке **Hardware** выделите SCSI-контроллер, в верхней правой части окна будет кнопка **Change Type** (рис. 5.12).

Типов всего четыре:

- ☐ **BusLogic Parallel** – этот виртуальный дисковый контроллер работает наименее эффективным способом, с большими, чем другие, накладными расходами. Однако для него есть драйверы для большого количества операционных систем;
- ☐ **LSI Logic Parallel** – работает с меньшими накладными расходами, чем BusLogic;
- ☐ **LSI Logic SAS** – новая версия LSI Logic. Отличается тем, что поддерживает протокол SCSI версии 3.

Используется в двух случаях:

- для VM с современными ОС (такими как Windows 2008) вместо старой версии контроллера LSI;
- для VM, которым необходима поддержка протокола SCSI 3. Главным образом для виртуальных машин – узлов кластера Microsoft Failover

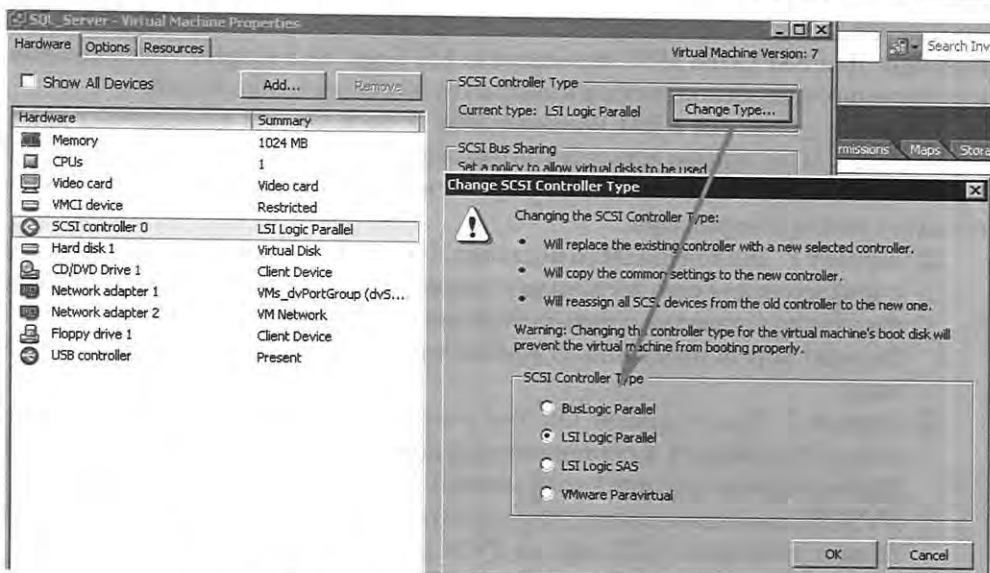


Рис. 5.12. Смена типа виртуального SCSI-контроллера

Cluster, для общего диска (в том смысле, что даже для узлов такого кластера системный диск может быть подключен к контроллеру другого типа);

❑ **VMware Paravirtual SCSI (PVSCSI)** – самая современная версия виртуального дискового контроллера. Обеспечивает наибольшую производительность и наименьшие накладные расходы. Однако не работает VMware Fault Tolerance для ВМ с этим контроллером, и список ОС, для которых есть драйверы, ограничен:

- Windows Server 2008 (включая R2);
- Windows Server 2003;
- Red Hat Enterprise Linux (RHEL) 5 и 6;
- Windows 7/Vista/XP;
- SUSE 11 SP1;
- дистрибутивы Linux с ядром 2.6.33 и более поздними версиями, включающие драйвер `vmw_pvscsi`;
- актуальный список см. в базе знаний, <http://kb.vmware.com/kb/1010398>.

Напомню, что «паравиртуализованный» виртуальный контроллер означает, что при его работе задействуются мощности контроллера физического напрямую, без какой-то эмуляции или перехвата со стороны гипервизора. Именно в этом кроется повышение эффективности паравиртуализованных дисковых и сетевых контроллеров.

Резюме. Если позволяют условия, используйте контроллер типа VMware Paravirtual. Если он не поддерживается гостевой ОС, используйте LSI Logic Parallel или LSI SAS (ориентироваться можно на то, который из них предлагается по

умолчанию для данной гостевой ОС). Если и LSI не поддерживается, используйте BusLogic. Для VM-узлов MFC используйте LSI Logic SAS вторым контроллером, для системного диска контроллер выбирается из общих соображений.

Пример – установка в VM Windows 2008. У этой ОС нет стандартных драйверов для PVSCSI, однако есть для контроллера LSI. Но на ESXi есть образ FDD с драйверами для PVSCSI. И теперь у вас есть несколько вариантов того, как поступить с типом контроллера:

- ☐ вариант 1. Тип контроллера поставить PVSCSI, подключить к VM образ fip с драйвером для него и в начале установки подложить эти драйверы. После окончания установки оставить этот тип контроллера.
Вариант хорош простотой. В данном случае его можно назвать оптимальным;
- ☐ вариант 2. При установке ОС тип контроллера оставить LSI. После установки ОС добавить в систему второй контроллер типа PVSCSI, включить VM (на горячую добавить тоже можно). Windows активирует драйверы для PVSCSI. Затем выключить VM, удалить второй контроллер (PVSCSI), тип первого поменять с LSI Logic на PVSCSI.
Вариант не очень удобен большим количеством шагов, зато нет нужды в подкладывании драйверов;
- ☐ наконец, драйверы можно интегрировать в дистрибутив – но их необходимо где-то взять. Например, из упомянутого образа FDD. Как вариант – из iso с VMware tools.

Образы дисков с драйверами для PVSCSI (Windows XP/Vista/7/2003/2008) и Bus Logic (Windows XP) доступны в каталоге vmimages. Зайдите в свойства VM, выделите **Floppy Drive** ⇒ **Use existing floppy image in datastore** ⇒ каталог **vmimages/floppies**.

Еще у виртуального SCSI-контроллера есть настройка **Bus sharing** – совместный доступ к SCSI-шине. Она нужна в ситуациях, когда вы хотите дать одновременный доступ к одним и тем же виртуальным дискам нескольким VM. Обычно это необходимо для построения кластеров, таких как MSCS/MFC. Варианты этой настройки:

- ☐ **None** – совместного доступа нет. Значение настройки по умолчанию;
- ☐ **Virtual** – к виртуальным дискам, висящим на этом контроллере, возможен доступ с других VM на этом же сервере. Такая организация кластера, когда узлы на одном ESXi, называется cluster-in-a-box;
- ☐ **Physical** – к виртуальным дискам, висящим на этом контроллере, возможен доступ с других VM, в том числе с других серверов. Настройка, нужная для организации cluster-across-boxes (когда узлы на разных ESXi) и physical-to-virtual.

Добавление контроллера. Если вы нажмете кнопку **Add** на вкладке **Hardware** в свойствах VM, то увидите список виртуальных компонентов, которые в VM можно добавить. Однако среди них нет SCSI-контроллера. Если вам надо добавить SCSI-контроллер, то делается это так.

Все-таки идем в мастер добавления виртуального оборудования **Edit Settings** ⇒ **Hardware** ⇒ **Add**. Но добавляем **Hard Drive**.

Проходим по мастеру создания жесткого диска. О подробностях – чуть ниже, сейчас нас интересует пункт «SCSI node». У каждого виртуального диска есть адрес вида «SCSI X:Y». Последняя цифра адреса – это номер диска на SCSI-шине, SCSI id. А первая цифра – номер контроллера. Таким образом, первый, дефолтный диск ВМ создается по адресу SCSI 0:0, то есть это первый диск на первом контроллере. Если для второго диска вы выберете адрес SCSI 1:0, то кроме диска у вас добавится и второй контроллер. Если для третьего диска выбрать SCSI 2:0, то добавится третий контроллер. Всего до четырех.

Справедливости ради надо добавить: если вы работаете через веб-интерфейс, то там SCSI-контроллер присутствует как отдельное устройство для добавления.

Добавлять несколько контроллеров вам придется преимущественно для ВМ – узлов отказоустойчивых кластеров, таких как Microsoft Cluster Services или Microsoft Failover Cluster. Для этих решений требуется, чтобы загрузочный диск и диски с данными висели на разных контроллерах.

5.3.10. Hard Disk

Виртуальным жестким дискам в виде файлов vmdk или LUN целиком посвящен весь следующий раздел.

5.3.11. Parallel port

К ВМ можно подключить параллельный (LPT) порт. Возможно подключение к ВМ физического LPT-порта сервера. Обратите внимание, что при подключенном физическом LPT-порту невозможна живая миграция (VMotion) этой ВМ и не будет работать FT.

Также виртуальный порт LPT может быть подключен к файлу.

5.3.12. Serial port

На ESXi возможно подключение к ВМ физического COM-порта сервера. Обратите внимание, что при подключенном физическом COM-порту невозможна живая миграция (VMotion) этой ВМ и не будет работать FT.

Альтернативой пробросу физического порта является подключение к именованному каналу (named pipe) для связи по COM-порту между ВМ одного ESXi.

Последовательный порт виртуальной машины может быть подключен к файлу, то есть записывать в файл исходящий поток данных.

Наконец, существуют сторонние решения Com-over-IP, они представляют собой программное решение, работающее изнутри гостевых ОС. Здесь они упомянуты для справки, так как работают абсолютно независимо от vSphere. Однако если необходима массовая работа с последовательными интерфейсами – этот вариант часто оказывается самым удобным, пусть и требует дополнительного ПО.

В актуальной версии vSphere есть еще один вариант подключения – концентратор виртуальных последовательных портов, пункт **Connect via Network** в свойствах виртуального последовательного порта.

Суть этого механизма – в том, что ESXi может подключить последовательный порт виртуальной машины к внешнему концентратору последовательных портов по сети.

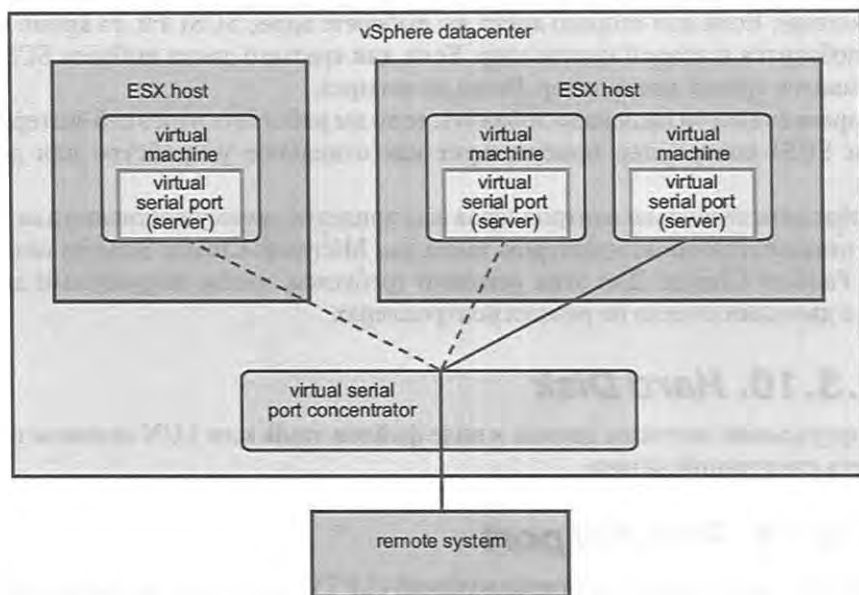


Рис. 5.13. Схема работы с концентратором последовательных портов
Источник: VMware

Например, таким внешним концентратором может быть виртуальная машина ACS v6000 Virtual Advanced Console Server. Администраторы обращаются (по telnet или ssh) к этой ВМ и получают доступ к последовательным интерфейсам виртуальных машин vSphere.

Для того чтобы воспользоваться данной возможностью, необходимо настроить внешний модуль, а затем указать его IP-адрес в настройках последовательного порта виртуальной машины.

Обратите внимание: обращение по сети на адрес внешнего модуля будет производиться сервером ESXi, не виртуальной машиной. Для виртуальных машин с таким вариантом последовательного порта будет поддерживаться vMotion.

5.3.13. SCSI device

К ВМ можно подключить SCSI-устройство, не являющееся жестким диском (LUN). Обычно такими устройствами являются стримеры, ленточные библиоте-

ки. Притом можно к ВМ подключить даже ленточный накопитель, подключенный в Fibre Channel сеть.

Обратите внимание на то, что VMware не поддерживает подключения ленточных накопителей к виртуальным машинам, и для значительного процента моделей такое подключение не работает.

Так же как SCSI device, можно из ВМ увидеть контроллер системы хранения (именно системы хранения, не HBA-сервера), если того требует ПО управления SAN, например HP Command View.

Некоторые (возможно, все) USB-CDROM следует подключать к ВМ не как устройства USB, а как SCSI Device.

5.3.14. USB controller и USB Device

Начиная с версии 4.1, к ВМ стало возможно подключить USB-устройство сервера, а начиная с версии 5 – USB-устройство с клиента.

Если вы хотите подключать к виртуальной машине физические устройства USB, то в конфигурацию этой ВМ следует предварительно добавить USB-контроллер (для многих ОС это возможно на горячую).

В пятой версии ESXi контроллер USB доступен в двух вариантах:

- ☐ **EHCI+UHCI** – поддержка USB версий 1.1 и 2. Работает со всеми или с подавляющим большинством гостевых ОС;
- ☐ **XHCI** – поддержка USB версий 1.1, 2 и 3. Однако на момент написания драйвер для этого контроллера был доступен только для некоторых версий Linux (ядро версии не ниже, чем 2.6.35). Кроме того, проброс устройств USB 3.0 возможен только с клиента vSphere, не с хостов ESXi.

Когда в конфигурации ВМ есть контроллер USB, вы можете:

- ☐ подключить устройство USB к машине, где запущен ваш клиент vSphere, и при помощи соответствующей пиктограммы подключить это устройство к ВМ. ВМ должна быть включена. Подключение будет разорвано при закрытии клиента vSphere;
- ☐ подключить USB-устройство к серверу ESXi (именно к тому, где сейчас работает эта ВМ) и подключить его в ВМ **Edit Settings** ⇒ вкладка **Hardware** ⇒ **Add** ⇒ **USB Device**. На шаге выбора устройства для подключения вы можете поставить флажок **Support vMotion while device is connected** – в этом случае данное USB-устройство останется подключенным к ВМ, даже если она была мигрирована на другой сервер ESXi.

Что очень здорово – vMotion виртуальной машины с подключенным USB-устройством будет возможен, во-первых, и не прервет работу с устройством, во-вторых. Таким образом, ситуация, когда виртуальная машина работает на сервере 2, но обращается к USB-ключу с сервера 1, нормальна и работоспособна. В документации вам доступен список официально поддерживаемых устройств USB (<http://kb.vmware.com/kb/1021345>).

К одному виртуальному контроллеру USB не может быть подключено более 15 устройств USB, но все 15 могут быть подключены к одной ВМ (некоторые

устройства USB являются составными, таких можно подключить, соответственно, меньше). Всего к ВМ можно подключить до 20 USB-устройств (если в ней два контроллера USB). Нельзя разделить одно устройство USB между несколькими виртуальными машинами.

В конфигурацию ВМ может быть добавлен только один контроллер USB каждого типа (EHCI+UHCI/xHCI).

Обратите внимание. USB-устройства следует отключить перед горячим добавлением в виртуальную машину процессоров, памяти или PCI-устройств. Иначе устройства USB окажутся отключены автоматически.

Однако, на мой взгляд, как и раньше, самый удобный способ подключения к ВМ USB-устройств – это использование решений USB-over-IP. В таком случае плановый или неплановый простой сервера ESXi, к которому подключено устройство USB, не вызовет недоступности этого USB-устройства для виртуальной машины и не потребует от администраторов действий для обеспечения этой доступности. См. информацию по ссылке <http://link.vm4.ru/usb>.

5.3.15. VMDirectPath

Эта функция позволяет «пробросить» в виртуальную машину PCI(e)-устройство.

Сервер должен поддерживать Intel Virtualization Technology for Directed I/O (VT-d) или AMD IP Virtualization Technology (IOMMU), и в BIOS это должно быть включено.

Изначально эта функция позиционируется как способ пробрасывать в ВМ высокоскоростные контроллеры ввода-вывода, такие как 10 Гбит Ethernet и FC HBA. Хорошо для абсолютной минимизации задержек и предотвращения совместного использования устройств.

Однако можно попытаться пробросить и другие устройства, такие как USB-контроллеры сервера. К одной ВМ – до двух устройств. Однако у VMware есть явный список поддерживаемых контроллеров. К слову, на момент написания в список входят контроллеры:

- ☐ Intel 82598 10 Gigabit Ethernet adapter;
- ☐ Broadcom 57710 10 Gigabit Ethernet adapter;
- ☐ QLogic QLA25xx 8Gb Fibre Channel;
- ☐ LSI 3442e-R и 3801e (1068 chip-based) 3Gb SAS adapter.

Напомню, что отсутствие устройств в этом списке не означает, что для них не заработает VMDirectPath.

К сожалению, задействование данной функции приводит к невозможности пользоваться:

- ☐ VMotion и Storage VMotion;
- ☐ Fault Tolerance;
- ☐ снимками состояния (snapshot) и паузой (suspend) для виртуальной машины;

□ горячим добавлением устройств.

Чтобы подключить какое-то устройство к ВМ, сначала необходимо отключить его от гипервизора. Для этого пройдите **Configuration** ⇒ **Advanced Settings** в разделе **Hardware** ⇒ ссылка **Configure Passthrough** (рис. 5.14).

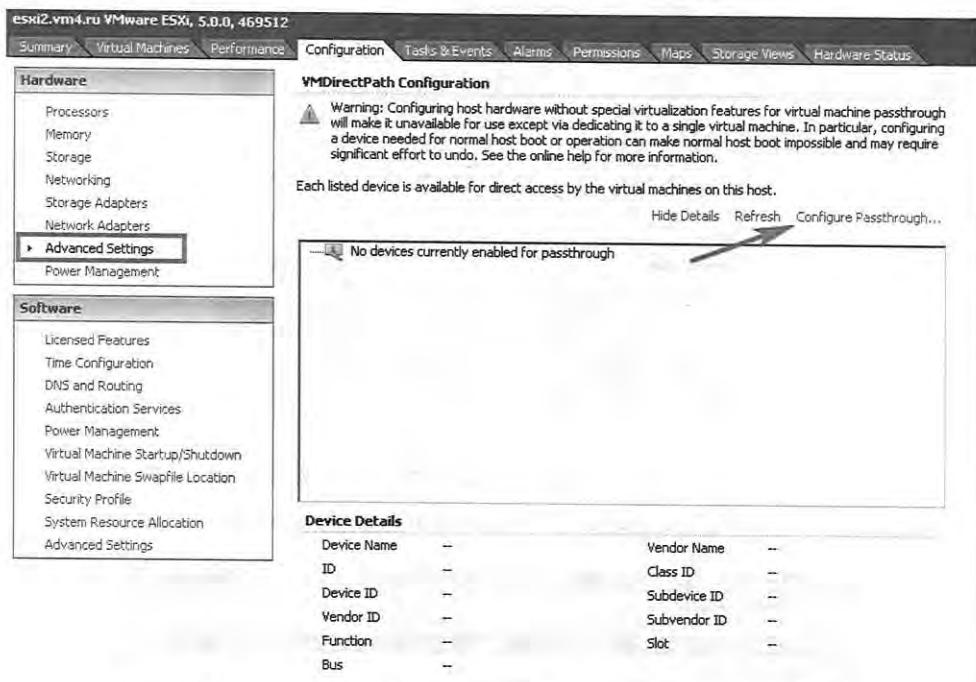


Рис. 5.14. Настройка VMDirectPath для конкретного устройства

Вам покажут список устройства сервера (рис. 5.15).

Здесь мы пометим устройство как не используемое самим гипервизором. Значит, оно теперь может использоваться виртуальной машиной.

Зайдя в свойства ВМ, на вкладке **Hardware** ⇒ **Add** ⇒ **PCI Device** ⇒ в выпадающем меню у вас должно получиться выбрать ранее указанное устройство (рис. 5.16).

После включения гостевая операционная система увидит устройство и запустит поиск и установку для него драйверов.

Ни одна другая виртуальная машина и сам ESXi не смогут теперь этим устройством воспользоваться.

Обратите внимание. Некоторые современные сервера могут обеспечить возможность vMotion виртуальной машины с подключенным через VMDirectPath контроллером. На момент написания как обладающие такой функцией мне известны только блейд-сервера от Cisco. Это реализуется за счет того, что те сетевые контроллеры, которые воспринимаются сервером и гипервизором как физические, на самом деле тоже «виртуальные».

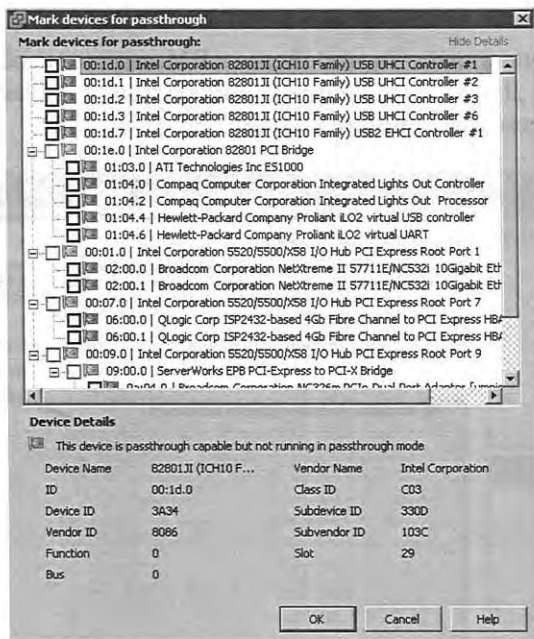


Рис. 5.15. Выбор устройства для проброса в VM

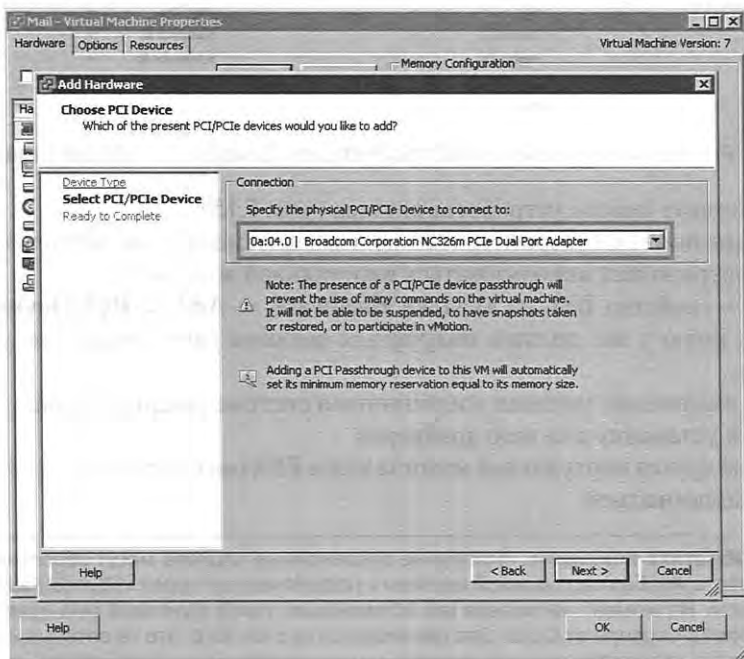


Рис. 5.16. Выбор устройства для подключения к VM

5.4. Все про диски VM

Здесь подробно коснусь того, что из себя могут представлять диски виртуальных машин.

Глобально вариантов здесь два:

1. Диск VM – это файл vmdk на хранилище VMFS или NFS.
2. Диск VM – это весь LUN на FC/iSCSI/локальных дисках. Называется такой тип подключения Raw Device Mapping, RDM.

И в первом, и во втором случае у нас есть несколько вариантов. Поговорим про все варианты подробнее.

5.4.1. Виртуальные диски – файлы vmdk

Про виртуальные диски именно как файлы будет рассказано чуть далее, в разделе «Файлы VM». Сейчас поговорим про различные настройки виртуальных дисков – файлов vmdk.

Для подключения диска к VM зайдите в ее свойства, нажмите кнопку **Add** на вкладке **Hardware** и выберите **Hard Disk**. После нажатия **Next** вы увидите следующие шаги мастера.

1. **Select a Disk** – здесь вы выберете, хотите ли создать новый файл vmdk, подключить уже существующий и расположенный на доступном этому ESXi хранилище, или же подключить RDM. Сейчас рассмотрим первый вариант.
2. **Create a Disk** – здесь вы можете указать следующие настройки:
 - **Disk Size** – номинальный размер диска. Столько места на нем увидит гостевая ОС. Размер же файла vmdk зависит от следующей настройки: напоминаю, что максимальный размер файла ограничен размером блока раздела VMFS, на котором вы его создаете. На VMFS, созданном по умолчанию, вы не создадите один файл vmdk размером больше 256 Гб;
 - **Disk Provisioning** – тип файла vmdk. О типах дисков – чуть ниже;
 - **Location** – на каком хранилище будет находиться создаваемый файл.
3. **Advanced Options** – эти настройки обычно менять не требуется:
 - **Virtual Device Node** – на каком ID какого виртуального контроллера будет располагаться этот виртуальный диск. SCSI (1:2) означает, что этот диск займет второе SCSI ID на виртуальном SCSI-контроллере номер 1 (нумеруются они с нуля). Обратите внимание, если этого контроллера в VM еще нет – он будет добавлен вместе с диском. В VM может быть до 4 SCSI-контроллеров и до 15 дисков на каждом. Также вы можете указать, что создаваемый диск подключен к контроллеру IDE. Для IDE-дисков недоступны некоторые функции, такие как горячее добавление и увеличение размера;
 - **Mode** – если поставить флажок **Independent**, то к этому виртуальному диску не будут применяться снимки состояния (snapshot). В режиме **Persistent** все изменения будут немедленно записываться в этот файл vmdk.

В режиме **Nonpersistent** все изменения с момента включения будут записываться в отдельный файл, который будет удаляться после выключения ВМ. Такой режим имеет смысл, например, для демонстрационных ВМ. Мы их подготовили, настроили, перевели их диски в данный режим. Теперь после выключения они всегда будут возвращаться к своему состоянию на момент включения этого режима.

Файлы vmdk могут быть разных типов, и типы эти следующие:

❑ **Thick Provision Lazy Zeroed** (иногда zeroedthick) – «обнуляемый предразмеченный». Этот режим для диска используется по умолчанию при создании файла vmdk на хранилищах VMFS. В данном режиме место под файл vmdk выделяется в момент создания. То есть если вы создаете для ВМ диск размером 50 Гб, файл vmdk займет на диске 50 Гб даже тогда, когда никаких данных ВМ еще не записала на этот диск. Блоки данных обнуляются (очищаются от данных, которые находились там ранее) перед первым к ним обращением – поэтому первое обращение к ранее свободному месту будет чуть медленнее, чем могло бы быть. Он является рекомендуемым под большинство задач;

❑ **Thick Provision Eager Zeroed** (иногда eagerzeroedthick) – «заранее обнуляемый предразмеченный». В этом режиме место под файл vmdk выделяется в момент создания. Также в момент создания происходит обнуление всех блоков, занимаемых этим файлом. Из-за обнуления процесс создания файла vmdk такого типа занимает намного больше времени, чем создание файла vmdk любого другого типа.

Используйте его для ВМ под защитой Fault Tolerance (при включении FT мастер оповестит о начале преобразования файлов vmdk ВМ к этому типу, даже если изначально вы создали их в другом формате).

Также файлы vmdk этого типа рекомендуется использовать под диск для данных отказоустойчивого кластера Майкрософт. Правда, лишь для варианта кластера, когда оба узла работают на одном ESXi, что бывает лишь в тестовых случаях. Для узлов, разнесенных по разным хостам, в качестве общего диска следует использовать RDM.

Теоретически использование файлов-дисков такого типа может дать прирост производительности – но официальная позиция VMware: «Недолжна». Если наше хранилище поддерживает VAAI, то создание файла такого типа может быть быстрым или даже очень быстрым;

❑ **Thin Provision** – «тонкий». Файлы vmdk этого типа создаются нулевого размера и растут по мере того, как гостевая ОС изменяет данные на этом диске. Отлично подходит для экономии места на хранилищах. Блоки данных обнуляются перед первым к ним обращением. Чуть больше подробностей дам позже.

Реже, а то и никогда, вам встретятся еще и другие типы виртуальных дисков:

❑ **2gbsparse** – файл разбивается на части размером по 2 Гб. Если файлы vmdk ВМ в таком формате, то включить ее на ESXi нельзя. Однако в подобном

формате VM используются в других продуктах VMware. Так что VM в таком формате вам может понадобиться при переносе ее на ESXi с другого продукта VMware или для запуска на другом продукте созданной на ESXi виртуальной машины. Преобразовывать vmdk в формат 2gbsparse или из него в thin/thick вы можете при помощи команды vmkfstools.

Хотя для переноса VM между разными продуктами VMware удобнее использовать продукт VMware Converter;

- ❑ **rdm** и **rdmp** – такой тип у vmdk, которые являются ссылками на LUN, подключенные Raw Device Mapping, RDM.

vRDM – virtual RDM,

pRDM – physical RDM.

Подробнее про RDM – чуть ниже;

- ❑ **monosparse** и **monoflat** – виртуальные диски в этих форматах используются в других продуктах VMware.

Обратите внимание. Понятие и технология thin disk также используются некоторыми аппаратными системами хранения (3Par, NetApp), причем такой «аппаратный thin disk» может быть создан независимо от «программного thin-диска ESXi». В случае поддержки системой хранения thin provisioning созданный thick-диск в thin-режиме СХД займет место на системе хранения только по мере заполнения его действительными данными. Но в книге рассматривается лишь thin provisioning в варианте от VMware.

На ESXi 3.x был еще тип виртуального диска thick – предразмеченный необнуляемый. При работе с ним ESXi не производил обнуления блоков. Однако в ESXi 5 создать файл vmdk в подобном формате нельзя. В графическом интерфейсе и в данной книге под типом «thick» понимается zeroedthick.

Режим thin для виртуального диска обычно используется по умолчанию при создании файла vmdk на хранилищах NFS. Но не всегда, это зависит от настроек на стороне сервера NFS.

По большому счету, в случае использования vmdk выбирать нам надо между thin и zeroedthick. Какие соображения имеет смысл принимать во внимание?

Thin-диски требуют намного меньше места на хранилище при создании и в начале эксплуатации VM. Однако через пяток-другой месяцев разница с предразмеченными дисками может сойти на нет, потому что операции уменьшения (Shrink) виртуального диска в ESXi не предусмотрено. Однако некоторые способы очистить thin-диск от записанных, а впоследствии удаленных данных все-таки существуют, см. раздел «Уменьшение размера виртуального диска».

Так происходит по той причине, что при удалении данных изнутри гостевой ОС происходит только очистка заголовков – ОС помечает какие-то блоки как «их теперь можно использовать». ESXi не может отличить блоки, занятые такими (удаленными, с точки зрения гостевой ОС) данными, от блоков с неудаленными данными.

Плюс к тому некоторые операционные системы (Windows, в частности) для записи новых данных предпочитают использовать изначально пустые блоки, не-

жели занятые ранее удаленными данными. Например, мы создадим ВМ с Windows Server 2008 и с диском в 20 Гб. В момент создания ВМ размер ее файла vmdk равен нулю. После установки ОС – порядка, допустим, 6 Гб. Если теперь скопировать на ее диск файл размером в 2 Гб, удалить его, опять скопировать и удалить и в последний раз скопировать, то:

1. В гостевой ОС мы увидим примерно 8 занятых гигабайт: $6 + 2 - 2 + 2 - 2 + 2$.
2. С точки зрения файлов ВМ, мы увидим, что файл vmdk занимает порядка 12 Гб: $6 + 2 + 2 + 2$.

Вывод: под некоторые задачи, когда данные часто добавляются и удаляются, thin-диски быстро вырастут до номинальных размеров.

В каких ситуациях нам интересно использовать thin-диски? Для производственных ВМ – когда мы хотим сэкономить на дисках в первое время эксплуатации виртуальной инфраструктуры. Поясню свою мысль.

Вот у нас есть задача запустить 30 ВМ, для простоты одинаковых. Допустим, приложению может потребоваться до 50 Гб места за пару лет работы. В первые полгода-год – вряд ли больше 15 Гб. И по статистике 50 Гб начинает использовать лишь небольшая доля таких серверов, в большинстве случаев для этого приложения хватает и 25 Гб. Получается:

- при использовании thick-дисков нам необходимо $1500 \text{ Гб} = 50 \text{ Гб} \times 30 \text{ ВМ}$. Но внутри большей части 50-гигабайтных файлов vmdk будет много свободного места. Скорее всего. Мы это просто предполагаем;
- при использовании thin-дисков мы можем обойтись $450 \text{ Гб} = 15 \text{ Гб} \times 30 \text{ ВМ}$. Через год понадобится от $1000 \text{ Гб} = 25 \text{ Гб} \times 30 \text{ ВМ}$, плюс еще гигабайт 300 для тех, кому среднестатистических 25 Гб все-таки недостаточно. Но! Все эти цифры являются приближением. Их точность зависит от имеющихся у нас данных по использованию места на диске конкретным приложением и оценкам по росту нагрузки в будущем. Если мы ошиблись в расчетах и не успели докупить еще дисков, то место на хранилище закончится, и работа всех (!) виртуальных машин с заполненных хранилищ станет невозможна. Когда хранилище заполняется на 99%, ESXi автоматически переводит все ВМ на этом хранилище в состояние паузы (suspend).

Вывод: использование thin provisioning позволяет в начальный момент обойтись меньшим количеством места на системе хранения, но повышает вероятность столкнуться с неработоспособностью сразу всех ВМ в связи с нехваткой места.

По данным VMware, производительность ВМ с дисковой подсистемой не ухудшается при использовании thin-дисков вместо thick.

И для thin, и для zeroedthick время самого первого обращения к блоку заметно выше, чем для eagerzeroedthick, потому что его еще надо обнулить перед первой записью туда. Если такая задержка может быть неприятной для вашего приложения, используйте eagerzeroedthick-диски. Однако все дальнейшие обращения, кроме самого первого, по скорости одинаковы и для thin, и для разного типа thick-дисков.

Обратите внимание. Диски ВМ мы можем конвертировать в любые форматы. При операциях Storage VMotion (или миграции выключенной ВМ) и Clone нас спросят, хо-

тим ли мы, чтобы диски были толстыми или тонкими (zeroedthick или thin). Также мы можем конвертировать файлы vmdk в эти и прочие форматы из командной строки с помощью консольной утилиты vmkfstools.

Тонкие диски и интерфейс

Выделите виртуальную машину с тонким диском и посмотрите на вкладку **Summary**. Там в разделе **Resources** вы увидите данные по занимаемому месту (рис. 5.17).

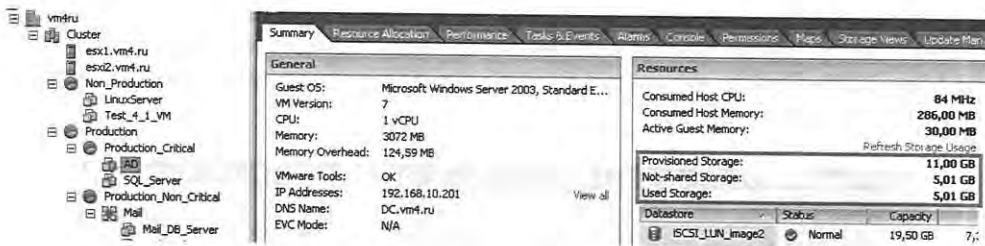


Рис. 5.17. Данные по размеру тонкого диска

- ❑ **Provisioned Storage** – это максимальный объем, который могут занять файлы виртуальной машины. То есть это номинальный объем ее диска плюс объем всех прочих файлов. Из «прочих файлов» заслуживают упоминаний два. Это файл подкачки (*.vswp), который гипервизор создает для этой VM, и файлы vmdk снимков состояния. Так как каждый снимок состояния (snapshot) может занять место, равное номинальному размеру диска, то при каждом снимке состояния величина Provisioned Storage увеличивается на размер диска/дисков;
- ❑ **Not-shared Storage** – сколько места эта виртуальная машина занимает именно под свои файлы, не разделяя их с другими VM;
- ❑ **Used Storage** – сколько места реально занимают на хранилище файлы-диски этой VM.

Not-shared Storage всегда равняется **Used Storage**, за исключением двух вариантов:

- ❑ когда используется функция **Linked Clone**. Она доступна при использовании поверх vSphere таких продуктов, как VMware Lab Manager или VMware View;
- ❑ когда диском VM является RDM и у вас кластер между VM (например, MSCS/MFC). В таком случае RDM выступает в роли общего хранилища, принадлежит сразу двум VM. **Not-shared Storage** будет показывать остальное место, что VM занимает на хранилище, кроме RDM.

Обратите внимание. Provisioned Storage – это ограничение размера файла vmdk. То есть это гипервизор не даст файлу вырасти больше. Однако если на хранилище закончится место, то гипервизор не сможет увеличить thin-диск (или файл снимка состояния), даже если тот не достиг своего максимума. Это приведет к неработоспособности VM.

5.4.2. Изменение размеров дисков VM

Поговорим про разные варианты изменения размеров виртуальных дисков разных типов.

Увеличение размера диска

Если у VM есть файл-диск, то в каком бы он ни был формате, нам может захотеться увеличить его размер, чтобы дать дополнительно место для гостевой ОС.

Делается это просто. Зайдите в свойства VM, выделите диск, который хотите увеличить. Вы увидите меню выбора нового размера и подсказку о максимальном размере диска – он зависит от количества свободного места на текущем хранилище (рис. 5.18). Однако останется вопрос увеличения раздела файловой системы гостевой ОС на этом диске.

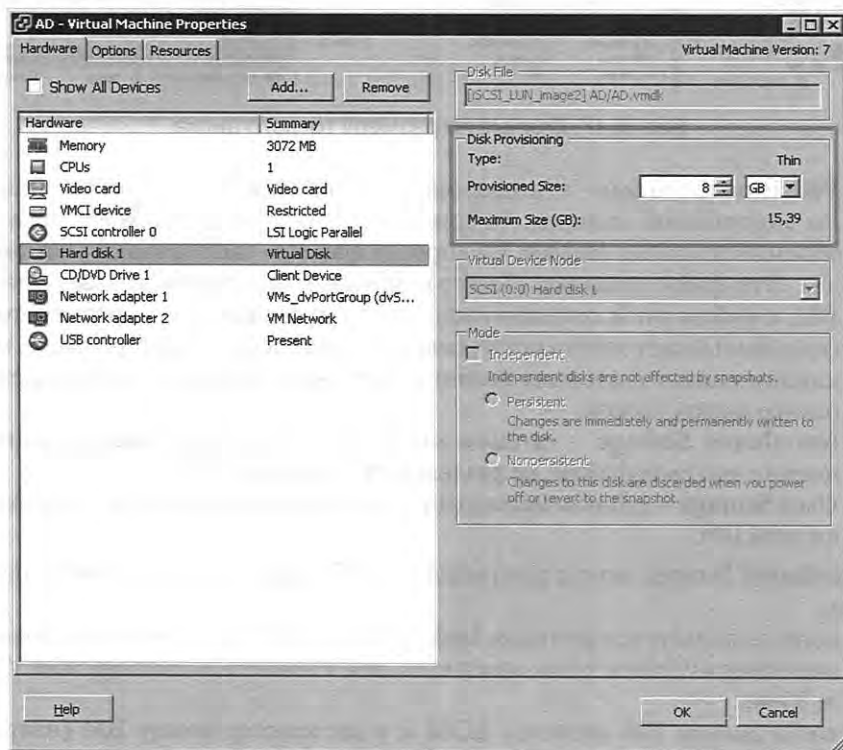


Рис. 5.18. Меню увеличения размера диска

Уменьшение номинального размера thin- или thick-диска

Если вы выделили для виртуальной машины диск некоего размера, а затем поняли, что выдали слишком много, то есть несколько способов отобрать лишнее место. Перечислим их все:

- ❑ в первую очередь следует упомянуть об использовании VMware Converter. С его помощью ВМ конвертируется в ВМ на том же ESXi, но можно указать диски меньшего размера;
- ❑ затем можно воспользоваться средствами, работающими «изнутри», – ПО работы с образами дисков типа Ghost или Acronis. Можно подключить к ВМ второй диск, нужного размера и переписать на него образ диска, размер которого хотим уменьшить. Часто эту операцию удобно выполнять, загрузив ВМ с LiveCD;
- ❑ попробовать найти какую-то стороннюю утилиту, выполняющую эту работу для ВМ на ESXi. К сожалению, подсказать что-то не могу, но, может быть, к моменту прочтения вами этих строк что-то и появится;
- ❑ вручную уменьшить сначала раздел в гостевой ОС, а затем файл vmdk.

Поговорим про эти способы чуть подробнее.

VMware Converter. Напомню, что VMware Converter Standalone бесплатно загружается с сайта VMware.

Установите конвертор, запустите мастер конвертации. В нем укажите, что вам необходимо конвертировать VMware Infrastructure Virtual Machine, укажите имя и учетную запись для доступа к vCenter. Затем, на шаге Source Data, у вас будет возможность выбора размера диска вновь создаваемой VM (рис. 5.19).

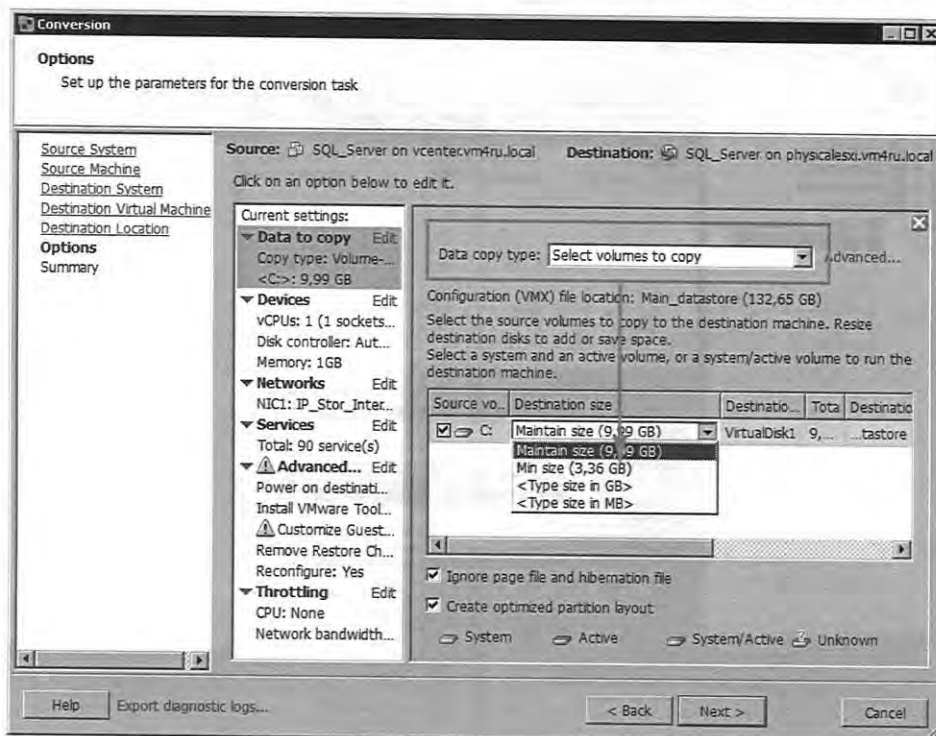


Рис. 5.19. Мастер конвертации ВМ, шаг выбора размера диска

Конвертор сам создаст новую ВМ с дисками нужного размера, сам скопирует данные и сам уменьшит размер раздела файловой системы гостевой ОС.

Если вам не хочется выключать ВМ для конвертации, тогда в начале мастера конвертации укажите, что вы хотите конвертировать Physical Machine. В этом случае обращение на конвертируемую ВМ произойдет по сети, на нее установится агент конвертера, который обеспечит конвертацию без выключения этой ВМ.

Перенос образа диска на диск меньшего размера. В принципе, в краткой аннотации я уже все рассказал про этот способ.

Уменьшение размера диска вручную. Это неподдерживаемый способ, который вы применяете на свой страх и риск. Тем не менее иногда его применение оправдано, удобно и успешно. Убедитесь в отсутствии снимков состояния (snapshot) для ВМ перед его применением.

Первый шаг, который вам необходимо выполнить, – это уменьшение размера раздела на уменьшаемом диске. Выбор средства для этого зависит от типа гостевой ОС. Например, в Windows Server 2008 для этого не требуется дополнительных утилит (рис. 5.20).

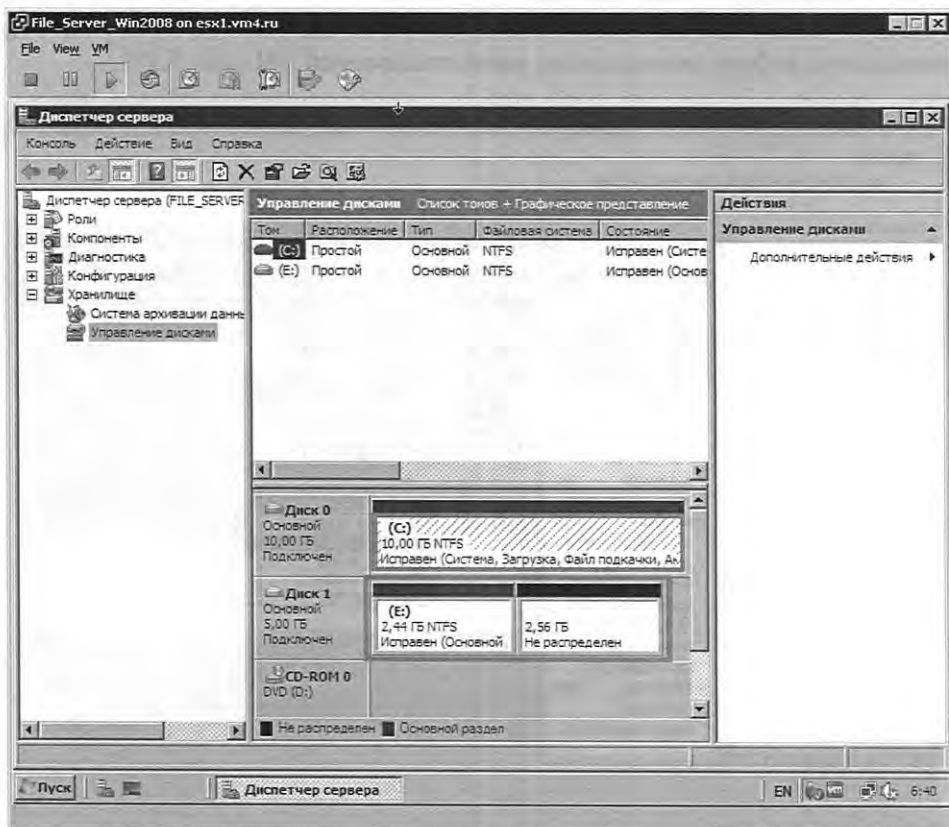


Рис. 5.20. На уменьшаемом диске должно появиться неразмеченное место

Следующий шаг – выключение VM и открытие в текстовом редакторе файла vmdk. Обратите внимание на то, что диск VM состоит из двух vmdk, с именами вида:

1. disk2.vmdk – это тестовый файл описания геометрии и структуры диска.
2. disk2-flat.vmdk – это непосредственно данные.

Вот *.vmdk для уменьшаемого диска нам и нужен. В командной строке ESXi можно использовать текстовый редактор vi. Можно воспользоваться утилитами FastSCP или WinSCP.

В открытом файле vmdk мы увидим что-то вроде:

```
# Extent description
RW 10485760 VMFS "foo-flat.vmdk"
```

Умножением RW на 512 получаем размер диска:

$$10485760 \times 512 = 5\,368\,709\,120 \text{ (5 Гб)}.$$

Например, хотим уменьшить диск до 3 Гб. Для этого делаем расчеты и меняем disk.vmdk:

$$3 \text{ Гб} = 3 \times 1024 \times 1024 \times 1024 \text{ байта} = 3\,221\,225\,472 \text{ байта}.$$

Поделим на 512, получим количество блоков = 6 291 456. Заменяем число блоков на новое:

```
# Extent description
RW 6291456 VMFS "foo-flat.vmdk"
```

Последний шаг – делаем горячую или холодную миграцию этой VM (или этого одного диска) на другое хранилище, и после этой операции диск становится нужного размера (рис. 5.21).

Если у нас нет vCenter, то есть данные операции недоступны, можно клонировать этот диск из командной строки:

```
# vmkfstools -i disk.vmdk disk_new_small.vmdk
```

Обратите внимание. Если последняя операция реализуется через Storage VMotion, то уменьшение диска произойдет без выключения виртуальной машины. Мигрировать можно не всю VM, а лишь уменьшаемый диск.

Уменьшение реального размера thin-диска

Что делать, если у вас есть vmdk, который хочется уменьшить? Это может быть vmdk типа thick, который так и так занимает много места – и иногда нам хочется перевести его в состояние thin. Или это может быть «распухший» thin-vmdk, внутри которого содержится много удаленных данных.

К сожалению, на текущий момент мне неизвестен такой способ для vSphere 5. Для справки оставляю способ, работавший в vSphere 4.

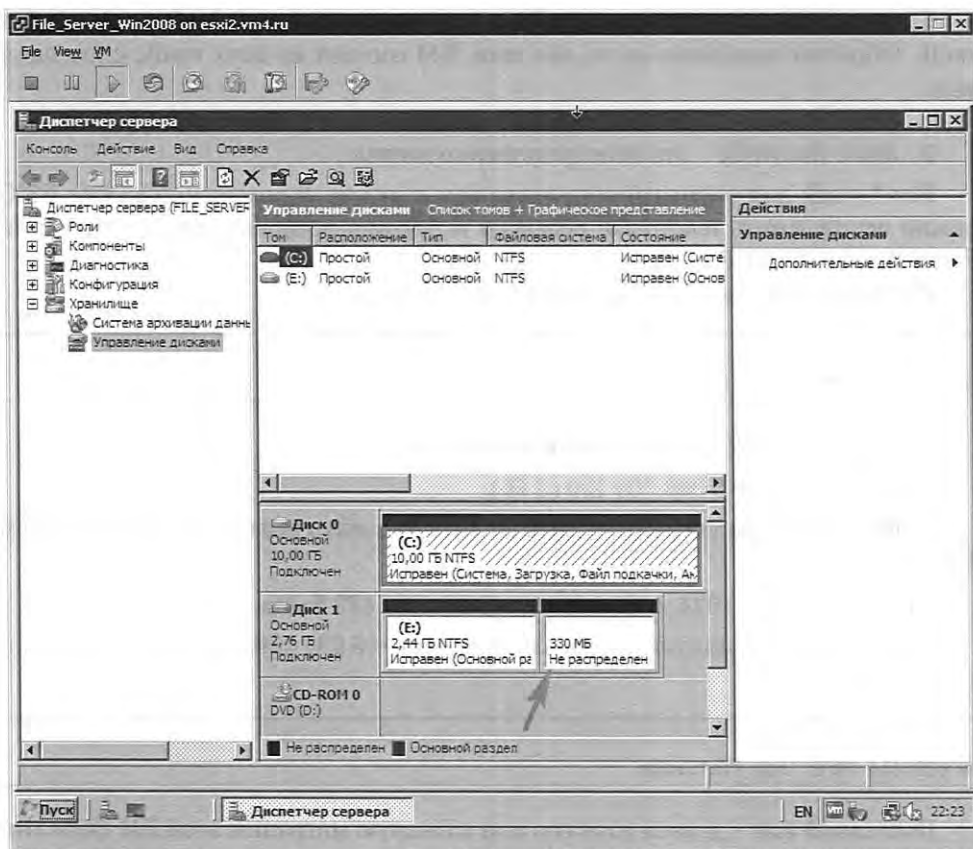


Рис. 5.21. Диск уменьшился

1. Необходимо обнулить блоки, занимаемые удаленными данными. Для Windows в этом может помочь утилита sdelete от Sysinternals. Запускаем ее внутри VM, направляя на диск с удаленными данными:

```
sdelete -c E:
```

Это для диска E:\.

Важно – работа этой утилиты вызовет увеличение реального размера тонкого диска до максимального значения – не запускайте этот процесс, если на хранилище недостаточно свободного места для роста этого vmdk.

2. После окончания ее работы запускаем процесс Storage VMotion и в мастере выбираем настройку **Change to Thin Provisioned Disk**.

Важно! Заработает (даже в vSphere 4) не всегда.

Должно быть выполнено одно (любое) из условий:

- миграция должна происходить между хранилищами с разных систем хранения (как вариант – между СХД и локальными дисками);
- миграция происходит между хранилищами одной СХД, но VMFS создан с разным размером блока (размер блока неактуален для VMFS 5, но был актуален для VMFS 3);
- миграция происходит как угодно, но перед ее началом мы выполнили команду

```
~ # vsish  
/> set /config/VMFS3/intOpts/EnableDataMovement 0
```

- Затем, после миграции, данный параметр стоит вернуть в исходное значение:

```
/> set /config/VMFS3/intOpts/EnableDataMovement 1
```

Все вышеописанные условия приводят к тому, что ESXi 4 использовал старый, неоптимальный механизм переноса данных, который зато умел переносить только реально занятые блоки vmdk-файлов.

По окончании миграции мы получим тонкий диск размером в объем реально занимающих место данных гостя.

Если имеющаяся у нас лицензия не позволяет использовать Storage VMotion (**Migration** ⇒ **Change Datastore** для включенной ВМ), вместо нее можно сделать холодную миграцию на другое хранилище (тот же пункт меню, когда ВМ выключена) или клон (**Clone**) этой ВМ. Полученная копия будет занимать меньше места на хранилище за счет очистки удаленных данных. Исходную же ВМ мы просто удалим.

Обратите внимание. В свойствах виртуальной машины, выделив HDD, вы увидите его тип (thick или thin) в строке **Type**.

Удаление диска

Когда вы заходите в свойства виртуальной машины, выделяете диск и нажимаете кнопку **Remove**, система спрашивает вас, как именно этот диск надо удалить (рис. 5.22).

Если этот диск вам еще нужен, например вы хотите подключить его к другой ВМ, то вам нужен пункт **Remove from virtual machine**.

Однако если вы выберете **Remove from virtual machine** в случае, когда хотите именно удалить данный файл-диск, то файл останется на хранилище и продолжит занимать место. Будьте внимательны и при необходимости именно удалить файл-диск выбирайте **Remove from virtual machine and delete files from disk**.

К сожалению, встроенного простого способа обнаружить неправильно удаленные, «осиротевшие» файлы-диски, впустую занимающие место, не существует. Рекомендую стороннюю утилиту RVTools (<http://www.robware.net>).

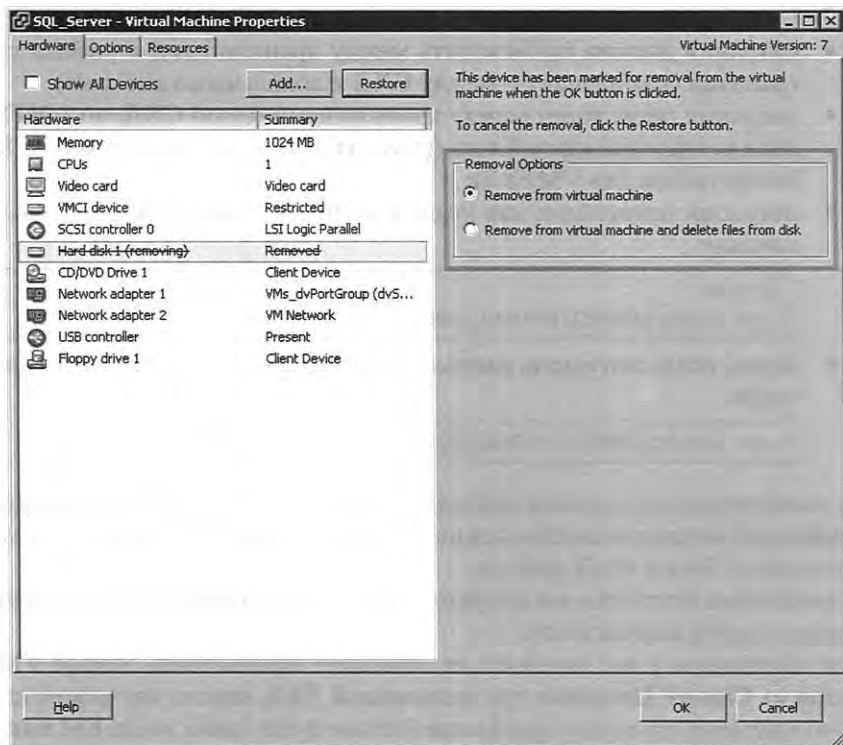


Рис. 5. 22. Варианты удаления диска ВМ

5.4.3. Выравнивание (*alligment*)

Существует такое понятие, как «выровненный» или «невыровненный раздел». Суть его в том, что при операциях чтения-записи массив оперирует некими блоками (или страйпами). И файловая система оперирует некими (другими) блоками. В некоторых случаях блоки файловой системы не выровнены по границам блоков (страйпов) системы хранения, так как граница создания первого блока файловой системы сдвинута. Это происходит потому, что x86 операционные системы создают в начале раздела master boot record (MBR), занимающую 63 блока.

Это означает, что при чтении или записи некоторых блоков с точки зрения файловой системы будет произведено чтение или запись двух блоков на системе хранения, что отрицательно сказывается на производительности дисковой подсистемы, потому что она оказывается ниже реально достижимой в случае оптимальной настройки.

В случае виртуализации ситуация даже немного сложнее: у нас есть блоки на СХД, блоки файловой системы VMFS и блоки файловой системы гостевой ОС в файле vmdk.

На рис. 5.23 показан невыровненный, плохой случай наверху и выровненный – внизу.

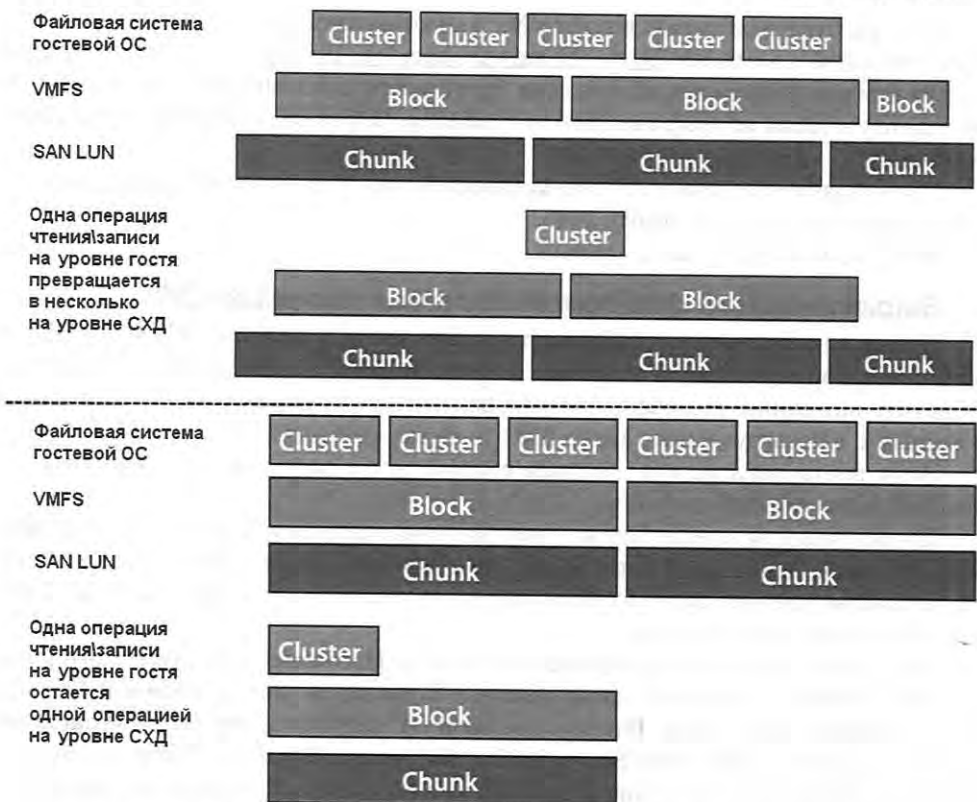


Рис. 5. 23. Иллюстрация выровненных и невыровненных разделов
Источник: VMware

В англоязычной документации вам могут попасться термины *Chunk*, *Block* и *Cluster* – соответственно, для SAN, VMFS и файловой системы гостевой ОС (NTFS в гостевой Windows).

По доступным мне данным, падение производительности в невыровненном случае не является значительным в большинстве случаев, порядка 10%. Однако имеет смысл производить его для шаблонов – тогда с минимальными усилиями диски большинства наших ВМ будут выровнены.

Также для виртуальных машин с интенсивной нагрузкой на дисковую подсистему, особенно при случайном доступе, выравнивание лучше производить.

Выравнивание VMFS

Этап первый – выравнивание VMFS. Для большинства СХД достаточно при создании LUN указать правильный тип (обычно «VMware»). Это выравнивает блоки создаваемого LUN в соответствии с параметрами VMFS. VMFS, создаваемый из графического интерфейса, создается выровненным по границе 2048.

Если вы будете создавать раздел VMFS из командной строки, то ознакомьтесь с инструкцией по ссылке <http://kb.vmware.com/kb/1036609>. Это может пригодиться вам при решении проблем, если вдруг окажетесь в ситуации, когда VMFS не удастся создать из графического интерфейса. Впрочем, вероятность этого невысока.

При создании раздела VMFS из командной строки вы должны будете указать начальный сектор – указывайте 2048.

Все, раздел VMFS у нас выровнен.

Выравнивание файловой системы гостевой ОС

Теперь надо выровнять файловую систему гостя в файле vmdk. Я расскажу об этом на примере данной операции для создаваемого шаблона ВМ с Windows. Обратите внимание, что выравнивание рекомендуется делать только для диска с данными, для загрузочного диска ВМ это не так критично.

Итак, постановка задачи – создать выровненный диск для ВМ с Windows, из которой затем сделаем шаблон.

Первое, что нам понадобится, – виртуальная машина с Windows. У этой ВМ уже есть диск, на который установлена ОС. Добавим к ней второй диск. Мы его сначала выровняем из-под этой Windows, затем отключим и подключим уже к новой ВМ – будущему шаблону.

Итак, заходим в свойства ВМ, **Add** на вкладке **Hardware**, добавляем **Hard Disk** нужного размера. После операции **rescan** в Менеджере дисков гостевой ОС мы обнаруживаем новый диск. Теперь в гостевой ОС запускаем утилиту **diskpart.exe** (она актуальна для Windows Server 2003 и 2008, но для Windows 2008 создаваемые по умолчанию разделы файловой системы уже выровнены). Выполните команды:

1. Для просмотра списка дисков:

```
List disk
```

Добавленный будет «Disk 1».

2. Для последующих операций над этим диском:

```
Select disk 1
```

3. Для создания раздела с правильным выравниванием:

```
Create partition primary align=64
```

4. Для назначения буквы созданному разделу:

```
select partition 1  
remove noerr  
assign letter=E noerr
```

5. Для выхода из diskpart:

```
Exit
```

6. Для форматирования созданного раздела в NTFS, с размером блока в 32 Кб:

```
Format E: /FS:NTFS /A:32K
```

Если вам хочется посмотреть, выровнены ли существующие разделы, сделать это можно так:

1. В гостевой ОС: **Start** ⇒ **Run** ⇒ **msinfo32**.
2. В открывшейся утилите пройдите **Components** ⇒ **Storage** ⇒ **Disks**. Для русской версии Windows это **Компоненты** ⇒ **Запоминающие устройства** ⇒ **Диски**.

Вас интересует поле **Partition Starting Offset** (Начальное смещение раздела).

Для выровненных разделов число из этого поля должно нацело делиться на размер блока данных (например, в случае кластера по умолчанию для NTFS – на 4096).

За дополнительной информацией обратитесь в статью базы знаний Майкрософт 929 491 (<http://support.microsoft.com/kb/929491>).

Обратите внимание, что в случае подключения к VM LUN как RDM в свойствах LUN (LUN Protocol Type) необходимо ставить тип гостевой ОС для корректного выравнивания без дополнительных усилий.

5.4.4. Raw Device Mapping, RDM

Raw Device Mapping (RDM) представляет собой механизм для прямого доступа виртуальной машины к конкретному LUN устройств хранения SAN (Fibre Channel, iSCSI, FCoE) или DAS (для DAS это не поддерживается официально, не всегда работает).

VM будет хранить свои данные непосредственно на этом LUN, а не в файле vmdk на разделе VMFS, созданном на LUN.

Для того чтобы подключить к VM какой-то LUN, сначала создайте его со стороны SAN. Этот LUN должен быть презентован всем ESXi, на которых эта VM может оказаться. На этом LUN не должно быть раздела VMFS. К VM подключается именно и только LUN целиком.

Зайдите в свойства VM, нажмите кнопку **Add** на вкладке **Hardware** и выберите **Hard Disk**. После нажатия **Next** вы увидите следующие шаги мастера:

1. **Select a Disk** – здесь вы выберете, хотите ли создать новый файл vmdk, подключить уже существующий и расположенный на доступном этому ESXi хранилище или же подключить RDM. Сейчас рассмотрим последний вариант. Если его нельзя выбрать – значит, нет подходящих LUN. Напомню, что RDM – это альтернатива VMFS, и если LUN уже отформатирован в VMFS, то из списка кандидатов на RDM-подключение он пропадает.
2. **Select Target LUN** – здесь мы увидим список LUN, которые можем подключить как RDM.
3. **Select Datastore** – выберем, где будет расположен файл vmdk, являющийся ссылкой на подключаемый RDM. Кстати, размер этого файла будет отображаться равным размеру LUN, хотя на самом деле он займет всего несколько мегабайт. Этот файл нужен для управления доступом к RDM, см. рис. 5.24. Положение данного файла влияет на то, где будут создаваться файлы разницы (delta) этого LUN при снимках состояния (snapshot).

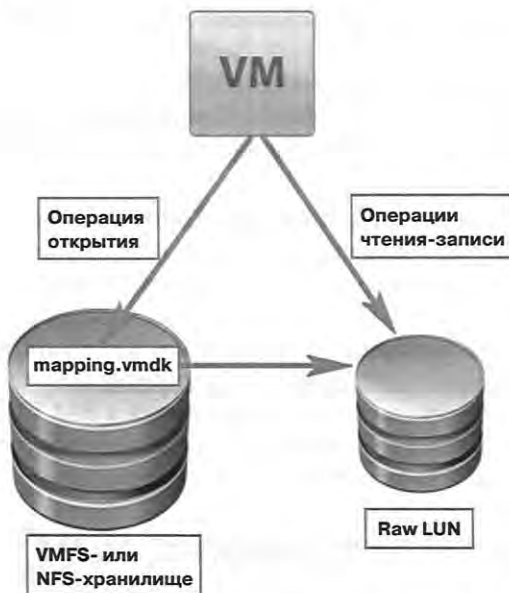


Рис. 5.24. Схема подключения RDM

4. **Compatibility Mode** – режим совместимости. Два варианта:
 - **Physical** – в этом режиме гипервизор не перехватывает и не изменяет SCSI-команды от VM на LUN (с одним исключением: команда REPORT). Также от VM не скрываются характеристики устройства. Режим нужен для подключения LUN размером более 2 Тб, для общих дисков кластера Майкрософт в варианте «виртуальный-физический» и для задач, требующих именно прямого доступа к диску. В примеры последним обычно приводят средства управления SAN;
 - **Virtual** – в этом режиме гипервизор имеет право перехватывать и изменять SCSI-команды, что позволяет применять к этому LUN некоторые механизмы ESXi, такие как снимки состояния (snapshot) и операция клонирования.
5. **Advanced Options** – эти настройки обычно менять не требуется:
 - **Virtual Device Node** – на каком ID какого виртуального контроллера будет располагаться этот виртуальный диск. SCSI (1:2) означает, что этот диск займет второе SCSI ID на виртуальном SCSI-контроллере номер 1 (нумеруются они с нуля). **Обратите внимание:** если этого контроллера в VM еще нет – он будет добавлен вместе с диском;
 - **Mode** – настройка доступна только для virtual RDM. Если поставить флажок Independent, то к этому виртуальному диску не будут применяться снимки состояния (snapshot). В режиме **Persistent** все изменения будут немедленно записываться в этот файл vmdk. В режиме **Non-persistent** все изменения с момента включения будут записываться

в отдельный файл, который будет удаляться после выключения VM. Такой режим имеет смысл, например, для демонстрационных VM. Мы их подготовили, настроили, перевели их диски в этот режим. Теперь после выключения они всегда будут возвращаться к своему состоянию на момент включения данного режима.

RDM пригодится вам в случаях:

- ☐ для подключения к VM диска (здесь – физического LUN) размером более 2 Тб. Напомню, что в качестве BIOS для этой VM должен использоваться EFI;
- ☐ организации кластера Майкрософт типа «виртуальный-виртуальный» и «виртуальный-физический»;
- ☐ из политических соображений – когда идея помещать данные VM в файл vmdk не находит понимания;
- ☐ при миграции в VM физического сервера, хранящего данные на СХД, данные можно не копировать. Можно LUN с этими данными подключить к VM как RDM. Впоследствии эти данные можно перенести в файл vmdk без остановки VM с помощью Storage VMotion;
- ☐ в случае RDM на LUN хранятся непосредственно данные VM. К ним можно применять функции системы хранения (например, снимки состояния (snapshot) для организации резервного копирования);
- ☐ для задействования NPIV. Дать каждой VM собственный WWN возможно, лишь если она использует RDM.

Сделать RDM подключение к LUN на системе хранения не представляет труда. Однако не всякий локальный RAID-контроллер позволит создать RDM из клиента vSphere. Можно попробовать два способа решения этой проблемы, если такая задача вдруг встала.

В первую очередь попробуйте настройку **Configuration ⇒ Advanced Settings ⇒ RDMFilter ⇒ флажок RDMFilter.HBAisShared**. Если после установки флажка локальный диск как RDM все равно не предлагает подключить – приходится выполнять эту операцию чуть хитрее, из командной строки.

Для подключения локального диска как RDM из командной строки делаем следующее:

1. Создаем новый диск для VM. Размер и параметры оставляем по умолчанию.
2. Подключаемся к серверу с помощью putty.
3. Выполняем


```
fdisk -l
```
4. Обнаруживаем (по размеру) диск, который хотим подключить как RDM;
5. Сопоставляем его с именем вида paa.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx с помощью команды


```
esxcfg-scsidevs -c
```
6. После этого вводим команду

```
vmkfstools -i [Путь к vmdk -файлу ] -d rdm:/vmfs/devices/disks/naa.
xxxxxxxxxxxxxxxxxxxxxx [vmdk-файл]
```

Например:

```
[root@esx1.vm4.ru]# vmkfstools -i /vmfs/volumes/SCSI_LUN_1/SQL_Server/ SQL_
Server.vmdk -d rdm:/vmfs/devices/disks/naa.60043560bd135e00123823443a44ag56
Local_RDM.vmdk
```

Мы получили vmdk, ссылающийся на LUN, то есть RDM-диск. Этот vmdk подключаем к виртуальной машине как обычный vmdk.

Обратите внимание. Содержимое LUN, подключенного как RDM, может быть перенесено в файл vmdk при таких операциях, как Storage vMotion, холодная миграция VM, клонирование VM. В таком случае этот LUN перестанет использоваться, диском VM станет файл vmdk, куда были скопированы данные. Если вам не нужен такой перенос – будьте внимательны при этих операциях. См. <http://kb.vmware.com/1005241>.

5.5. Настройки VM

Если зайти в свойства VM, то вы увидите несколько вкладок: **Hardware**, **Options**, **Resources**, **Profiles** и **vServices**. Поговорим здесь про настройки, доступные под вкладкой **Options** (рис. 5.25).

General Options

Здесь вы можете поменять имя VM. Обратите внимание: при изменении этого имени файлы VM и ее каталог не переименовываются. Это очень неудобно, поэтому переименований имеет смысл избегать.

Еще здесь вы можете посмотреть, какой каталог является рабочим для VM. Именно в нем располагаются все конфигурационные файлы VM, в нем по умолчанию предлагают создавать файлы-диски, в нем по умолчанию создается файл подкачки при включении VM.

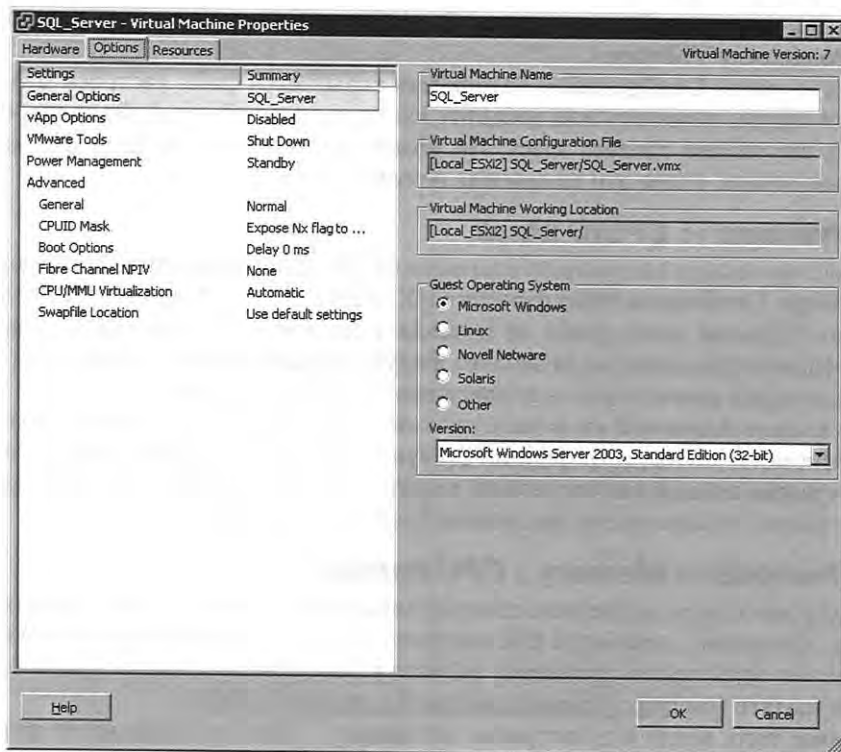
Наконец, здесь мы можем поменять тип гостевой ОС. Тип влияет на выбор дистрибутива VMware tools, который автоматически подмонтируется к VM при выборе пункта **Install/Upgrade VMware tools** в контекстном меню VM. Также ESXi не будет предлагать нам не поддерживаемые гостевой ОС компоненты для VM (например, PVSCSI или vmxnet3).

vApp Options

Подробности см. в разделе про vApp.

VMware tools

Здесь мы можем указать, какие операции выполняются при нажатии на иконки управления питанием для VM. В ESXi 5 по умолчанию делаются «Shutdown guest» и «Restart Guest», то есть корректное завершение работы и перезагрузка VM. Если вы работаете с VM, в которой нет VMware tools (например, еще нет, только боремся с установкой ОС), то нажатия иконок вызывают ошибку «отсут-

Рис. 5.25. Вкладка **Options** для ВМ

ствуют VMware tools для корректного завершения работы гостя». Так вот, здесь мы можем поменять действия по умолчанию на «Power Off» и «Reset». Иногда бывает полезно.

VMware tools могут запускать сценарий при включении, при восстановлении из состояния «suspend», перед вводом в состояние «suspend», перед выключением гостевой ОС. Здесь мы флажками можем настроить, при каких событиях сценарии запускать надо. Обычно тут мы ничего не меняем.

Еще мы можем указать проверку версии VMware tools на актуальность при каждом включении ВМ. Если флажок стоит, то будет не только проверяться версия VMware tools, но они еще будут и автоматически обновляться.

Наконец, мы можем указать синхронизацию времени гостевой ОС со временем ESXi через VMware tools. Если у ВМ нет возможности синхронизировать время по сети через NTP, эта возможность часто выручает. Но может создать проблемы в случае, если ВМ сама является источником времени (например, контроллер домена).

Advanced ⇒ General

Здесь вы можете отключить ускорение (acceleration) для этой ВМ. Имеется в виду ускорение Intel VT. Выключать его нужно редко – это может помочь в ситуациях зависания ВМ при запуске какого-то приложения.

Флажки в пункте **Debugging and Statistics** пригодятся вам в случае обращения в поддержку VMware.

Под кнопкой **Configuration Parameters** скрывается файл настроек ВМ. Если вам необходимо поменять или добавить какие-то настройки в файл настроек ВМ (*.vmx), это можно сделать здесь. На момент внесения изменений ВМ должна быть выключена, иначе эти изменения не сохранятся.

Advanced ⇒ CPUID Mask

Здесь мы можем настраивать сокрытие от ВМ функций физических процессоров сервера. Особняком стоит функция NX/XD (Intel No eXecute и AMD eXecute Disable). Скрытие этого флага не позволяет гостевой ОС использовать данную процессорную функцию, но позволяет живую миграцию между серверами, на одном из которых процессоры не поддерживают этой функции.

По кнопке **Advanced** мы можем указывать, какие регистры процессора скрывать. Вспомогательная информация доступна по кнопке **Legend**. Несколько слов про ситуации, когда и как это можно делать, – в посвященном VMotion разделе.

Де-факто – скорее всего, вам эти настройки не пригодятся.

Advanced ⇒ Memory / CPU hotplug

Здесь мы можем разрешить горячее добавление процессоров и памяти. Разрешить это можно, лишь если ВМ выключена. Обратите внимание, что Windows 2003 может испытывать проблемы, работая в ВМ со включенным горячим добавлением памяти (см. <http://support.microsoft.com/kb/913568>).

Кроме того, когда эта настройка включена – ESXi не использует функцию «виртуальная NUMA», см. раздел 6.2.1.

Advanced ⇒ Boot Options

Здесь мы можем указать количество секунд паузы после процедуры POST и поставить флажок «Зайти в BIOS при следующем включении ВМ». Эти настройки бывают полезны при работе с ВМ через WAN, когда задержки не позволяют нам вовремя нажать **F2**, **F12** или **Esc** для вызова меню загрузки или захода в BIOS.

Также задержка загрузки иногда применяется для того, чтобы какие-то ВМ загружались позже других.

Начиная с версии 4.1 в этом пункте настроек появился флажок **Failed Boot Recovery**, который позволяет автоматически перезагрузить виртуальную машину, если она не нашла загрузочное устройство в течение указанного количества секунд. Эта настройка бывает полезна в первую очередь для виртуальных машин, загружающихся по PXE.

В пятой версии vSphere появилась поддержка нового типа загрузчика – EFI вместо BIOS. Для тех ОС, для которых мы можем выбирать, что использовать, EFI имеет смысл выбрать, когда хотим, чтобы загрузочным диском ВМ был диск более 2 Тб размером.

Advanced ⇒ Fibre Channel NPIV

В разделе, посвященном системам хранения, я рассказал, что такое NPIV и в каких ситуациях вам может быть интересно его задействовать.

Настройка выдачи уникального WWN для конкретной VM делается в этом пункте настроек. Обратите внимание, что FC HBA и коммутаторы FC должны поддерживать NPIV для успешного использования этой функции и на них эта функция должна быть включена.

Напомню, что использование NPIV возможно лишь для VM с дисками RDM. WWN должен быть уникальным. На стороне системы хранения подключаемый к VM LUN должен быть презентован для ее WWN.

CPU/MMU Virtualization

В современных серверах используются две технологии аппаратной поддержки виртуализации:

- ☐ поддержка виртуализации процессора. Названия – Intel VT / AMD V;
- ☐ поддержка виртуализации памяти. Названия – AMD NPT (Nested Page Tables) / Intel EPT (Extended Page Tables). Еще в разнообразных технических и маркетинговых материалах могут встречаться названия HAP (Hardware Assisted Paging), NPT (Nested Page Tables), EPT (Extended Page Tables) и RVI (Rapid Virtualization Indexing). Все это – об одном и том же.

Так вот. В данном пункте настроек мы можем указать, какие из этих функций должны использоваться для данной VM. Скорее всего, вы будете оставлять вариант по умолчанию – «Automatic».

Swapfile Location

Здесь мы можем указать, где хранить файл подкачки, который гипервизор создает для этой VM при ее включении. Варианты настройки:

- ☐ **Default** – хранить там, где указано на уровне сервера или DRS-кластера. Для сервера это указывается в пункте **Configuration** ⇒ **Virtual Machine Swapfile Location**. Выбрать можем какое-то одно хранилище, на котором будут появляться файлы подкачки всех VM, работающих на этом сервере, и у которых настроено хранить файл подкачки там, где скажет сервер. Этим местом может быть каталог VM – так это настроено по умолчанию;
- ☐ **Always store with the virtual machine** – значит, файл подкачки всегда расположен в рабочем каталоге VM, там, где ее vmx. Имя и путь к каталогу можно посмотреть в свойствах VM ⇒ вкладка **Options** ⇒ **General**;
- ☐ **Store in the host's swapfile datastore** – всегда хранить там, где указано в настройках сервера.

5.6. Файлы VM, перемещение файлов между хранилищами

В каталоге VM мы можем увидеть разные файлы – см. рис. 5.26.

На этом рисунке вы видите файлы VM с именем SQL_Server:

- ☐ **SQL_Server.vmx** – главный файл настроек VM;
- ☐ **SQL_Server.vmx** – вспомогательный файл настроек VM;
- ☐ **SQL_Server.vmdk** и **SQL_Server-flat.vmdk** – такая пара образует диск VM;

vmware.log	52 998
vmware-10.log	56 635
vmware-5.log	83 543
vmware-6.log	61 638
vmware-7.log	60 165
vmware-8.log	26 869
vmware-9.log	26 849
SQL_Server.nvram	8 684
SQL_Server.vmdk	522
SQL_Server-000001.vmdk	323
SQL_Server-000001-delta.vmdk	16 789 504
SQL_Server-flat.vmdk	5 368 709 120
SQL_Server.vmsd	396
SQL_Server-Snapshot1.vmsn	29 214
SQL_Server.vmtx	3 731
SQL_Server.vmx	265
SQL_Server-3eeb0bec.vswp	1 073 741 824

Рис. 5.26. Список файлов VM

- ❑ **SQL_Server-xxxxxxx.vswp** – файл подкачки ВМ. Это внешний относительно ВМ файл подкачки, задействуется он гипервизором. Данная функция называется VMkernel Swap (см. главу 6);
- ❑ **SQL_Server.nvram** – файл содержит настройки BIOS ВМ;
- ❑ **SQL_Server.vmsd** – файл с информацией о снимках состояния (snapshot) этой ВМ. Про сами снимки будет сказано чуть далее;
- ❑ **SQL_Server-000001.vmdk** и **SQL_Server-000001-delta.vmdk** – файлы-диски снимков состояния;
- ❑ **SQL_Server-Snapshot1.vmsn** – память, сохраненная при снимке состояния ВМ;
- ❑ несколько файлов журналов (*.log).

Несколько слов о каждом типе файлов поподробнее.

Файл VMX

В текстовом файле с расширением vmx описана вся конфигурация ВМ. В первую очередь это информация о виртуальном оборудовании: сетевые контроллеры, их MAC-адреса, к каким группам портов они подключены, SCSI-контроллеры и их тип, путь к дискам (файлам-vmdk), к файлу подкачки, к файлу BIOS, тип гостевой ОС и отображаемое имя (Display Name) ВМ, а также некоторые другие параметры, изменение которых невозможно из интерфейса.

Для нормальной работы ВМ этот файл должен существовать. Если вы хотите зарегистрировать на ESXi какую-то ВМ, то сделать это можно через **Browse Datastore**, в контекстном меню файла настроек (*.vmx). Также если вы выполняете какие-то манипуляции с ВМ из командной строки (включение, снимки состояния и прочее), то указанием, с какой ВМ делать эту операцию, является путь к ее файлу настроек.

Вам может потребоваться вносить какие-то правки в файл настроек. Для этого в клиенте vSphere зайдите в свойства выключенной ВМ, вкладка **Options** ⇒ **General** ⇒ кнопка **Configuration Parameters**. Или откройте его в текстовом редакторе. Подойдут или vi локальной командной строки, или утилиты WinSCP/FastSCP.

Пример параметров файла настроек (*.vmx), которые могут пригодиться:

```
isolation.device.connectable.disable = "true"  
isolation.device.edit.disable = "true"
```

Укажите эти две настройки, если вы хотите запретить пользователям без административных привилегий отключать сетевые карты виртуальной машины через механизм `usb safely remove` и на вкладке **Devices** в настройках VMware Tools.

Обычно источником тех или иных настроек служат рекомендации специалистов поддержки VMware и статьи в Базе знаний. Полного списка параметров в открытом доступе не существует.

Файл NVRAM

В файле `.lvmgpt` содержатся настройки BIOS виртуальной машины. Эти настройки можно тиражировать простым копированием файла с нужными настройками между ВМ. Если этот файл удалить, он будет создан при следующем включении ВМ, с настройками по умолчанию.

Файл подкачки VSWP

Этот файл создается при включении ВМ и удаляется после выключения. Его размер равен количеству выделенной ВМ памяти минус значение настройки `memory reservation`. По умолчанию резерв для ОЗУ равен нулю. Обратите внимание, что если на хранилище не будет достаточно места для создания файла подкачки, то ВМ не включится.

По умолчанию файл подкачки создается в каталоге с ВМ. Однако мы можем указывать для сервера ESXi произвольное хранилище, на котором будут создаваться файлы подкачки ВМ, работающие на этом сервере. Указывать, хранить файлы подкачки на этом выделенном хранилище или в каталоге ВМ, мы можем для всех ВМ кластера, для всех ВМ сервера, для отдельной ВМ.

Для кластера мы можем указать, хранить ли по умолчанию файл подкачки ВМ в ее каталоге или на каком-то LUN, который указан как хранилище файлов подкачки для каждого сервера. В свойствах кластера за это отвечает настройка **Swapfile Location** (рис. 5.27).

Для сервера **Configuration** ⇒ **Virtual Machine Swapfile Location** ⇒ **Edit** (рис. 5.28).

Наконец, мы можем указать, где хранить файл подкачки в свойствах конкретной ВМ ⇒ вкладка **Options** ⇒ **Swapfile Location** (рис. 5.29).

Файлы VMDK

Чаще всего дисками ВМ выступают файлы `vmdk`, расположенные на разделах VMFS или NFS. Притом, когда вы добавляете к ВМ один диск (рис. 5.30), создаются сразу два файла (рис. 5.31).

Это файлы `<имя ВМ>.vmdk` и `<имя ВМ>-flat.vmdk`. Первый – текстовый, содержащий в себе описание геометрии диска и путь к `-flat`-файлу. А во втором хранятся непосредственно данные.

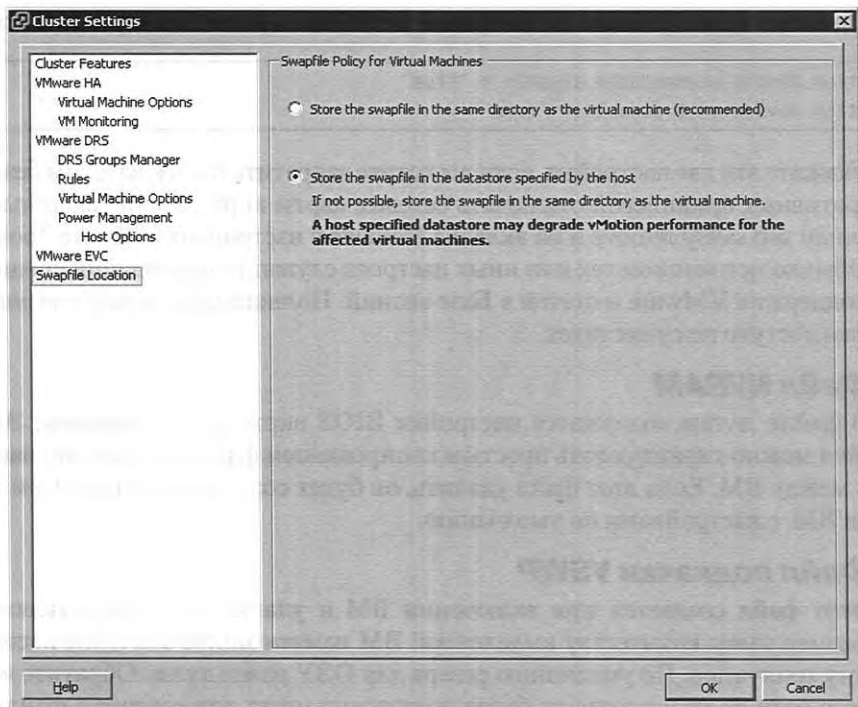


Рис. 5.27. Настройки расположения файлов подкачки ВМ для кластера

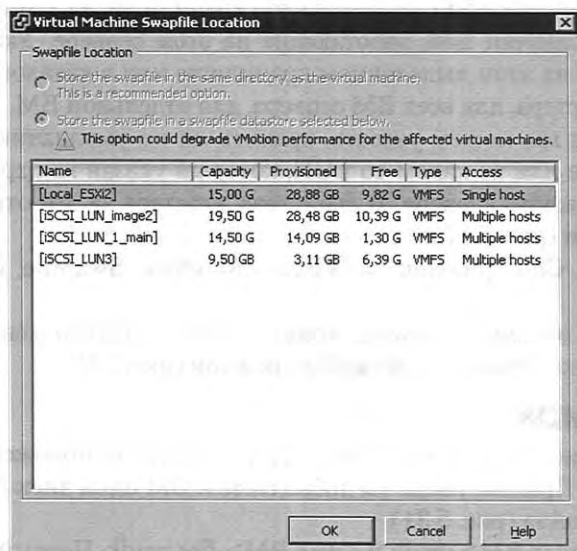


Рис. 5.28. Указание хранилища для файлов подкачки на сервере ESXi

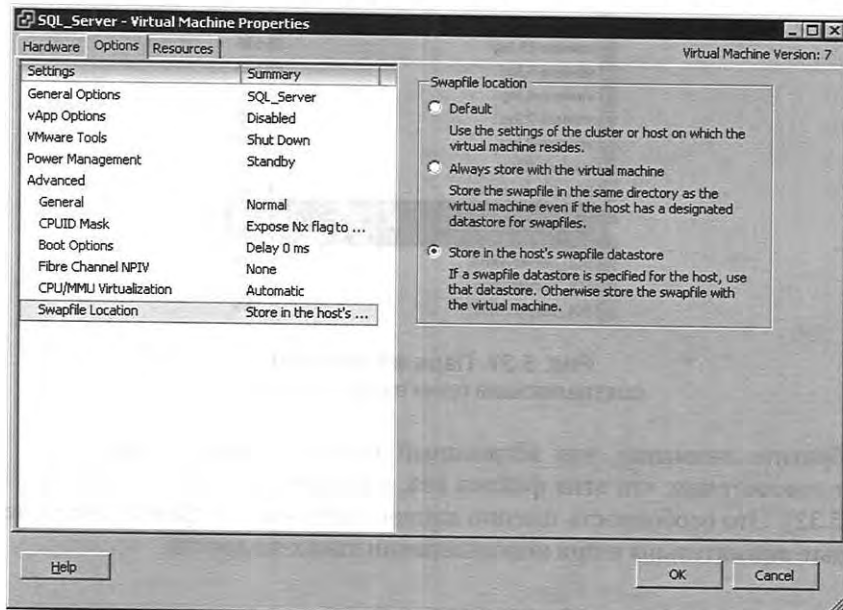


Рис. 5.29. Настройка местоположения файла подкачки для ВМ

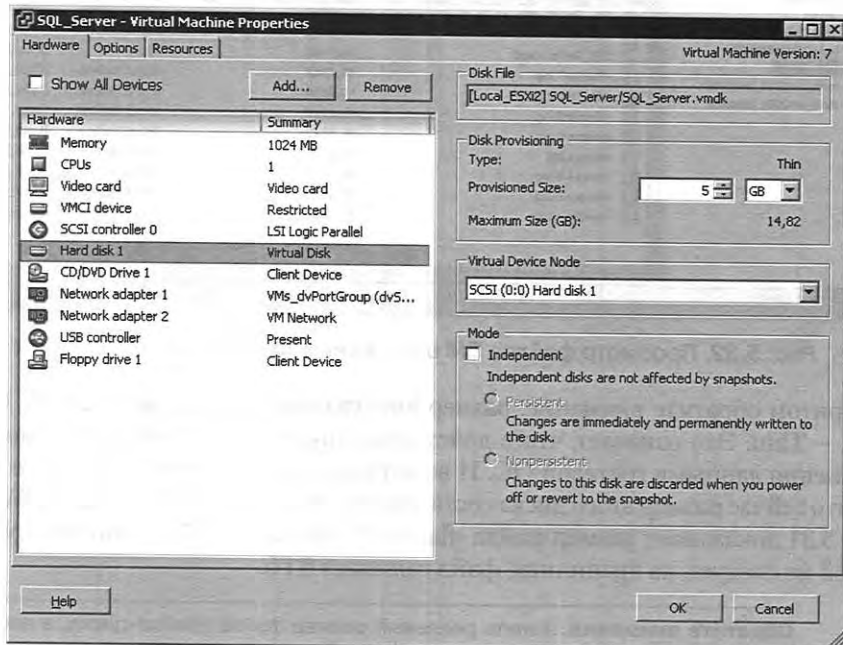


Рис. 5.30. Виртуальный HDD в свойствах ВМ

vmware.log	57 365
vmware-10.log	56 635
vmware-5.log	83 543
vmware-6.log	61 638
vmware-7.log	60 165
vmware-8.log	26 869
vmware-9.log	26 849
SQL_Server.nvram	8 684
SQL_Server.vmdk	545
SQL_Server-flat.vmdk	5 368 709 120
SQL_Server.vmsd	43
SQL_Server.vmx	3 728
SQL_Server.vmxs	265

Рис. 5.31. Пара файлов vmdk, составляющая один виртуальный диск

Обратите внимание, что встроенный в клиент vSphere файловый менеджер не покажет вам, что этих файлов два, – вы увидите только <имя VM>.vmdk (рис. 5.32). Это особенность именно данного встроенного файлового менеджера, но будьте внимательны и при использовании каких-то других.

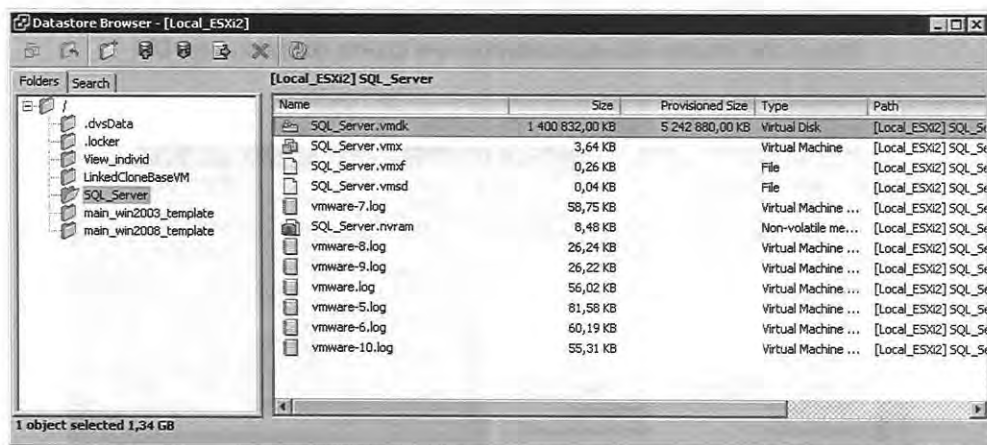


Рис. 5.32. Просмотр файлов VM встроенным файловым менеджером

Притом обратите внимание: размер виртуального диска равен 10 Гб. Но тип диска – Thin. Это означает, что в действительности файл vmdk растет по факту заполнения данными гостевой ОС. И во встроенном файловом менеджере мы видим, что сейчас размер этого диска составляет чуть более 6 Гб. Но утилита WinSCP с рис. 5.31 показывает размер файла -flat.vmdk равным 10 Гб. В данном случае верить ей не следует, на хранилище файл занимает 6 Гб.

Обратите внимание. Узнать реальный размер файла (файла-диска, в частности) из командной строки можно командой `du`. Параметр `-h` является указанием на то, что объем занимаемого места следует отображать в удобном для восприятия виде.

Параметр `-a` позволит отобразить информацию обо всех файлах указанного или текущего каталога. Например, следующая команда покажет размер всех файлов в указанном каталоге: `du -h -a /vmfs/volumes/iscsi_LUN_1/SQL_Server/`

Что мы увидим изнутри гостевой ОС, показано на рис. 5.33.

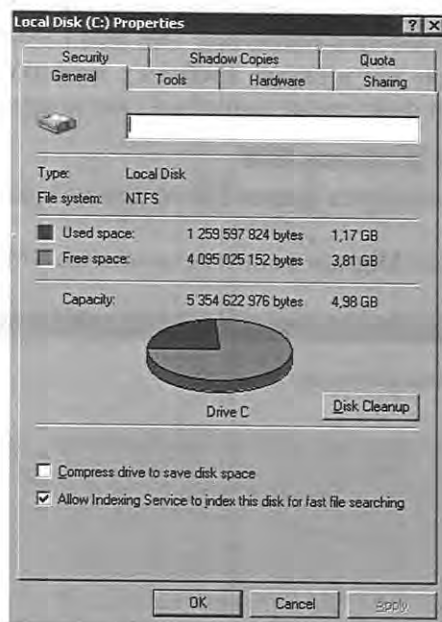


Рис. 5.33. Заполненность диска изнутри гостевой ОС

Кроме этой пары файлов, в каталоге с ВМ могут быть расположены еще несколько файлов `vmdk` для каждого ее диска. Они появляются при создании снимка состояния ВМ, и про них я расскажу чуть позже.

Полезно знать: в файле `vmx` указаны пути к файлам `vmdk` (не `-flat.vmdk`). В `vmdk` – путь к `-flat.vmdk`. У снимков состояния тоже есть структура, о которой – в разделе про снимки (snapshot). Если вы захотите переименовать файлы ВМ, то переименовывать их надо последовательно, прописывая новые пути и имена в соответствующие файлы. Впрочем, конкретно для решения этой проблемы в разы проще мигрировать ВМ на другое хранилище или сделать клон. Эти операции автоматически приведут в соответствие название ВМ и имена ее файлов.

Обратите внимание. В каталоге виртуальной машины могут появиться дополнительные файлы, кроме описанных. Не все из них одинаково полезны.

Например, файл вида `VM-b3ab8ade.vms` – в этом файле сохраняется содержимое оперативной памяти остановленной (suspended) виртуальной машины. При старте такой suspended-машины файл (в теории) должен удаляться. Но иногда он остается. Не удаляется он и при перезагрузке ВМ. Удалится он только при

полной остановке. Однако если виртуальная машина была перемещена при помощи *SVmotion*, то файл остается в старом каталоге, так как нигде в настройках машины (*VM.vmx*) он уже не фигурирует и никогда не будет удален автоматически. Соответственно, возможна ситуация появления файлов, впустую потребляющих место на хранилище.

Также в старом каталоге виртуальной машины после *SVmotion* можно обнаружить файл вида *vmware-vmx-zdump.000*. Это файл *coredump* от виртуальной машины, и интересен он вам в случае проблем с этой *VM*, для передачи дампа в поддержку. Иначе этот файл также потребляет место впустую.

Перемещение файлов *VM*

Для того чтобы переместить файлы *VM* на другое хранилище, в *vCenter* есть операция **Migrate**.

Итак, выберите пункт **Migrate** в контекстном меню *VM* (рис. 5.34).

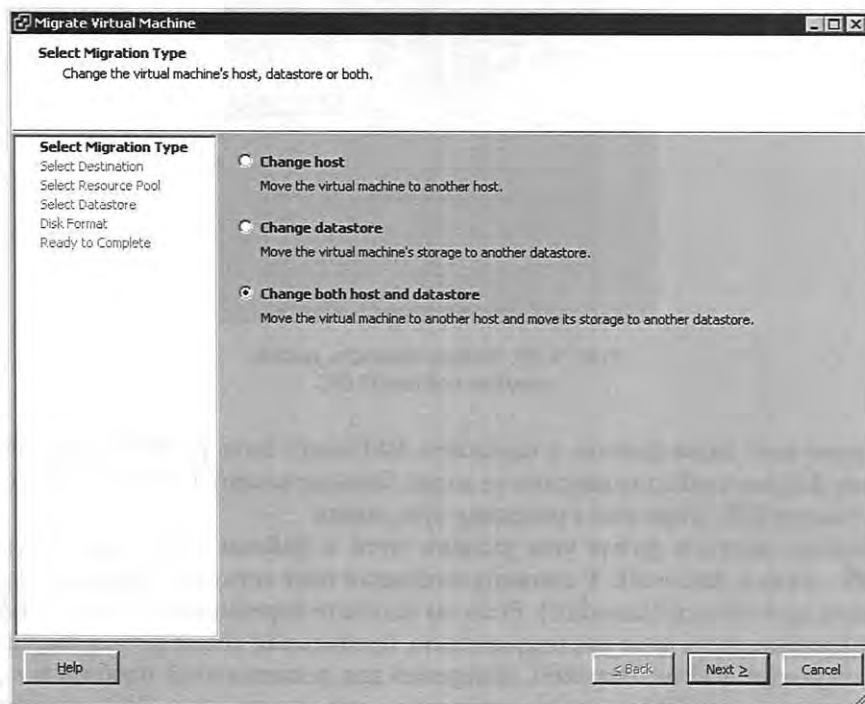


Рис. 5.34. Начало мастера миграции *VM*

Change host предполагает регистрацию *VM* на другом сервере, но без изменения местоположения файлов виртуальной машины. В данном контексте этот пункт нас не интересует.

Change datastore предполагает миграцию файлов *VM* (всех или только выбранных дисков) на другое хранилище, но видимое серверу, где виртуальная ма-

шина числится сейчас. Эта операция возможна без ограничений для выключенной виртуальной машины. Для включенной ВМ осуществление данной операции требует наличия лицензии на Storage VMotion.

Change both host and datastore предполагает смену и хоста, и хранилища. Таким образом, возможен перенос ВМ на другой сервер и на хранилище, видимое только другому серверу. Такая операция возможна лишь для выключенной виртуальной машины.

При выборе второго или третьего пункта в коротком мастере вас попросят указать новое хранилище, притом вы можете перенести только один или несколько файлов vmdk этой ВМ, не обязательно всю ее.

Альтернативный способ запуска этой процедуры более нагляден. Перейдите в **Home** ⇒ **Inventory** ⇒ **Datastores** ⇒ вкладка **Virtual Machines** для хранилища, откуда хотите перенести ВМ. Затем просто перетащите эту ВМ на то хранилище, куда хотите ее переместить.

Если у вас нет vCenter, а желание перенести ВМ есть, то это также несложно:

1. В контекстном меню ВМ выберите **Remove From Inventory**.
2. Затем любым файловым менеджером, подойдет и встроенный, перенесите файлы ВМ на нужное хранилище.
3. Из встроенного файлового менеджера вызовите контекстное меню для файла vmtx перенесенной ВМ и выберите **Register Virtual Machine**. ВМ появится в клиенте vSphere.

5.7. Снимки состояния (Snapshot)

ESXi позволяет нам создавать для виртуальных машин снимки состояния. Снимок состояния – это точка возврата. На рис. 5.35 показан диспетчер снимков (Snapshot Manager) виртуальной машины с двумя снимками.

Обратите внимание на первый снимок с именем «Snapshot1_before_VMware_tools». Его иконка с зеленым треугольником указывает на то, что он был сделан при работающей ВМ, с сохранением ее памяти. То есть при возврате на этот снимок мы вернемся в состояние работающей ВМ.

Создание снимка состояния весьма тривиально. В контекстном меню ВМ выбираем **Snapshot** ⇒ **Take Snapshot**.

В открывшемся окне (рис. 5.36) нас попросят ввести имя снимка и описание. В случае если ВМ включена в момент снятия снимка, то доступны два флажка:

- ☐ если верхний флажок стоит, то в отдельном файле будет сохранено содержимое памяти ВМ на момент снятия снимка. В таком случае при восстановлении на сохраненное состояние виртуальная машина окажется работающей;
- ☐ нижний флажок указывает VMware Tools, чтобы они попробовали обеспечить целостность данных ВМ. «Quiesce» означает, что будет произведена попытка остановить работу всех служб с диском, чтобы в момент снятия снимка на диске не было недописанных файлов.

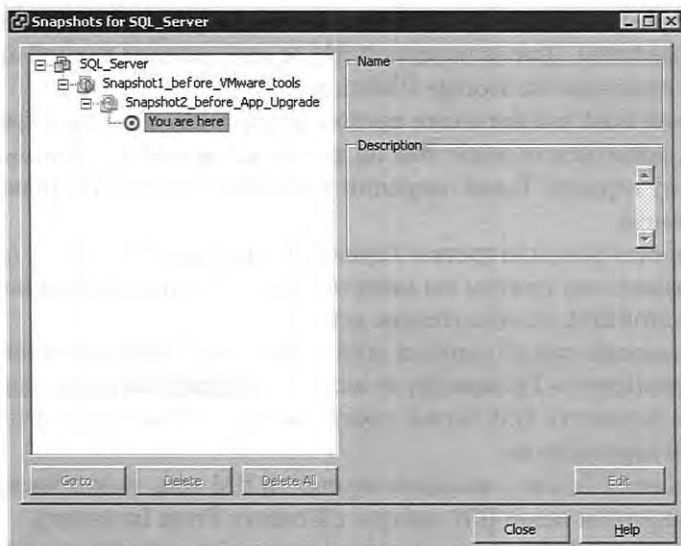


Рис. 5.35. Диспетчер снимков состояния

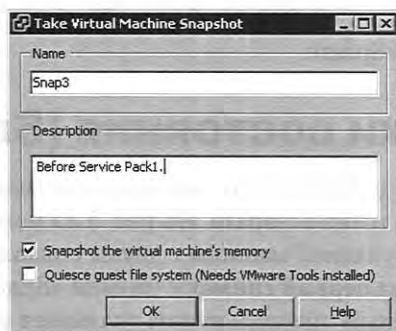


Рис. 5.36. Мастера создания снимка состояния

Снятие снимка создает точку возврата не только для состояния и данных гостевой ОС, но и для конфигурации ВМ. Несмотря на то что ВМ со снимками можно мигрировать VMotion и Storage VMotion, мастер миграции проверяет на корректность только текущую конфигурацию ВМ. Например, если мы перенесли ВМ на другой сервер, а затем откатали ее на какой-то из ранее созданных снимков, а на момент этого снимка к ней был подключен диск с приватного хранилища – вы столкнетесь с неработоспособностью ВМ на текущем сервере.

У одной ВМ может быть множество снимков, притом даже несколько веток (рис. 5.37).

Когда вы откатываетесь на какой-то из них, вы теряете состояние «You are here», то есть все наработки с последнего снимка состояния. Если вы хотите их

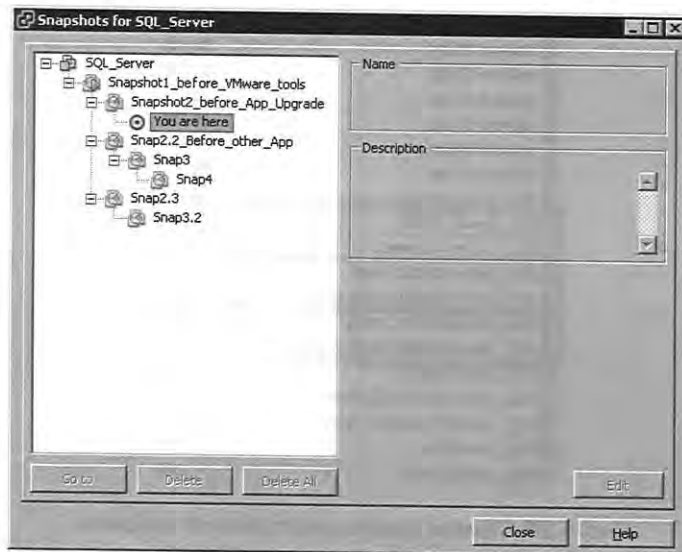


Рис. 5.37. Диспетчер снимков состояния для ВМ с большим количеством снимков

сохранить – делайте еще один снимок, а потом уже откатывайтесь к желаемому. У этой ситуации прямая аналогия с сохранением в компьютерной игре – если вы загружаете какое-то сохранение, то текущее состояние игры теряется. А вернуться на него вы можете, лишь если сохраните уже текущее состояние перед загрузкой другого сохранения.

Кнопка **Delete** удаляет выделенный снимок состояния. Притом если текущее состояние ВМ – после этого снимка, то файл-дельта этого снимка не удаляется, а сливается с файлом родительского снимка. На примере рис. 5.36:

- ❑ если вы удалите последний в ветке снимок (Snap4 или Snap3.2), то соответствующий файл-дельта с диска удалится;
- ❑ если вы удалите любой из остальных снимков, то соответствующий ему файл-дельта сольется с файлом-дельта дочернего снимка. Так происходит потому, что при удалении (например) Snap3 у нас остается Snap4 – а он содержит все изменения относительно Snap3, который содержит все изменения относительно Snap2.2, и т. д. Получается, если данные Snap3 удалить – Snap4 «повиснет в воздухе». Поэтому файл-дельта Snap3 не удаляется, а сливается с файлом-дельта Snap4. А если удалить родительский снимок верхнего уровня (Snapshot1_before_VMware_tools), то его файл-дельта сольется с основным файлом – диском ВМ.

Давайте посмотрим на снимки состояния с точки зрения файловой системы. Если у нас есть ВМ с двумя снимками, как на рис. 5.35, то с помощью утилиты WinSCP мы увидим следующую картину – рис. 5.38.

Среди прочих вы видите два файла, составляющие диск ВМ, – это файлы SQL_Server.vmdk и SQL_Server-flat.vmdk. Однако, кроме них, вы видите еще не-

vmware.log	59 988
vmware-10.log	56 635
vmware-11.log	57 365
vmware-12.log	83 754
vmware-13.log	63 621
vmware-14.log	56 684
vmware-15.log	59 585
SQL_Server.nvram	8 684
SQL_Server.vmdk	545
SQL_Server-flat.vmdk	5 368 709 120
SQL_Server-000001.vmdk	323
SQL_Server-000001-delta.vmdk	16 789 504
SQL_Server-000002.vmdk	275
SQL_Server-000002-delta.vmdk	12 288
SQL_Server.vmsd	750
SQL_Server-Snapshot2.vmsn	1 075 011 305
SQL_Server-Snapshot3.vmsn	29 221
SQL_Server.vmx	3 735
SQL_Server.vmxs	265

Рис. 5.38. Список файлов виртуальной машины, у которой есть снимки состояния

сколько файлов *.vmdk, а также не рассмотренные ранее файлы vmsn и vmsd. Поговорим про них.

Файлы VMSD

Это текстовые файлы, в которых описаны существующие снимки состояния и их структура. Пример содержимого этого файла:

```
.encoding = "UTF-8"
snapshot.lastUID = "2"
;Это порядковый номер последнего
;снимка.

snapshot.numSnapshots = "2"
;Текущее количество снимков.
snapshot.current = "2"
;Какой снимок является последним,
;то есть после какого идет состояние
;"You are here".

snapshot0.uid = "1"
;Это начало описания первого снимка.
snapshot0.filename = "SQL_Server-Snapshot1.vmsn" ;Какой vmsn-файл содержит в себе
;содержимое памяти на момент
;создания снимка.

snapshot0.displayName = "Snapshot1_before_VMware_tools"
snapshot0.description = ""
snapshot0.type = "1"
snapshot0.createTimeHigh = "290757"
snapshot0.createTimeLow = "-728460977"
snapshot0.numDisks = "1"
snapshot0.disk0.fileName = "SQL_Server.vmdk" ;Какой файл vmdk содержит все
;изменения на диске начиная с момента
;создания этого снимка.

snapshot0.disk0.node = "scsi0:0"
```

```

snapshot1.uid = "2"
;Здесь начинается описание второго и
;последнего (в данном случае) снимка
;этой ВМ.

snapshot1.filename = "SQL_Server-Snapshot2.vmsn"
snapshot1.parent = "1"
snapshot1.displayName = "Snapshot2_Before_App_Upgrade"
snapshot1.description = ""
snapshot1.createTimeHigh = "291622"
snapshot1.createTimeLow = "-717423872"
snapshot1.numDisks = "1"
snapshot1.disk0.fileName = "SQL_Server-000001.vmdk"
snapshot1.disk0.node = "scsi0:0"

```

Файлы *vmsn*

В этих файлах находятся содержимое оперативной памяти ВМ и конфигурация на момент снятия снимка состояния. Если в этот момент ВМ была выключена или флажка «сохранять память» при создании снимка не стояло, то этот файл текстовый, только с описанием конфигурации ВМ на момент снимка.

Файлы *-delta.vmdk*

На рис. 5.38 вы видите две пары файлов:

- ❑ SQL_Server-000001.vmdk и SQL_Server-000001-delta.vmdk;
- ❑ SQL_Server-000002.vmdk и SQL_Server-000002-delta.vmdk.

Первичный .vmdk = SQL_Server.vmdk.

.vmdk, созданный после 1-го снимка = SQL_Server-000001.vmdk.

.vmdk, созданный после 2-го снимка = SQL_Server-000002.vmdk.

Файлы 00000#.vmdk являются файлами метаданных. Пример содержания:

```

# Disk DescriptorFile
version=1
encoding="UTF-8"
CID=26a39a09
parentCID=411fd5ab
createType="vmfsSparse"
parentFileNameHint="SQL_Server.vmdk"
# Extent description
RW 6291456 VMFSSPARSE "SQL_Server-000001-delta.vmdk"

# The Disk Data Base
#DDB

ddb.longContentID = "8d7a5c7af10ea8cd742dd10a26a39a09"
ddb.deletable = "true"

```

Обратим внимание на три поля в .vmdk-файле:

- ❑ поле CID;
- ❑ ссылка на parentCID;
- ❑ поле parentNameHint.

Обратите внимание. Первичный .vmdk не содержит поля «parentNameHint», а его «parentCID» всегда равняется «ffffff».

Суть здесь в следующем: снимки состояния образуют цепочку, и все звенья этой цепочки зависят друг от друга. Если по каким-то причинам целостность цепочки нарушается, ВМ перестает включаться. В таком случае имеет смысл вручную проверить и при необходимости восстановить цепочку. То есть открываем файл 00000#.vmdk последнего снимка перед текущим состоянием. В нем должен быть указан CID предпоследнего снимка. В предпоследнем должен быть указан CID предпредпоследнего. Итерацию повторить.

Посмотрите на рис. 5.39.

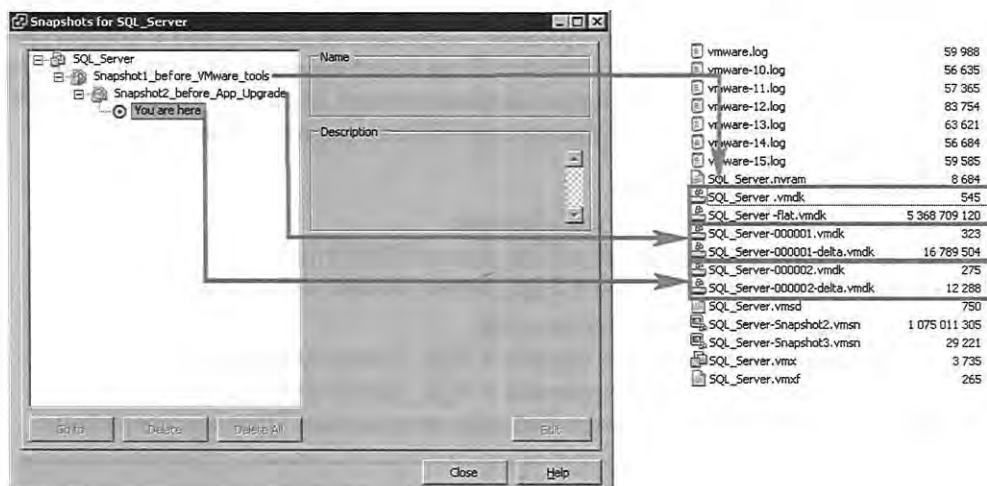


Рис. 5.39. Связь снимков с файлами vmdk

Когда ВМ работает без снимков, ее дисками является исходная пара файлов vmdk. Когда вы делаете первый снимок состояния, эта исходная пара переводится в режим только чтения. Теперь изменять эти файлы гипервизор не может – ведь мы зафиксировали состояние ВМ. Поэтому гипервизор создает пару 00001.vmdk и 00001-delta.vmdk – теперь все изменения на дисках с момента первого снимка пишутся в них. Но затем мы фиксируем и следующее состояние снимком номер 2. Теперь изменения начинают писаться в созданные гипервизором файлы 00002.vmdk и 00002-delta.vmdk. Но они не самодостаточны, поэтому файлы 00001-delta.vmdk и -flat.vmdk продолжают использоваться на чтение.

Обратите внимание: в файлы -delta.vmdk пишутся все изменения данных на диске ВМ. Как максимум, на диске может поменяться все. Поэтому каждый файл -delta.vmdk по размеру может быть равен номинальному объему диска ВМ. Изначально файлы delta.vmdk создаются размером в один блок VMFS и по одному блоку увеличиваются при необходимости. Напомню, что размер блока VMFS мы указываем при создании раздела, и допустимые значения – от 1 (по умолчанию) до 8 МБ.

Также если размер диска ВМ близок к максимально возможному на VMFS размеру в 2 Тб, мы не сможем создать снимок этой ВМ. Это связано с тем, что `-delta.vmkd`-файл требует до 2 Гб дополнительного места из-за накладных расходов. Получается, что если «размер диска ВМ» плюс 2 Гб больше, чем 2 Тб, то снимок не создается.

Обратите внимание. При создании снимка состояния ВМ у нас фиксируется состояние гостевой файловой системы и, необязательно, памяти ВМ. Однако, кроме этого, сохраняется и конфигурация ВМ как объекта ESXi. Возможна ситуация, когда вы сделали снимок, после этого как-то изменили конфигурацию инфраструктуры и связанные с этим параметры конфигурации ВМ. Например, поменяли имена групп портов и переключили сетевые контроллеры ВМ на новые группы портов. Или пересоздали хранилища, и CD-ROM виртуальной машины теперь ссылается на файл iso по другому, чем раньше, пути. Тогда при возврате на снимок состояния виртуальная машина окажется в некорректной конфигурации. Возможно, ее нельзя будет включить до устранения конфликта. Более того, на ее вкладке **Summary** (и в некоторых других местах интерфейса) мы увидим упоминание тех групп портов и хранилищ, которые она задействует хотя бы в одном снапшоте – и не важно, задействует ли их в данный момент.

Плюсы и минусы снимков состояния

Плюсами снимков состояния является их суть – создание точки возврата. Это, без сомнения, здорово для тестовых виртуальных машин. Это бывает полезно для производственных ВМ. Например, удобно сделать снимок состояния перед установкой какого-нибудь большого обновления – в случае возникновения каких-то проблем после его установки нам легко возвратиться на предыдущее состояние. Практически все средства резервного копирования виртуальных машин создают снимок состояния на время самой операции резервного копирования. Однако все остальное является минусами, и давайте эти минусы перечислим:

- ❑ если вы используете диски ВМ в `thick`-формате, то они сразу занимают все выделенное им место. Если теперь сделать снимок состояния ВМ, то этот снимок занимает место сверх уже занятого самим диском. Как максимум ВМ с одним снимком состояния может занять в два раза больше места, чем номинальный объем ее диска. Это, конечно, граничный случай, однако треть, а то и половину сверх объема диска снимок за несколько месяцев занять способен вполне. Это сильно зависит от интенсивности изменения данных ВМ;
- ❑ уменьшается надежность – для ВМ со снимками состояния выше вероятность каких-то проблем именно из-за наличия снимков. Обычно эти проблемы выглядят как не запускающаяся после миграции или лунного затмения виртуальная машина;
- ❑ возможны казусы с конфигурацией виртуальной машины. Если ВМ подключена к группе портов «vlan15», затем для нее сделали снимок состояния, потом подключили второй виртуальный hdd, затем сделали еще один снимок состояния, то:
 - если вы удалите группу портов «vlan15», то это название все равно будет фигурировать на вкладке **Summary** для этой ВМ и в списке всех групп

портов (**Home** $\Rightarrow$ **Inventory** $\Rightarrow$ **Networking**). Если состояние ВМ потом откатить на первый снимок состояния – ВМ окажется отрезанной от сети, так как группы портов, к которой ВМ считает себя подключенной, уже не существует;

- если откатить состояние ВМ на первый снимок состояния, то второй жесткий диск пропадет из конфигурации ВМ, хотя сам файл vmdk останется на хранилище;
- огромной осторожности требует использование снимков состояния на ВМ с распределенными приложениями (самый типичный пример – Active Directory). Если не все, а только часть ВМ, участвующих в этом приложении (один или несколько контроллеров домена), вернулись в прошлое – мы рискуем получить серьезные проблемы вплоть до полной неработоспособности всего распределенного приложения (в случае Active Directory см. «USN Rollback»). Практически 100%-ная вероятность таких проблем привела к тому, что Майкрософт не поддерживает использование любых механизмов снимков (включая снимки ВМ, снимки СХД и резервное копирование с помощью программ снятия образов) с контроллерами домена Active Directory;
- когда на одном хранилище очень много ВМ со снимками состояния, может уменьшиться производительность дисковой подсистемы из-за накладных расходов, возникающих вследствие того, что файлы vmdk снимков увеличиваются блоками;
- сам факт наличия снимка состояния влечет за собой penalties к производительности ВМ. Дело в том, что диском ВМ теперь является цепочка файлов vmdk, и для каждой одной операции чтения и записи от гостевой ОС гипервизору приходится совершать несколько операций ИО. При записи блока – отметить, что этот блок хранится в дельте, а не в исходном vmdk. Перед чтением – проверить, в исходном vmdk нужный блок или в какой-то дельте;
- удаление снимков состояния в некоторых случаях требует очень много свободного места на хранилище. Это связано с тем, что когда мы удаляем снапшот, гипервизор должен прочитать содержимое файла-дельты и записать это содержимое в предыдущий файл-дельту или оригинальный файл-диск. Получается, что до последнего мгновения процесса удаления снапшота дельта присутствует два раза – добавленная к предыдущей дельте и сама по себе. И лишь в последний миг эта дельта будет удалена (это неверно, когда удаляется самый старый снапшот – его дельта добавляется к исходному vmdk);
- удаление снимков состояния, кроме места на хранилище, потребляет производительность системы хранения. Ведь если у нас есть дельта размером гигабайт (или десять. Или сто. Или тысяча), нам файл этого размера следует прочитать и записать. А если СХД уже перегружена в данный момент? Прочим операциям (ВМ, работающей на этом же LUN/RAID группе) достанется меньше IOPS.

Вывод: для производственных ВМ снимки состояния используем строго по делу и удаляем сразу после того, как в них пропала нужда.

Обратите внимание. В этой книге под «снимками состояния (snapshot)» понимаются «снимки состояния (snapshot) VMware». Эта оговорка делается по той причине, что многое из перечисленного неверно для «аппаратных» снапшотов систем хранения. Работают такие снапшоты где-то сходным, а где-то иным образом.

5.8. VMware tools

VMware tools – это наборы драйверов и утилит под многие поддерживаемые ОС. Несмотря на то что операционные системы могут работать в виртуальных машинах и без VMware tools, настоятельно рекомендуется их устанавливать.

Именно в составе VMware tools содержатся драйверы для разнообразных виртуальных комплектующих. Для тех из них, что не имеют физических аналогов (pvscsi, vmxnet#), VMware tools являются единственным источником драйверов (хотя упомяну, что драйверы для виртуального оборудования VMware включены в ядро Linux актуальных версий). Кроме драйверов, VMware tools содержат скрипты и службы, обеспечивающие такие возможности, как автоматическое «освобождение» курсора мыши при покидании им окна консоли и операции копирования и вставки текста между консолью виртуальной машины и ОС клиента (это неполный список возможностей VMware tools).

Дальнейшее будет в основном ориентировано на гостевые ОС Windows. Для популярных Linux-дистрибутивов отличия в основном интерфейсные, вида «не нажать **ОК** три раза», а «запустить такой-то сценарий». Для устаревших или менее распространенных ОС (например, Solaris, Netware) какие-то функции могут быть недоступны. Так что если у вас не только Windows – загляните в документацию.

Обратите внимание. Для быстрой установки VMware tools под Linux с настройками по умолчанию вам поможет ключ `--default`. Полная команда выглядит примерно так:
`[root@linuxServer ~]# vmware-config-tools.pl -default.`

Для установки (и обновления) VMware tools выберите в контекстном меню ВМ пункт **Guest ⇒ Install/Upgrade VMware tools**. В появившемся меню можно выбрать, хотите ли вы интерактивную установку с помощью мастера или установку автоматическую, с указанными настройками (рис. 5.40).

Выбор не появляется, если VMware tools в ВМ не установлены, автоматический вариант доступен только для обновления VMware tools.

Какую бы вы ни выбрали, в первый из виртуальных CD-ROM этой ВМ автоматически подмонтируется образ iso с дистрибутивом VMware tools. Обратите внимание, что тип дистрибутива зависит от типа гостевой ОС, указанного в настройках ВМ.

Обратите внимание. Даже если у ВМ нет ни одного привода CD/DVD-ROM, но какая-то старая версия VMware Tools уже установлена, то обновление VMware Tools все равно будет произведено. Дистрибутив новой версии передается через внутренний механизм коммуникации между сервером и ВМ.



Рис. 5.40. Выбор типа установки VMware tools

Если выбрать интерактивную установку, то после подмонтирования в случае Windows сработает автозапуск и вы увидите мастер установки. В случае других ОС вам придется выполнить несколько команд – см. документацию.

Если выбрать автоматическую установку, то без указания дополнительных параметров VMware tools будут установлены с настройками по умолчанию. Дополнительные параметры актуальны только для гостевых Windows – это параметры для msi дистрибутива VMware tools.

Обратите внимание: после установки VMware tools вас попросят перезагрузить ВМ. При установке в автоматическом режиме перезагрузка также произойдет автоматически. Часто VMware tools обновить хочется, а вот перезагрузить ВМ сразу после установки – нет. В этом поможет следующий параметр, добавленный в поле **Advanced Options** в автоматическом режиме установки:

```
/S /v "/qn REBOOT=R"
```

В случае интерактивной установки вы сможете выбрать компоненты VMware tools для установки. Менять список по умолчанию обычно не приходится.

В некоторых ОС необходимо включить аппаратное ускорение видео. Для этого зайдите в персонализацию рабочего стола, выберите пункт **Параметры дисплея** ⇒ на вкладке **Диагностика** нажмите кнопку **Изменить параметры** ⇒ ползунок до упора вправо (рис. 5.41). В пятой версии vSphere эта настройка обычно делается автоматически, но если вам некомфортна работа в консоли (низкая отзывчивость курсора мыши), то этот параметр стоит проверить.

После установки VMware tools в Windows в трее и в панели управления появляются характерные иконки. Двойной клик запускает настройки VMware tools в гостевой ОС.

На вкладке **Options** мы можем:

- ❑ включить или выключить синхронизацию времени через VMware tools. Не надо включать такую синхронизацию, если гостевая ОС уже использует NTP для синхронизации времени. Если есть выбор что использовать, то мне более надежной представляется синхронизация времени не через VMware tools. Этот флажок можно поставить в свойствах ВМ ⇒ вкладка **Options** ⇒ **VMware tools**;

- ❑ обновить можно, и не заходя в консоль ВМ, точно так же, как вы устанавливаете VMware tools: в контекстном меню ВМ выбрав **Guest** ⇒ **Install/Upgrade VMware tools**. Скорее всего, удобнее будет выбрать автоматическую установку. Обратите внимание, что эту операцию можно запустить для нескольких ВМ сразу. В клиенте vSphere пройдите **Home** ⇒ **Inventory** ⇒ **Hosts and Clusters** ⇒ выделите **Datacenter** ⇒ вкладка **Virtual Machines** ⇒ рамкой или с помощью **Ctrl** и **Shift** выделите нужные ВМ, в контекстном меню для них выберите **Guest** ⇒ **Install/Upgrade VMware tools**;
- ❑ обновить VMware tools можно при помощи PowerCLI:

```
Get-VM <выборка из одной или нескольких ВМ> | Update-Tools -NoReboot
```

- ❑ наконец, для массового обновления VMware tools на многих ВМ удобнее всего использовать VMware Update Manager. См. посвященный ему раздел.

Не забывайте, что данная операция связана с перезагрузкой ВМ. Сначала установщик деинсталлирует старую версию, затем устанавливает новую.

На вкладке **Devices** вы можете отключить какие-то из контроллеров этой ВМ (см. рис. 5.42).

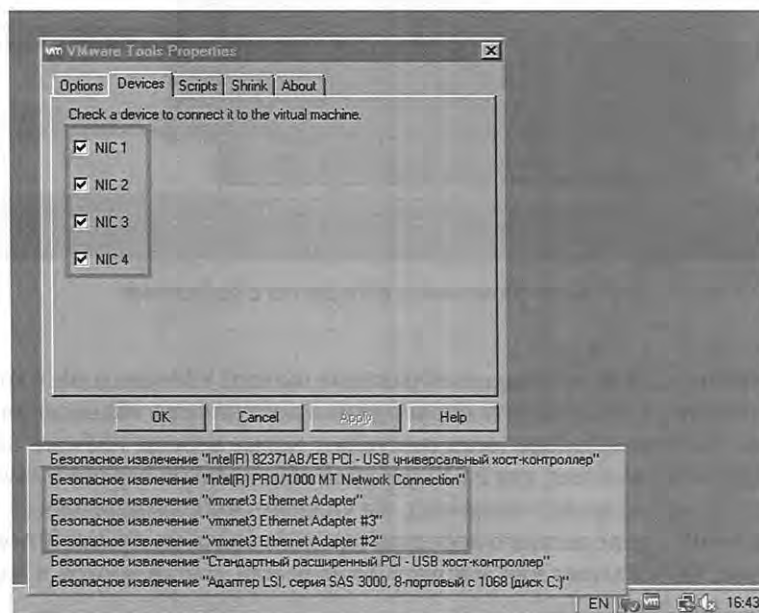


Рис. 5.42. Отключение контроллеров ВМ

Часто эта возможность вредна, так как позволяет пользователям ВМ вывести ее из строя отключением сетевых контроллеров. Чтобы предотвратить такую возможность, добавьте в файл настроек ВМ настройки:

```
devices.hotplug = "false"  
isolation.device.connectable.disable = "true"  
isolation.device.edit.disable = "true"
```

Первая из этих настроек запретит горячее удаление или добавление любых устройств в эту ВМ (пропадет возможность удалять контроллеры как USB-устройства), а две другие запретят отключение устройств через VMware tools.

На вкладке **Scripts** мы можем посмотреть или указать, какие сценарии запускаются VMware tools по тем или иным событиям по питанию ВМ.

Вкладки **Shared Folders** и **Shrink** не актуальны для ESXi, достались как присутствующие в VMware Workstation – а VMware tools и набор виртуального оборудования у этих продуктов весьма близки (хотя и не идентичны).

Обратите внимание. После установки VMware tools в каталоге C:\Program Files\VMware\VMware Tools находятся все драйверы для виртуального оборудования VMware под данную версию ОС. Они могут пригодиться, если вдруг драйвер на какое-либо устройство не установился автоматически.

Версии VMware tools для разных версий ESXi можно загрузить на сайте <http://packages.vmware.com/tools>.

5.9. vAPP

vApp – это контейнер для виртуальных машин, который позволяет производить некоторые манипуляции над группой помещенных в него ВМ как над единым объектом (рис. 5.43).



Рис. 5.43. Схема vApp
Источник: VMware

В контекстном меню сервера или DRS-кластера вы увидите пункт **New vApp**. В запустившемся мастере вас попросят указать имя и настройки количества ресурсов для создаваемого vApp. Настройки количества ресурсов абсолютно такие же, как и для пулов ресурсов, – то есть vApp является пулом ресурсов, кроме прочего. О пулах ресурсов и о настройках распределения ресурсов см. в соответствующем разделе.

Для существующего vApp мы можем указать следующие настройки на вкладке **Options** (к сожалению, для собственноручно созданных vApp эффективно можно использовать только некоторые из описываемых настроек, см. резюме в конце раздела):

- ☐ **Resources** – здесь указываем настройки распределения ресурсов. См. «Пулы ресурсов»;
- ☐ **Properties** – здесь указываем значения произвольных полей. Сами произвольные поля задаются в пункте **Advanced** ⇒ кнопка **Properties**;
- ☐ **IP Allocation Policy** – VM в vApp могут получать сетевые настройки одним из трех типов:
 - **Fixed** – когда настройки произведены в гостевой ОС;
 - **Transient** – когда для включаемой VM выдается IP из диапазона IP-адресов (где настраивается этот диапазон, чуть далее). После выключения VM этот IP освобождается;
 - **DHCP**.
Но будут ли доступны Transient и DHCP, настраивается в окне настроек: **Advanced** ⇒ кнопка **IP Allocation**;
- ☐ **Advanced** – здесь мы можем указать информацию о vApp:
 - по кнопке **Properties** можно указать произвольные переменные, значения которых затем могут быть присвоены в пункте **Properties**, описанном выше;
 - по кнопке **IP Allocation** мы можем разрешить использовать **Transient** и **DHCP** варианты настройки IP для VM в этом vApp.

На вкладке **Start Order** указываем порядок запуска и интервалы между запуском VM в этом vApp. Эти настройки актуальны, когда мы выполняем операцию **Power On** для vApp, не для отдельной VM в нем.

Диапазон IP-адресов создается совершенно в другом месте – выделите дата-центр (объект vCenter) и перейдите на вкладку **IP Pools**. Нажав **Add**, вы указываете все сетевые настройки, которые затем могут использоваться в vApp этого дата-центра. **IP Pool** привязывается к сети (группе портов для VM). Таким образом, указав для VM в vApp использовать конкретную группу портов и в настройках vApp раздачу адресов **Transient**, вы и указываете, какой IP Pool будет ими использоваться. Однако, для того чтобы воспользоваться данным механизмом назначения IP-адресов, придется задействовать сценарий, запускаемый в гостевой ОС при старте, и этот сценарий будет обращаться к свойствам vApp (которые будут передаваться в каждую VM в виде файла xml на подмонтированном iso). Подробности см. по ссылке <http://link.vm4.ru/sepof> и в блоге <http://blogs.vmware.com/vapp>.

Готовый vApp можно экспортировать через меню **File** ⇒ **Export** ⇒ **Export OVF Template**. В едином пакете, с единственным файлом описания в формате ovf будут все ВМ этого vApp и его собственные настройки.

Резюме

vApp – это развитие идей Virtual Appliance для случаев, когда единое решение – это несколько виртуальных машин. Для администраторов vSphere vApp – это средство, в первую очередь внешнее. В том смысле, что мы можем экспортировать загруженные готовые vApp в нашу среду.

Однако если у вас есть своя группа виртуальных машин, которые являются одним решением, вы можете объединить их в собственноручно созданный vApp. Из плюсов вы получите:

- ☐ возможность включения и выключения группы целиком;
- ☐ возможность указывать автостарт, порядок старта и паузы между включениями виртуальных машин именно этой группы.

Обратите внимание. Если у вас есть созданный на ESXi-сервере vApp, а затем вы этот сервер добавляете в кластер DRS, то vApp придется удалить для завершения этой операции. Создавайте vApp после внесения сервера в кластер DRS.



Глава 6. Управление ресурсами сервера. Мониторинг достаточности ресурсов. Живая миграция ВМ. Кластер DRS

В этом разделе поговорим про различные способы более эффективного задействования ресурсов сервера или нескольких серверов ESXi.

Для работающей на сервере ВМ мы можем сделать настройки количества ресурсов, которое ей гарантировано. ESXi выполнит эти настройки с помощью механизмов работы с ресурсами, которые у него есть. Если серверов несколько, мы можем перераспределить нагрузку между ними с помощью vMotion и DRS. Наконец, нам необходимо наблюдать, достаточно ли ресурсов выдается нашим ВМ. Если нет – определять, что является узким местом. Обо всем этом поговорим в данном разделе.

6.1. Настройки распределения ресурсов для ВМ. Пулы ресурсов

Сначала поговорим про настройки, которые позволят гарантировать или ограничить количество ресурсов, выделяемое для одной ВМ или группы ВМ в пуле ресурсов.

6.1.1. Настройки *limit*, *reservation* и *shares* для процессоров и памяти

Для процессоров и памяти виртуальных машин мы можем задавать настройки *limit*, *reservation* и *shares*. По-русски их можно обозвать как «максимум», «минимум» и «доля» соответственно. Поговорим про них по порядку. В конце приведены мои соображения и рекомендации по планированию этих настроек.

***Limit*, *reservation* и *shares* для процессора**

Если вы зайдете в настройки ВМ ⇒ вкладка **Resources**, то увидите настройки ресурсов для этой ВМ. Выделим настройки процессорной подсистемы (рис. 6.1).

Reservation – это количество мегагерц гарантированно закрепляется за данной ВМ в момент ее включения. Обратите внимание: резерв – это блокирующая настройка. Если у сервера недостаточно мегагерц, чтобы обеспечить резерв ВМ, то виртуальная машина не включится с соответствующим сообщением об ошибке (рис. 6.2).

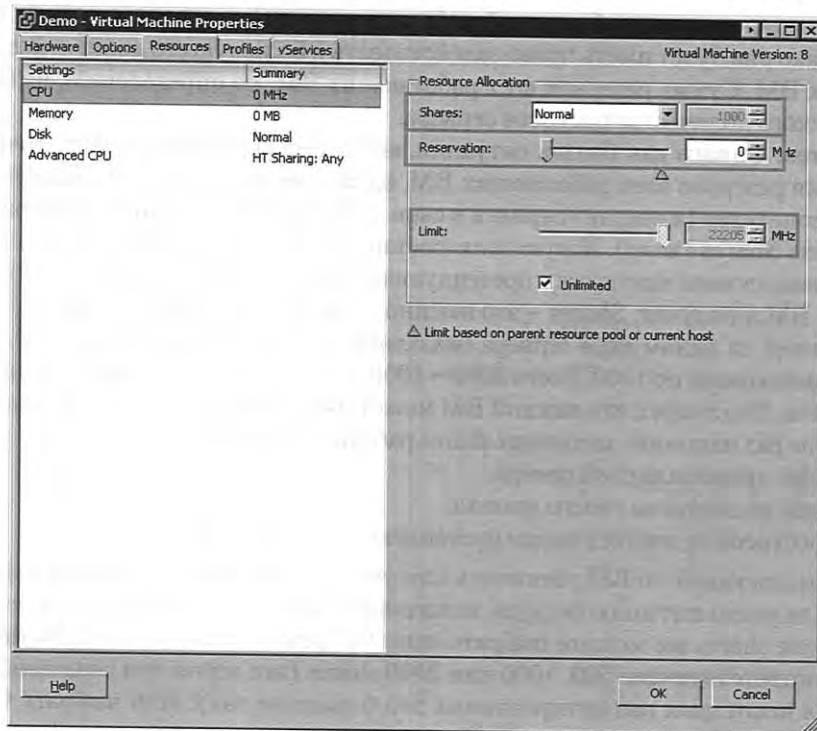


Рис. 6.1. Настройки ресурсов для процессоров ВМ

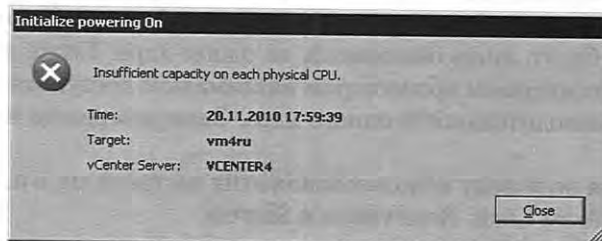


Рис. 6.2. Сообщение о нехватке ресурсов для обеспечения резерва процессора ВМ

Limit. Для процессоров – максимальное количество мегагерц, которое может быть выделено для этой ВМ на все ее процессоры. По умолчанию стоит флажок **Unlimited** – это означает, что искусственно мы не ограничиваем процессорные ресурсы ВМ, они ограничены только физически. Один виртуальный процессор (или, правильнее сказать, одно виртуальное ядро) не может получить мегагерц больше, чем предоставляет одно ядро процессора физического.

Если ВМ не задействует ресурсы процессора, они ей не выдаются. То есть даже если поставить ВМ высокий резерв, в то время как ее нагрузка невелика, то для

нее будет выделяться требуемое ей небольшое количество процессорного времени. Однако она будет иметь право задействовать и большую его долю – «отняв» ее у других ВМ. Сумма резервов всех работающих ВМ, по определению, не больше физического количества ресурсов сервера.

Shares. Однако как быть в ситуации, когда ресурсов сервера достаточно для покрытия резервов всех работающих ВМ, однако недостаточно для удовлетворения их аппетитов (и они не уперлись в свои лимиты)? В таких ситуациях работает настройка **Shares** (доля). Какую долю составляет количество shares одной ВМ относительно суммы shares всех претендующих на ресурс ВМ – такую долю этого ресурса ВМ и получит. Shares – это именно доля, это безразмерная величина.

Пример: на одном ядре сервера оказались три однопроцессорные ВМ. Shares у них одинаковый, по 1000. Всего $3000 = 1000 \times 3$, следовательно, доля любой ВМ – одна треть. Это значит, что каждой ВМ может быть выделена треть ресурсов этого ядра. Еще раз напомним: механизм shares работает, когда ВМ:

- ☐ уже превысила свой резерв;
- ☐ еще не достигла своего лимита;
- ☐ ресурсов не хватает на все претендующие на них ВМ.

Если для какой-то ВМ увеличить или уменьшить shares – ей немедленно увеличат или уменьшат долю ресурса, выключения ВМ для этого не требуется.

В поле shares вы можете выбрать одну из трех констант – **Low**, **Normal** или **High**, соответствующие 500, 1000 или 2000 shares (это верно для созданных вами ВМ, для некоторых импортированных это бывает не так). Или выбрать **Custom** и указать произвольное их число. Данные константы введены для вашего удобства – ведь все равно у вас будут типовые, более и менее важные ВМ.

Обратите внимание на мой пример: «На одном ядре оказались три ВМ...». Это важный нюанс – процессорный ресурс дискретен. Реально бороться за ресурсы процессора ВМ будут, лишь оказавшись на одном ядре. Также для ВМ с одним виртуальным одноядерным процессором максимально доступная производительность – это производительность одного ядра. Задирать резерв или лимит выше бессмысленно.

Соображения по поводу использования этих настроек см. в п. 6.1.3 «Рекомендации по настройкам Limit, Reservation и Shares».

Limit, reservation и shares для памяти

Если вы зайдете в настройки ВМ ⇒ вкладка **Resources**, то увидите настройки ресурсов для этой ВМ. Выделим настройки памяти (рис. 6.3).

На первый взгляд, все точно так же, как и для процессора, но есть нюанс.

Reservation – это количество мегабайт физической оперативной памяти гарантированно закрепляется за данной ВМ в момент ее включения. Обратите внимание: резерв – это блокирующая настройка. Если у сервера недостаточно мегабайт, чтобы обеспечить резерв ВМ, то ВМ не включится с соответствующим сообщением об ошибке.

Reserve all guest memory – этот флажок резервирует 100% памяти, в отличие от ползунка Reservation. Если вы сделали Reservation = 5 Гб, а затем на вкладке

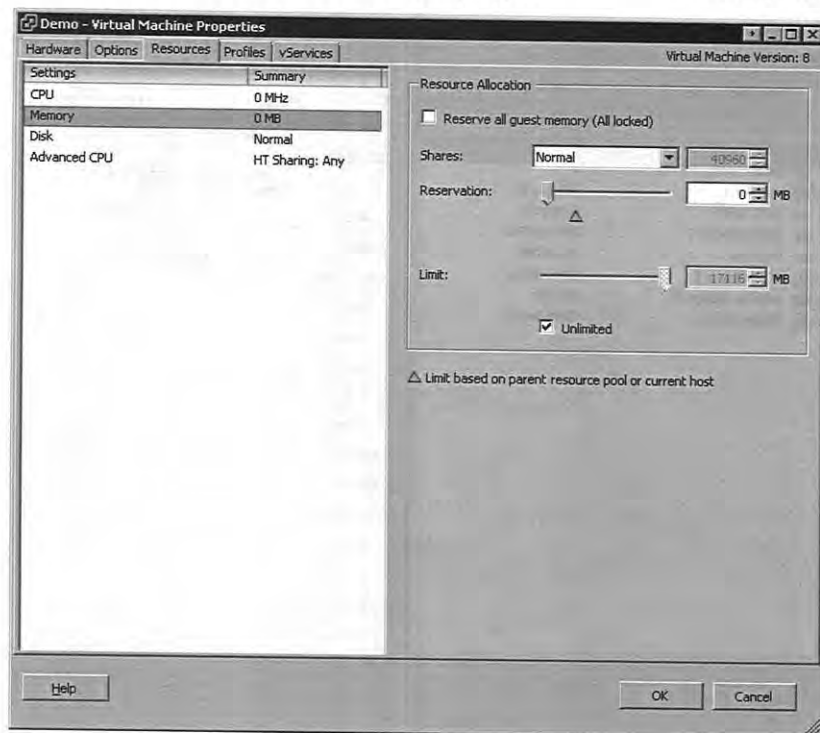


Рис. 6.3. Настройки ресурсов для памяти ВМ

Hardware увеличили размер памяти ВМ до 10 Гб – зарезервированной окажется половина. А этот флажок всегда резервирует 100%, и не важно, сколько это в абсолютных цифрах.

Limit – максимальное количество мегабайт, которое может быть выделено для этой ВМ. Но что означает стоящий по умолчанию флажок **Unlimited**?

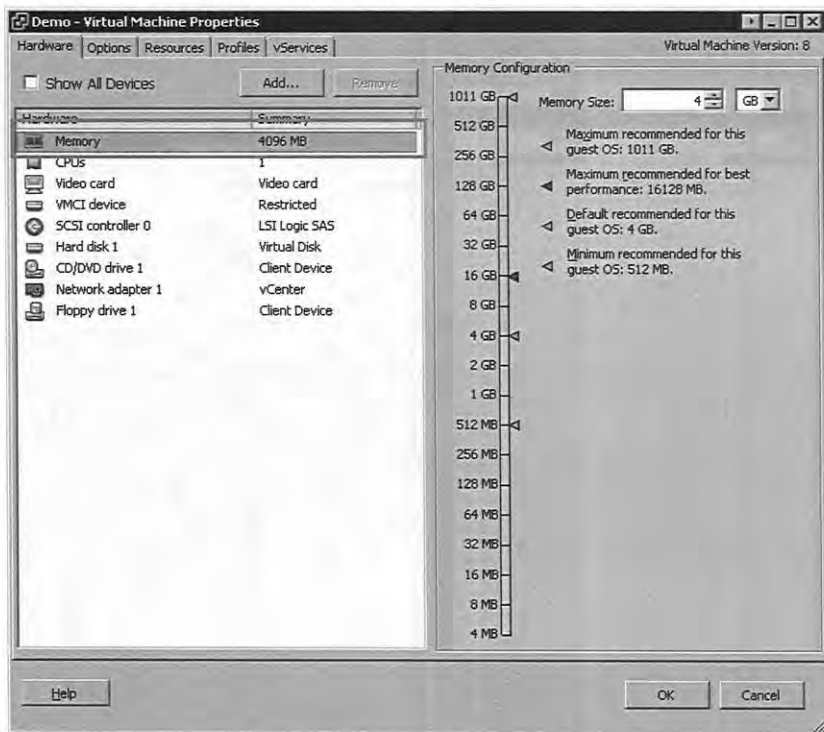
И что за память тогда настраивается на вкладке **Hardware** (рис. 6.4)?

Что означает **Reservation**, если гостевая ОС в любом случае видит весь выделенный ей объем памяти?

С памятью ситуация следующая.

Верхней границей, то есть количеством памяти, которое видит гостевая ОС, является настройка памяти на вкладке **Hardware**. Я ее в дальнейшем буду называть «**hardware memory**», и такое название вам может встретиться в документации. Лицензирование пятой vSphere привязано именно к этой величине, именно ее называют **vRAM** в контексте лицензирования. Таким образом, если вы хотите ограничить ВМ сверху, меняйте не настройку **Limit**, а количество памяти на вкладке **Hardware**.

Reservation – столько мегабайт памяти гарантированно выделяется из физической оперативной памяти. Все, что больше резерва, может быть выделено из файла подкачки.

Рис. 6.4. Настройка **Hardware** памяти

Limit – больше этого количества мегабайт не будет выделено из физической оперативной памяти. Остаток до **hardware memory** обязательно будет выдан из файла подкачки, даже если на сервере нет недостатка в свободной оперативной памяти.

Скорее всего, вы не будете использовать настройку **Limit** на уровне ВМ. Если вам необходимо выделить для ВМ меньше памяти, уменьшайте значение настройки **Hardware memory**. Ситуаций, в которых вам может потребоваться изменение настройки **Limit**, немного. Например, стоит задача протестировать приложение X, у которого жесткие системные требования, и оно отказывается запускаться, если считает, что компьютер им не удовлетворяет (у него меньше А гигабайт ОЗУ). Если у сервера ESXi мало ресурсов, то можно настройкой **hardware memory** указать достаточное для запуска приложения X количество памяти. А настройкой **Limit** ограничить реальное потребление оперативной памяти сервера этой ВМ. Или, как вариант, вы сейчас не хотите выделять какой-то ВМ много памяти, но в будущем это может понадобиться. Для увеличения **Hardware memory** требуется выключение ВМ (за исключением случая использования тех гостевых ОС, которые поддерживают горячее добавление памяти), для увеличения **Limit** – нет. Впрочем, сам я не особо верю в целесообразность использования приведенных примеров.

Shares. Однако как быть в ситуации, когда ресурсов сервера достаточно для покрытия резервов всех работающих ВМ, однако недостаточно для удовлетворения их appetites (и они не уперлись в свои лимиты)? В таких ситуациях работает настройка **Shares** («доля»). Какую долю составляет количество shares одной ВМ относительно суммы shares всех претендующих на ресурс ВМ – такую долю этого ресурса ВМ и получит. Shares – это именно доля, это безразмерная величина.

Обратите внимание: если для процессора константы Low, Normal и High соответствуют 500, 1000 или 2000 shares на ВМ, то для памяти это не так. Для памяти Low, Normal или High соответствуют 5, 10 или 20 shares на каждый мегабайт памяти ВМ.

Пример: на сервере оказались три ВМ. Объем памяти у двух равен 500 Мб, у третьей – 1000 Мб. Shares у них одинаковый, Normal, то есть по 10 на мегабайт (рис. 6.5).



Рис. 6.5. Описание примера распределения долей памяти

Всего shares 20 000. Виртуальные машины А и Б имеют право на до четверти от всей памяти. ВМ В – на половину, при такой же настройке shares. И это логично, так как appetites ВМ В вдвое выше каждой из прочих.

Еще раз напомним – механизм shares работает, когда ВМ:

- ☐ уже превысила свой резерв;
- ☐ еще не достигла своего лимита;
- ☐ памяти не хватает на все претендующие на нее ВМ.

Если для какой-то ВМ увеличить или уменьшить shares, ей немедленно увеличат или уменьшат долю ресурса, выключения ВМ для этого не требуется.

Обратите внимание: если ВМ не задействует память, ESXi не адресует ее для этой ВМ. Посмотреть это можно на вкладке **Summary** для виртуальной машины (рис. 6.6).

Выделена настройка Memory = 1 Гб. Столько памяти видит гостевая ОС, это настройка «Hardware memory». Справа показана «Active Guest Memory» – столько памяти активно использует гостевая ОС. А «Consumed Host Memory» показывает, сколько физической памяти ESXi выделил под данные этой ВМ. В упрощенной формулировке это означает, что ESXi может уменьшить Consumed Memory до Active Memory при необходимости, без ущерба для производительности ВМ.

Если выделить пул ресурсов, vApp, сервер, кластер или датацентр и перейти на вкладку **Virtual Machines**, то подобную информацию можно получить для всех ВМ выделенного объекта (рис. 6.7).

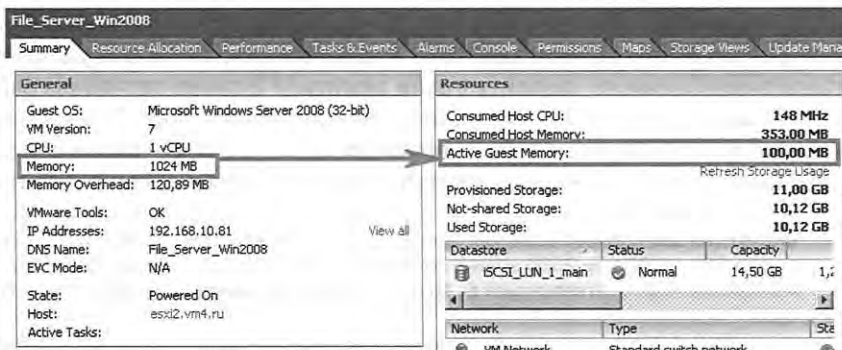


Рис. 6.6. Информация об объемах выделенной и используемой памяти

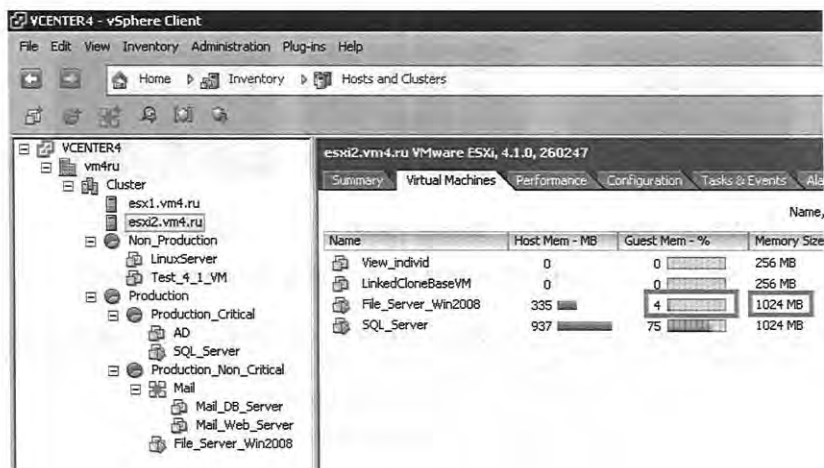


Рис. 6.7. Данные по использованию памяти для всех VM на одном сервере

По умолчанию вы увидите не совсем тот же набор столбцов. Это настраивается в контекстном меню данной вкладки ⇒ **View Column**.

Столбец **Memory Size** – это «hardware memory». Столбец **Guest Mem %** – какой процент памяти активно использует гостевая ОС.

Если у сервера недостаточно памяти для удовлетворения резерва ВМ, то эта ВМ не включится. Если у сервера недостаточно памяти для удовлетворения аппетита ВМ сверх резерва, то недостаток физической оперативной памяти будет компенсирован файлом подкачки. Механизм подкачки используется два – файл подкачки гостевой ОС и файл подкачки VMkernel, создаваемый для каждой виртуальной машины при ее включении. Дальше про эти механизмы я расскажу чуть подробнее, здесь же хочу отметить: при включении ВМ создается файл .vswp. Именно в этот файл ESXi адресует часть памяти ВМ, если памяти физической не хватает. Как велика эта часть? В наихудшем случае это вся память сверх резерва

до hardware memory. Размер файла подкачки как раз такой: «hardware memory» минус reservation. Таким образом, если оставить reservation равным нулю, при включении каждой ВМ создается файл размером с ее оперативную память. Выводов отсюда два:

- ❑ если на хранилище для файлов подкачки (по умолчанию файл подкачки создается в каталоге ВМ) недостаточно места для создания этого файла – ВМ не включится;
- ❑ если вам необходимо освободить сколько-то места на разделах VMFS, один из способов – это увеличение reservation для памяти ВМ: чем больше резерв, тем меньше места резервируется под файл .vswp, файл подкачки VMkernel (альтернатива этому – расположение файлов подкачки VMkernel на отдельном хранилище).

Иллюстрация работы механизма распределения ресурсов на примере памяти

Ситуация:

- ❑ сервер, у сервера 16 Гб памяти. Расход памяти на сам ESXi как ОС, на накладные расходы опустим для простоты;
- ❑ три ВМ. Каждой выделено по 10 Гб памяти (hardware memory):
 - у ВМ А shares = normal, reservation = 0;
 - у ВМ Б shares = normal, reservation = 5 Гб;
 - у ВМ В shares = high, reservation = 0.

Шаг 1 – рис. 6.8. Большая окружность – память сервера, 16 Гб. Три ВМ под маленькой нагрузкой – А и Б активно используют не более 3 Гб памяти, ВМ В –

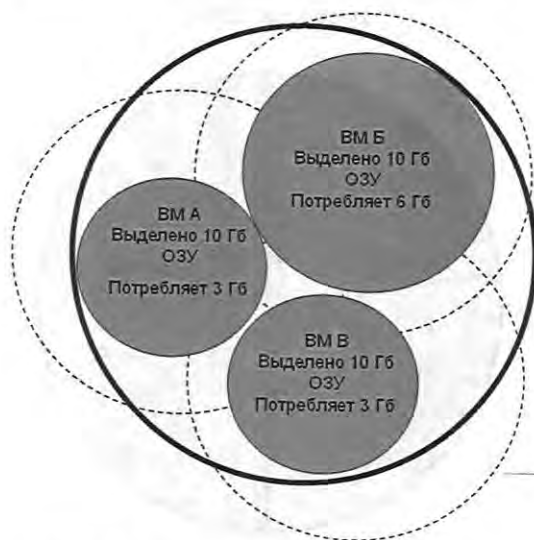


Рис. 6.8. Иллюстрация использования ресурсов. Шаг 1

не более 6 Гб. Ресурсов хватает на всех, поэтому настройки reservation и shares не оказывают влияния на распределение ресурсов.

Здесь пунктирными окружностями показаны 10 Гб, которые номинально выделены каждой ВМ.

Шаг 2 – рис. 6.9. Нагрузка на ВМ возрастает, и памяти сервера на всех уже не хватает. ESXi начинает рассчитывать доли ресурсов.

Шаг 3 – рис. 6.10. Ресурсы памяти разделены в соответствии с reservation и shares. За счет Shares виртуальные машины А и Б имеют право на 4 Гб памяти

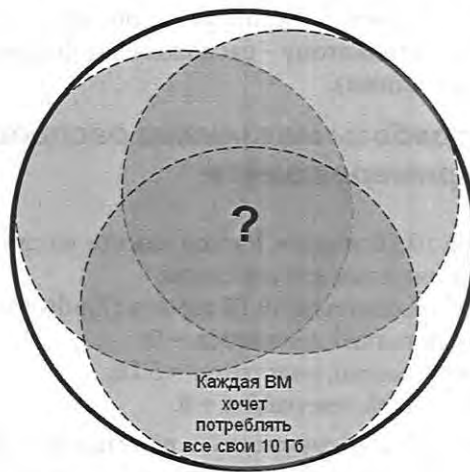


Рис. 6.9. Иллюстрация использования ресурсов. Шаг 2

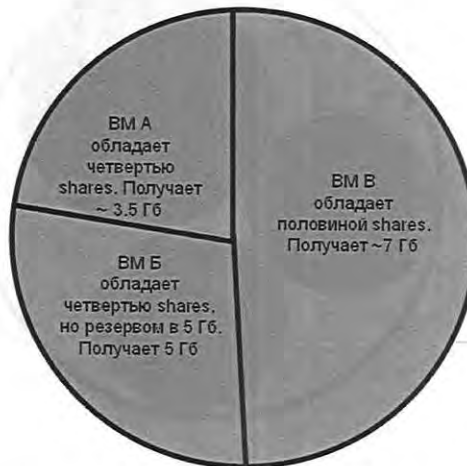


Рис. 6.10. Иллюстрация использования ресурсов. Шаг 3

каждая, а виртуальная машина В – на 8 Гб. Но виртуальной машине В досталось 5 Гб, так как такое значение имеет настройка reservation для этой ВМ. Оставшиеся 11 Гб делятся между ВМ А и ВМ В с учетом их shares. Всю недостающую память ВМ получают из файлов подкачки.

Соображения по поводу использования пулов ресурсов см. в п. 6.1.3 «Рекомендации по настройкам Limit, Reservation и Shares».

6.1.2. Пулы ресурсов

Настройки Limit, Reservation и Shares для процессора и памяти можно задавать на уровне ВМ. Можно, но неинтересно. Сколько у вас виртуальных машин? Сотни? Несколько десятков? Десятки?

Даже если десяток-другой – их число будет изменяться. Какие-то ВМ создаются, какие-то удаляются, какие-то клонируются и размножаются. Отслеживать эти настройки для каждой из них неудобно и утомительно.

Намного естественнее выполнять эти настройки для групп виртуальных машин. В этом и состоит суть пулов ресурсов.

Создание пула ресурсов состоит из единственного шага: пройдите **Home** ⇒ **Inventory** ⇒ **Hosts and Clusters** и в контекстном меню сервера или DRS-кластера выберите **New Resource pool**. Откроется единственное окно настроек (рис. 6.11).

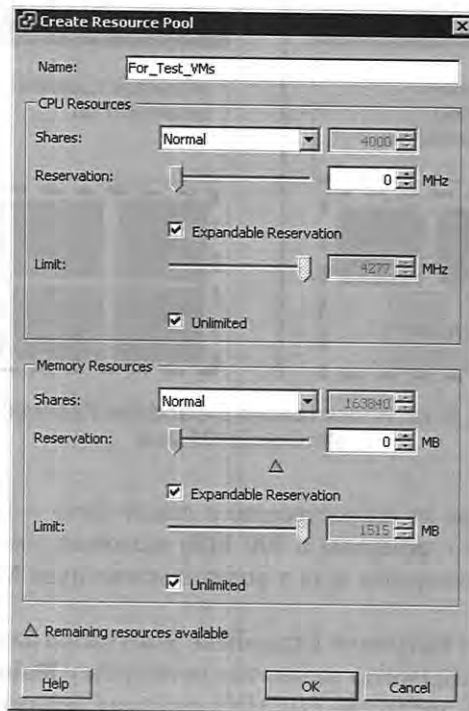


Рис. 6.11. Настройки пула ресурсов

Как видно, настройки пула ресурсов такие же, как настройки распределения ресурсов для ВМ. Это Limit, Reservation и Shares для процессора и памяти. Единственное отличие от настроек ВМ – наличие флажка **Expandable Reservation**. Если флажок стоит, то пул ресурсов может «одалживать» свободные reservation у родительского пула. Объясню эту настройку на примере:

Вы создали пул ресурсов «Main», а в нем – два дочерних, «Child 1» и «Child 2». В дочерние пулы были помещены какие-то ВМ, притом для этих ВМ вы планируете указать reservation. Для того чтобы ВМ с резервом включилась, необходимо, чтобы у пула ресурсов, в котором она находится, были свои reservation в достаточном количестве (рис. 6.12).

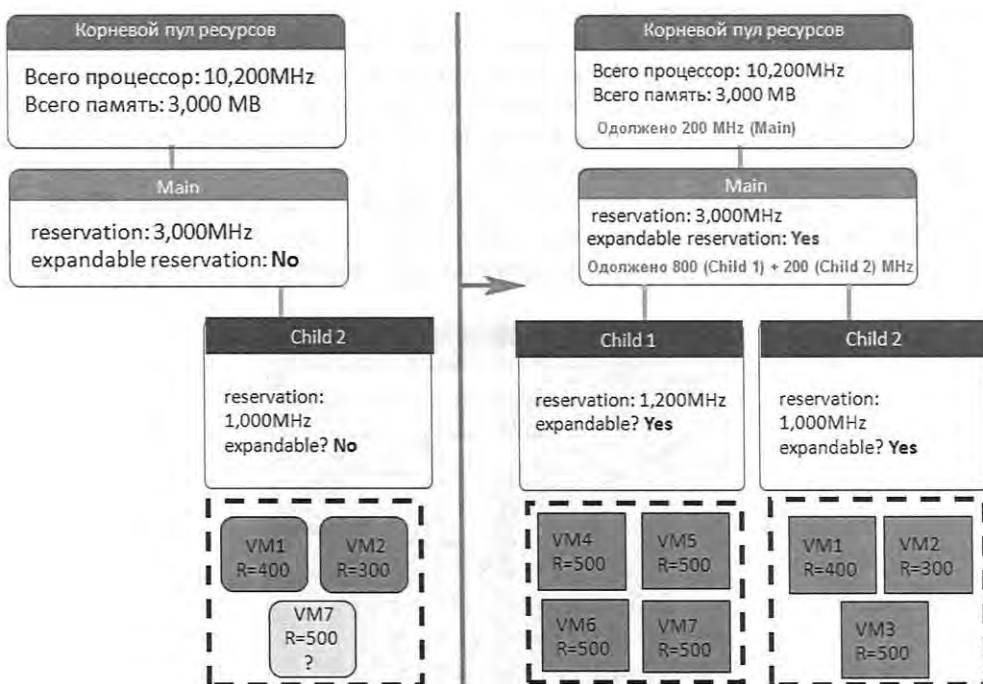


Рис. 6.12. Иллюстрация Expandable Reservation
Источник: VMware

Обратите внимание на иллюстрацию в левой части – в пуле Child 2 виртуальную машину VM7 с резервом в 500 МГц включить уже не удастся. Однако свободные 800 МГц reservation есть у родительского пула Main – и они никак не задействуются.

В правой части мы включили Expandable Reservation для пула Child 2 и Main. Теперь Child 2 смог «одолжить» незанятые мегагерцы у Main. А когда «одолжить» захотел еще и Child 1, тогда уже сам Main одолжил ресурсов у своего родительского пула.

Получается, что если резерв для пула должен быть жестко ограничен, то Expandable Reservation включать не надо. Зато включенный, он позволяет не рассчитывать точное количество reservation для дочерних пулов: если им не хватит своих – одолжат у родительского пула.

Пулы ресурсов можно создавать для сервера вне кластера или для DRS-кластера. Важно! Если ваши сервера в кластере без функции DRS, то ни для серверов, ни для кластера пулы создать будет нельзя. Пулы ресурсов могут быть вложены друг в друга. В данном контексте vApp тоже является пулом ресурсов (рис. 6.13).

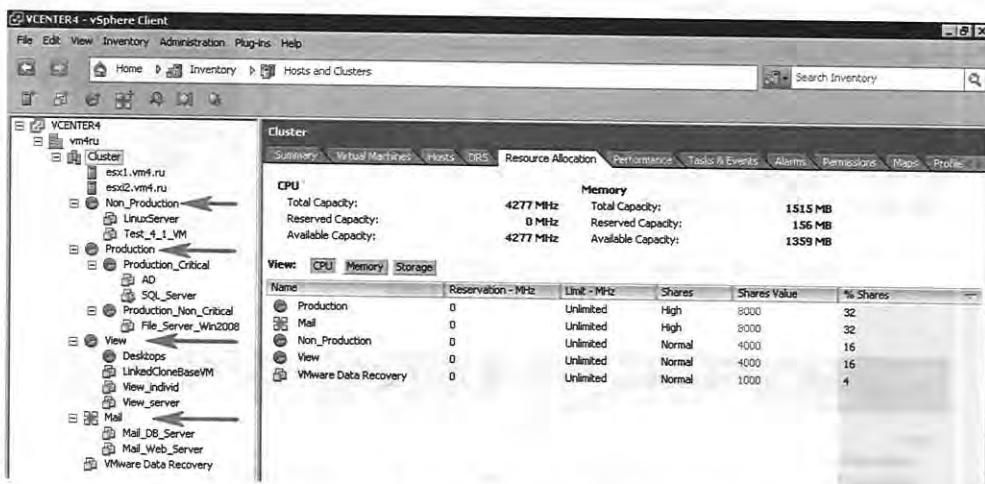


Рис. 6.13. Схема пулов ресурсов для кластера DRS

На этом рисунке я выделил пулы ресурсов, находящиеся на одном уровне. Обратите внимание, что виртуальная машина VMware Data Recovery находится на одном уровне с пулами ресурсов, для них родительским объектом является кластер. Это означает, что в случае борьбы за ресурсы эта ВМ будет бороться с пулами.

Те же виртуальные машины, которые находятся в пуле ресурсов, отсчитывают свою долю от ресурсов пула (рис. 6.14).

В данном примере указано, что все ВМ работают на одном ядре. Это допущение сделано для простоты данной иллюстрации. Связано оно с тем, что работа на одном ядре – обязательное условие того, что между ВМ возникает борьба за процессорный ресурс. В общем случае предполагается, что виртуальных машин у нас по несколько на каждое ядро сервера, и от такой борьбы мы никуда не денемся. В ином случае борьбы за процессорные ресурсы не будет, и эти механизмы не нужны.

Обратите внимание на вкладку **Resource Allocation** для пула ресурсов, рис. 6.15.

Эта вкладка – хороший источник информации по настройкам limit, reservation, shares для дочерних объектов пула ресурсов, сервера или кластера. Особенно

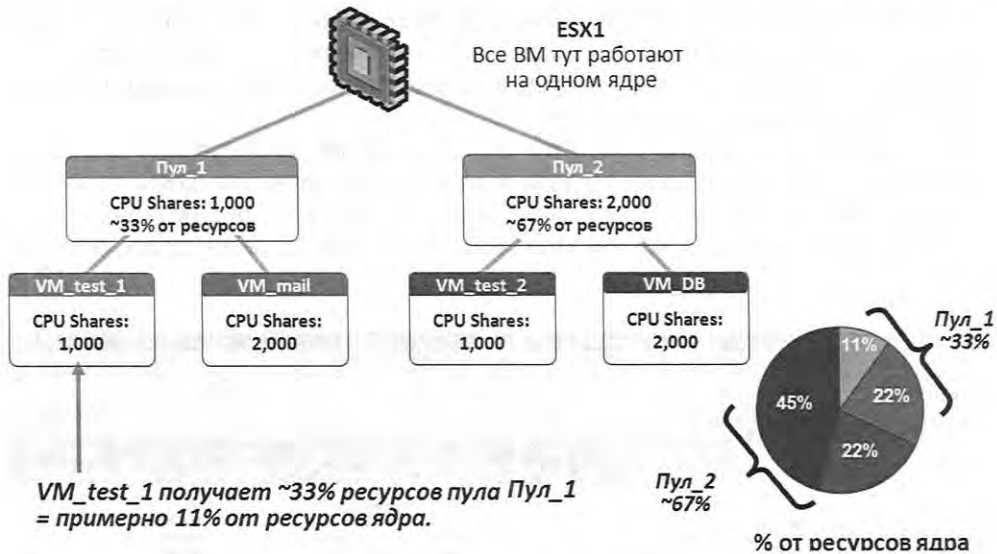


Рис. 6.14. Иллюстрация распределения ресурсов ВМ в пуле
Источник: VMware

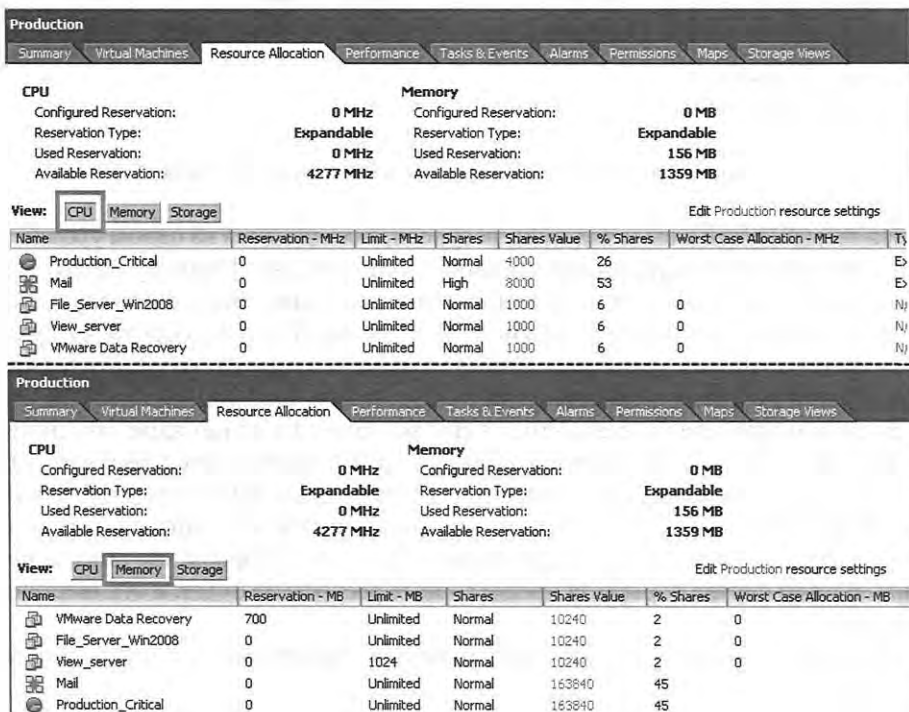


Рис. 6.15. Вкладка **Resource Allocation**

обратите внимание на столбец **Shares Value** – он показывает посчитанную долю каждого из дочерних объектов одного уровня. Важно – пул ресурсов «Production_Critical», vApp «Mail» и несколько VM находятся на одном уровне, они дочерние объекты пула «Production». И они борются за ресурсы по тем же правилам, по каким боролись бы между собой объекты какого-то одного типа. Столбцы Reservation, Limit и Shares являются активными, то есть значения в них можно менять прямо с этой вкладки.

Кнопка **Storage** появилась лишь в версии 4.1 – она является интерфейсом к механизму Storage IO Control. Данный механизм работает на уровне виртуальных машин одного хранилища, не на уровне пулов ресурсов.

Соображения по поводу использования пулов ресурсов см. в следующем разделе.

6.1.3. Рекомендации по настройкам *Limit*, *Reservation* и *Shares*

Основной идеей мне кажется следующая: ситуаций, когда вам приходится эти настройки, следует избегать. Чуть ранее я уже приводил иллюстрацию – см. рис. 6.16.

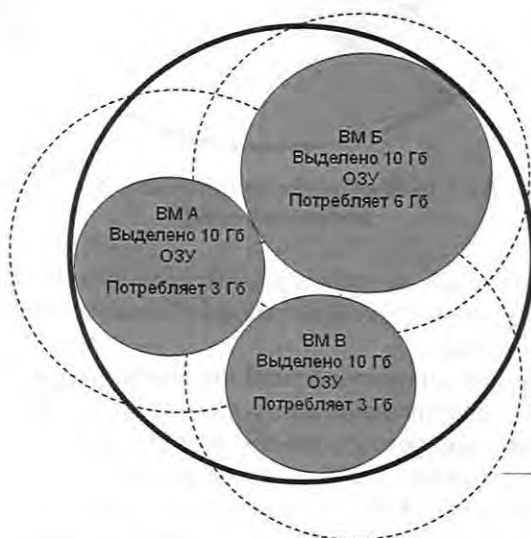


Рис. 6.16. Иллюстрация недостаточного количества ресурсов сервера

Здесь вы видите, что ресурсов сервера (большая окружность) достаточно для удовлетворения текущих аппетитов VM (маленькие круги). Однако их недостаточно для одновременного удовлетворения максимально возможных аппетитов (три пунктирные окружности). Если у нас нет твердой уверенности в том, что эти

виртуальные машины не будут требовать максимума своих ресурсов одновременно, то эта ситуация неправильна. Правильной ситуацией является та, когда ресурсов сервера заведомо больше, чем необходимо для всех ВМ (рис. 6.17).

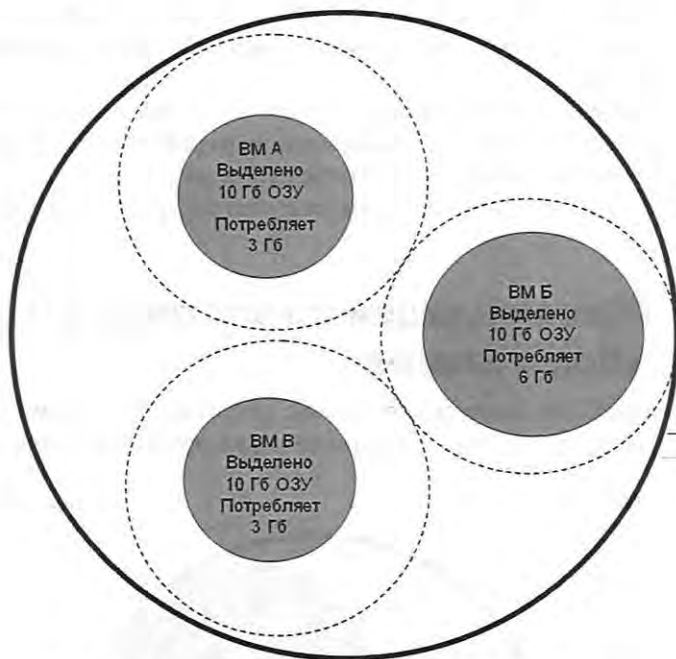


Рис. 6.17. Иллюстрация достаточного количества ресурсов сервера

Если у вас ситуация как на втором рисунке, то Limit, Reservation и Shares вам особо и не нужны, и именно к такой ситуации вы должны стремиться при сайзинге сервера/серверов под ESXi.

Отлично, этот момент принят во внимание, конфигурация и количество серверов достаточны для удовлетворения нагрузки со стороны ВМ. Проблема в том, что количество серверов может резко уменьшиться в результате сбоя. Это не проблема для доступности приложений в ВМ – при наличии разделяемого хранилища мы легко можем перезапустить ВМ с отказавшего сервера на остальных за минуты. А при наличии настроенного кластера с функцией VMware High Availability это произойдет автоматически.

Таким образом, возможна ситуация, когда все наши ВМ работают на меньшем количестве серверов – и ресурсов на всех может уже не хватить.

Вывод: нам следует воспользоваться настройками распределения ресурсов для подготовки к такой ситуации. Задача состоит в том, чтобы при нехватке ресурсов они в первую очередь доставались критичным и важным виртуальным машинам за счет отъема их у неважных.

Когда ресурсов в достатке

Все-таки как мы можем использовать эти настройки, когда ресурсов в достатке:

- ❑ может быть неплохой идеей создать пул ресурсов с установленным `limit` для некритичных (тестовых, временных) ВМ, от которых можно ожидать всплесков нагрузки или нагрузка со стороны которых мало прогнозируема;
- ❑ если у вас запрашивают сервер с большими ресурсами памяти/процессора, притом вы твердо знаете, что запрашивают лишнего, то можно с помощью `limit` указать достаточное количество ресурсов. Плюсом является то, что вам не нужно спорить и доказывать завышенные требования к ресурсам, и то, что если больше ресурсов дать все-таки захотите, то достаточно будет поднять `limit`, и ВМ без перезагрузки сможет воспользоваться дополнительными ресурсами;
- ❑ увеличение `reservation` для оперативной памяти уменьшает размер файла подкачки, который ESXi создает при включении каждой ВМ;
- ❑ если политика использования виртуальной инфраструктуры предполагает выделение под какие-то задачи фиксированного количества ресурсов, то имеет смысл создать пул ресурсов, в настройках которого и зафиксировать максимальное (`limit`) и минимальное гарантированное (`reservation`) количество ресурсов. Классический пример таких задач – хостинг ВМ, когда клиент платит за некую производительность, и в наших интересах сделать так, чтобы он получал то, за что заплатил, и не больше.

Когда ресурсов не хватает

Типовые рекомендации таковы.

Создавайте пулы ресурсов для ВМ разной важности. Кроме того, пулы ресурсов часто являются удобными объектами для назначения прав доступа – поэтому для ВМ одной важности может быть создано несколько пулов, потому что эти ВМ относятся к разным администраторам. Не забывайте, что пулы ресурсов могут быть вложенными – это иногда удобно, в основном из организационных соображений. Однако избегайте создания пулов только для раздачи прав – для этого правильнее (и часто удобнее) использовать голубые каталоги в иерархии **Virtual Machines and Templates**.

Какими настройками лучше манипулировать – `shares` или `reservation/limit`? В типовых случаях лучше `shares`, потому что это не блокирующая настройка. Напомню, что если недостаточно ресурсов для обеспечения `reservation`, ВМ не включится. В случае `shares` таких проблем гарантированно не будет.

`Reservation` может иметь смысл давать для критичных ВМ, для того чтобы гарантировать ресурсы для их работы в случае нехватки ресурсов. Однако зачастую не стоит резервировать весь объем памяти для них. Для виртуальных машин гипервизор предоставляет нам счетчики `Active Memory` – к такому объему оперативной памяти виртуальная машина обращается активно, часто. Обычно имеет смысл резервировать среднее значение `Active Memory` за период времени не менее недели. Это гарантирует, что достаточно большой объем памяти для ВМ будет выделен в любом случае.

Чем меньше reservation, тем меньше вероятность того, что ВМ не включится из-за нехватки ресурсов для обеспечения этого reservation. Еще ВМ с высоким reservation может быть минусом для кластера НА, см. посвященный ему раздел.

Не забывайте, гарантия ресурсов для одних ВМ означает гарантированное вытеснение в файл подкачки и недостаточность процессорных тактов для других, **в случае когда ресурсов перестанет хватать на всех.**

Если ВМ не потребляет много ресурсов, но достаточно критична (например, контроллер домена), может иметь смысл зарезервировать небольшие потребляемые ей ресурсы целиком или близко к тому.

Если какая-то ВМ критична для нас и критичен уровень отклика этой ВМ (то есть скорость работы) – для нее имеет смысл зарезервировать все или большую часть выделяемых ресурсов.

Итак, сводный план действий примерно такой:

1. Думаем, нужна ли нам эта схема распределения ресурсов. Может быть (а для многих из вас – скорее всего), даже выход из строя сервера-другого не приведет к недостатку ресурсов для ВМ. Если так – пулы ресурсов не используем, если на это нет организационных причин.
2. Создаем пулы ресурсов для ВМ разной степени критичности.
3. Соответственно критичности настраиваем shares: High – для критичных пулов, Low – для некритичных. Оставляем Normal для всех остальных. Если есть пулы для тестовых ВМ, пулы, ВМ в которых создаются и удаляются не нами, – может иметь смысл для них поставить limit, чтобы эти ВМ не задействовали слишком много ресурсов.
4. При необходимости гарантировать каким-то ВМ уровень ресурсов настраиваем для них reservation. Это потребует настроить reservation для пулов, в которых они находятся.

Не настраивайте reservation пулов ресурсов «впритык». Вернитесь к рис. 6.14 – сумма reservation дочерних объектов меньше, чем reservation родительского пула, и так поступать правильно.

6.1.4. Storage IO Control, SIOC для дисковой подсистемы

Начиная с версии 4.1 сервера ESXi получили возможность управлять распределением количества операций ввода-вывода в секунду между виртуальными машинами.

Мы можем оперировать двумя настройками – limit и shares, эти настройки делаются на уровне диска виртуальной машины.

Limit здесь – явное число операций ввода-вывода в секунду, которое может получить этот диск виртуальной машины как максимум. По умолчанию limit не задан. Limit указывается в операциях ввода-вывода в секунду. Если вам удобнее оперировать Мб/сек, то, поделив необходимое число Мб/сек на размер одной операции ввода-вывода (которым оперирует данная ВМ), вы получите искомое

количество IOPS. Например, для получения максимум 10 Мб/сек для виртуальной машины, оперирующей блоками в 64 Кб, укажите ее диску limit, равный 160 IOPS.

Обратите внимание. Если пользоваться настройкой Limit, то она должна быть указана для всех дисков виртуальной машины. Если это не так, то настройка применяться не будет.

Shares – как и для других ресурсов, эта величина является долей ресурса, здесь это IOPS.

Данный механизм включается для отдельного хранилища VMFS. Притом работает он не все время, а лишь по срабатыванию условия – превышение пороговой величины задержки (Latency). Когда значение этого счетчика превышает указанную нами пороговую величину, настройки Limit и Shares применяются. И в соответствии с этими настройками перераспределяются дисковые операции хранилища между дисками виртуальных машин, на нем расположенных.

Давайте пройдемся по фактам о настройке и эксплуатации этого механизма.

SIOC выключен по умолчанию. Сделано это по той причине, что далеко не все лицензии vSphere позволяют задействовать данную функцию.

Пороговое значение Latency по умолчанию – 30 миллисекунд. Притом вычисляется и оценивается среднее значение Latency для всех серверов, обращающихся к хранилищу VMFS.

Для включения SIOC необходим vCenter. Для работы SIOC vCenter не является необходимым. Сервера записывают необходимые данные на само хранилище, в первую очередь это значения Latency и настройки Limit/Shares виртуальных дисков. Исходя из этих данных, работает соответствующая служба на каждом ESXi. Эта служба называется «PARDA Control Algorithm», и основные ее компоненты – это «latency estimation» и «window size computation». Первый используется для оценки Latency относительно порогового значения (это делается каждые 30 секунд), второй высчитывает необходимую глубину очереди для сервера. При чем тут глубина очереди?

Дело в том, что именно за счет динамического изменения этой самой глубины и реализовано разделение IOPS хранилища между виртуальными машинами с разных серверов. Сравните рис. 6.18 и 6.19. Первый из них иллюстрирует ситуацию без SIOC – когда два сервера обращаются на хранилище и полностью его нагружают, то сначала система хранения делит операции ввода-вывода между серверами поровну. Затем каждый сервер может поделить обрабатываемые хранилищем для него IOPS в требуемой пропорции между своими виртуальными машинами.

А вот на следующем рисунке показана та же ситуация, но уже с SIOC.

Как видим, ESXi 2 пропорционально уменьшил глубину своей очереди (до 16), благодаря чему система хранения предоставила для него (его виртуальных машин) пропорционально меньшую долю операций ввода-вывода данного LUN. ESXi уменьшает глубину очереди с таким расчетом, чтобы операции ввода-вывода

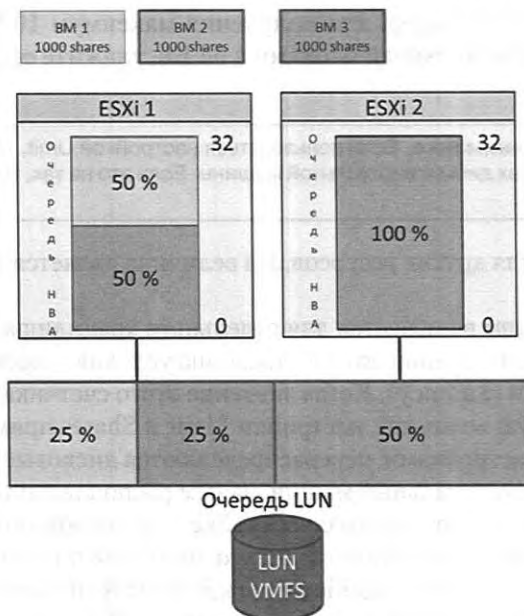


Рис. 6.18. Распределение операций ввода-вывода между VM разных серверов без SIOC

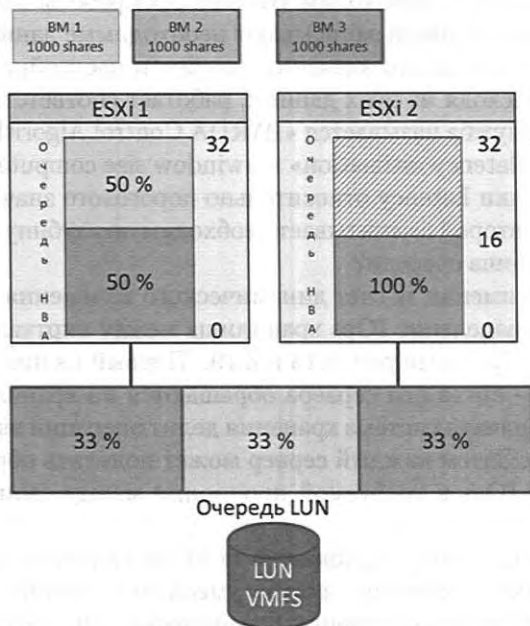


Рис. 6.19. Распределение операций ввода-вывода между VM разных серверов при включенном SIOC

делились между виртуальными машинами с одного хранилища VMFS (LUN) пропорционально их Shares. Глубина очереди не может стать меньше 4.

В пятой версии vSphere этот механизм работает и для NFS-хранилищ.

Обратите внимание. Для VMFS с extend данный механизм не работает. Для RDM SIOC не работает. Если после включения SIOC изменилось число серверов, работающих с хранилищем, то данную функцию следует выключить и включить заново.

SIOC включается на уровне хранилища. Включение этой функции весьма тривиально: выберите один из серверов, которому доступно интересующее вас хранилище VMFS ⇒ **Configuration** ⇒ **Storage** ⇒ выбираем хранилище VMFS ⇒ **Properties**. Нас интересует флажок **Storage I/O Control**. По кнопке **Advanced** нам доступно изменение порогового значения Latency, по достижении которого механизм SIOC вмешивается в распределение операций ввода-вывода между виртуальными машинами. В документе «Storage I/O Control Technical Overview and Considerations for Deployment» (<http://www.vmware.com/resources/techresources/10118>) VMware приводит рекомендации по рекомендуемым значениям Latency в зависимости от типа хранилища. Например, для хранилища с накопителями SSD нормальным считается Latency = 10–15 мс.

Я предполагаю, что это значение можно корректировать, анализируя данные своей инфраструктуры. Если вчера на производительность ВМ не жаловались, а сегодня жалуются – то стоит сравнить Latency, и если сегодня величина задержек больше, то сделать выводы о нормальных и ненормальных значениях.

VMware рекомендует включать SIOC. Этот механизм будет полезен даже с настройками по умолчанию на уровне виртуальных машин – тем, что какая-то одна виртуальная машина не сможет задействовать всю производительность LUN при всплеске активности.

Обратите внимание на два элемента интерфейса, полезных при работе с SIOC (рис. 6.20 и 6.21).

Production

SummaryVirtual MachinesResource AllocationPerformanceTasks & EventsAlarmsPermissionsMapsStorage Views

CPU

Configured Reservation:0 MHz
Reservation Type:Expandable
Used Reservation:0 MHz
Available Reservation:4277 MHz

Memory

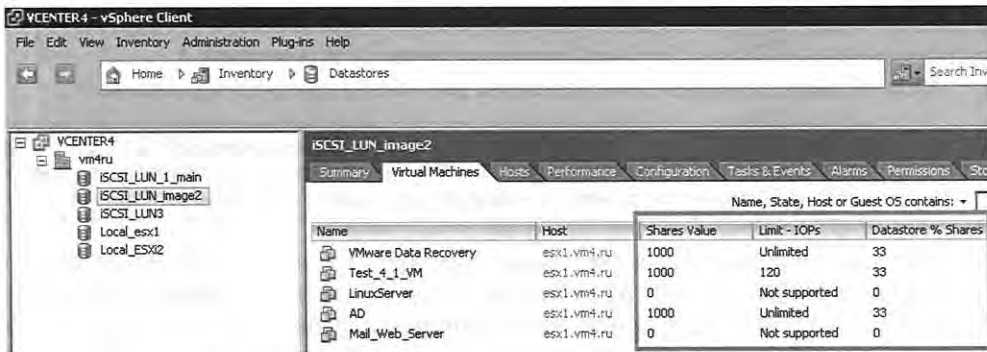
Configured Reservation:0 MB
Reservation Type:Expandable
Used Reservation:156 MB
Available Reservation:1359 MB

View:CPUMemoryStorage

Edit Production resource settings

Name	Disk	Datastore	Limit - IOPS	Shares	Shares Value	Datastore % Shares
 VMware Data Recovery	Hard disk 1	iSCSI_LUN_image2	Unlimited	Normal	1000	50
 File_Server_Win2008	Hard disk 1	iSCSI_LUN_1_main	300	Normal	1000	100
 View_server	Hard disk 1	Local_esx1	Unlimited	Normal	1000	50
 Mail_DB_Server	Hard disk 1	Local_esx1	Unlimited	Normal	1000	50
 SQL_Server	Hard disk 1	Local_ESX2	Unlimited	Normal	1000	100
 AD	Hard disk 1	iSCSI_LUN_image2	Unlimited	Normal	1000	50

Рис. 6.20. Вкладка **Resource Allocation** ⇒ **disk**

Рис. 6.21. Вкладка **Virtual Machines** для Datastores

6.1.5. Network IO Control, NIOC и traffic shaping для сети

Для управления ресурсами сетевой подсистемы серверов ESXi существуют три механизма: группировка контроллеров (NIC Teaming), traffic shaping и появившийся в версии 4.1 Network IO Control (NIOC).

Когда к одному вКоммутатору подключены несколько физических сетевых контроллеров, тогда ВМ с этого вКоммутатора могут использовать пропускную способность сразу нескольких из них. Более того, в некоторых ситуациях даже одна-единственная ВМ способна задействовать сразу несколько каналов во внешнюю сеть (в зависимости от собственной конфигурации и от метода балансировки нагрузки на виртуальном коммутаторе). О различных вариантах балансировки нагрузки рассказано в главе 2.

Там же, на виртуальном коммутаторе или группе портов, настраивается traffic shaping.

В случае использования стандартных виртуальных коммутаторов мы можем настроить ширину канала для исходящего трафика.

В случае использования распределенных виртуальных коммутаторов VMware ширину канала можно ограничивать и для входящего, и для исходящего трафика.

Напоминаю, что применяется эта настройка лишь к тому трафику, что проходит через физический сетевой контроллер. Следовательно, с помощью этого механизма нельзя ограничить канал между виртуальными машинами на одном сервере. Подробнее о работе этого механизма рассказано в главе 2.

В распределенном коммутаторе начиная с версии 4.1 появилось управление распределением сетевого ресурса между задачами ESXi – Network IO Control, NIOC.

Этот механизм предназначен для ситуаций, когда все или большая часть источников трафика ESXi разделяет один набор 10-гигабитных физических сетевых интерфейсов (для гигабитной сети механизм NIOC, скорее всего, не окажет ощутимого влияния). В этом случае возможна ситуация, когда (например) запущенная живая миграция негативно влияет на (например) производительность iSCSI. С помощью NIOC мы избежим такого рода негативных влияний.

Говоря более конкретно, появилась возможность указывать Limit и Shares для типов трафика. Обратите особое внимание – именно для типов трафика, не для отдельных групп портов.

Этот механизм выделяет следующие виды трафика:

- ☐ виртуальные машины. Притом для виртуальных машин мы можем выделять отдельные группы портов и приоритезировать их – это нововведение пятой версии распределенных коммутаторов;
- ☐ Fault Tolerance;
- ☐ iSCSI;
- ☐ NFS;
- ☐ Management;
- ☐ vMotion.

vSphere Replication – это трафик функции, которая добавляется продуктом VMware Site Recovery Manager.

Указав Limit, мы указываем количество Мб/сек, которые как максимум получают соответствующий тип трафика сразу для всех каналов во внешнюю сеть виртуального коммутатора, через которые этот трафик может покидать ESXi. Ограничивается лишь исходящий за пределы ESXi трафик.

Shares – это указание доли, которую получает тот или иной тип трафика при недостатке пропускной способности сети. По факту является минимальной гарантированной пропускной способностью. Доля (shares) высчитывается для каждого канала во внешнюю сеть (физического сетевого контроллера) независимо.

Для настройки этого механизма пройдите **Home** ⇒ **Inventory** ⇒ **Networking** ⇒ распределенный виртуальный коммутатор ⇒ вкладка **Resource Allocation**. Сначала в пункте **Properties** необходимо включить саму функцию NIOC, затем мы получим возможность изменять настройки Limit и Shares для трафика разных типов (рис. 6.22).

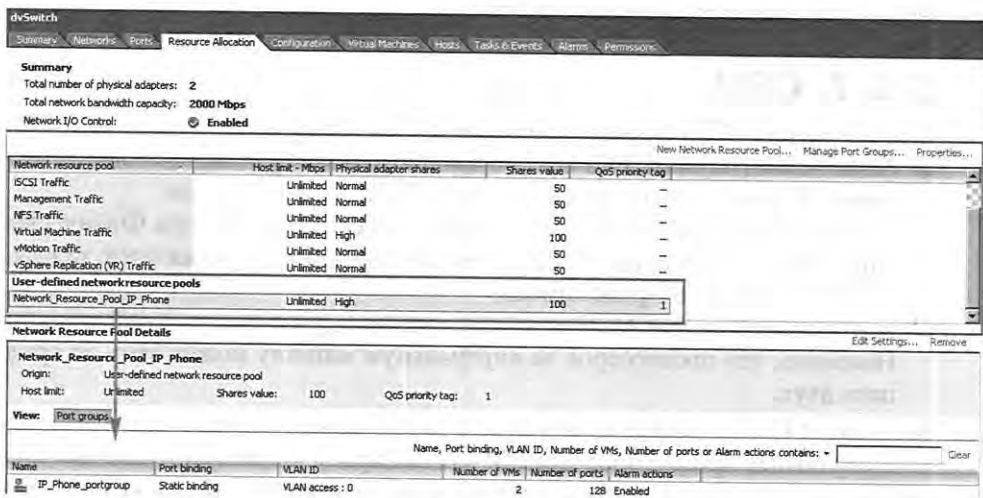


Рис. 6.22. Настройка NIOC

Каждый сервер ESXi рассчитывает доли трафика независимо – это важно по той причине, что у разных серверов может быть разная конфигурация (здесь – разное количество физических сетевых контроллеров).

Начиная с версии 5 появилась ссылка **New Network Resource Pool** – нажав на нее, вы можете определить «пул сетевых ресурсов». В рамках этого пула вы указываете Limit и Shares, а также тэг приоритизации трафика, QoS. А затем по ссылке **Manage Port Groups** вы сможете выбрать группы портов этого распределенного коммутатора, к которым применить параметры пула. По смыслу такой пул является возможностью настроить Limit, Shares и QoS сразу для нескольких групп портов.

VMware предлагает некоторые рекомендации:

- ☐ использовать Shares, нежели Limit, так как Shares является более гибкой настройкой;
- ☐ использовать новый тип балансировки нагрузки – LBT, Load Based Teaming;
- ☐ если вы примете решение использовать еще и Traffic Shaping, то будет удобно, если источник трафика каждого типа будет помещен в отдельную группу портов на распределенном коммутаторе.

Дополнительные подробности следует искать в документе Network IO Control – <http://www.vmware.com/resources/techresources/10119>.

6.2. Механизмы перераспределения ресурсов в ESXi

Limit, shares, reservation – это настройки, которыми мы указываем, как распределять ресурсы. А теперь поговорим о том, благодаря каким механизмам ESXi умеет эти настройки претворять в жизнь и как устроено распределение ресурсов сервера между виртуальными машинами.

6.2.1. CPU

С точки зрения ESXi, процессоры бывают трех типов:

- ☐ физические (physical, PCPU). Это именно процессоры, иногда говорят «сокеты». В зависимости от их количества ESXi лицензируется;
- ☐ логические (logical, LCPU). Это ядра физического процессора. Физические ядра или, в случае hyperthreading, ядра логические. Ключевая идея: каждый LCPU – это одна очередь команд;
- ☐ виртуальные (virtual, VCPU). vCPU – это процессор виртуальной машины. Напомню, что процессоров на виртуальную машину может быть до тридцати двух.

Внимание! Снова сделаю оговорку – в пятой версии vSphere при настройке числа виртуальных процессоров для ВМ мы выбираем число «виртуальных процессоров» и «число ядер в каждом виртуальном процессоре». В строке ниже нам покажут произведение этих чисел – столько потоков сможет использовать данная

виртуальная машина. Так вот, в этом разделе (да и почти везде в этой книге) я под vCPU буду понимать именно эти потоки. То есть если у виртуальной машины один «виртуальный сокет», и он восьмиядерный, и если у виртуальной машины восемь «одноядерных виртуальных сокетов», в данном контексте мы будем говорить, что у нее восемь vCPU – потому что в обоих случаях с точки зрения производительности мы получим восемь идентичных потоков.

Самое первое, что необходимо сказать:

- ❑ один vCPU работает на одном LCPU. То есть виртуальная машина с одним виртуальным процессором работает на одном ядре. Следовательно, даже если у вас однопроцессорный восьмиядерный сервер, на котором работает только одна VM с одним процессором, – она задействует только одно ядро;
- ❑ а вот если эта VM с восемью vCPU, то она задействует все восемь ядер – ESXi в обязательном порядке разводит по разным ядрам процессоры одной VM;
- ❑ на одном ядре может выполняться несколько vCPU разных виртуальных машин (рис. 6.23).

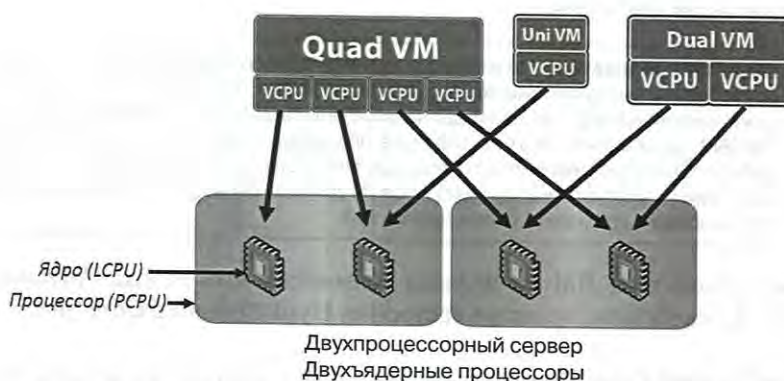


Рис. 6.23. Иллюстрация работы с процессорной подсистемой
Источник: VMware

На иллюстрации показан двухпроцессорный сервер, каждый процессор двухъядерный.

Гипервизор осуществляет балансировку нагрузки, с интервалом в 20 миллисекунд может переносить vCPU между LCPU для лучшей их утилизации.

Операционная система ожидает, что все ее процессоры будут работать одновременно. В случае физических серверов так и происходит. Но в случае виртуализации процессоры, видимые гостевой ОС, являются виртуальными процессорами, которыми управляет гипервизор. В частности, гипервизор перераспределяет ресурсы физических процессоров (ядер) между многими процессорами виртуальными. И именно из-за того, что на каждом физическом ядре работают несколько виртуальных процессоров, гипервизору и неудобно выделять такты виртуальным процессорам одной VM одновременно (ведь нагрузка на разные физические ядра

разная, где-то в данный момент есть свободные такты, где-то нет). А если выделять их не одновременно, то гостевые ОС как минимум будут получать штрафы по производительности, а как максимум – падать в BSOD и Kernel panic.

Для гипервизоров предыдущих поколений эта проблема решалась тем, что гипервизор отслеживал «рассинхронизацию», то есть ситуацию, когда какие-то vCPU работали, а какие-то нет. Когда разница во времени работы достигала порогового значения (несколько миллисекунд), гипервизор останавливал «убежавшие вперед» vCPU и ждал возможности выделить процессорные такты всем vCPU этой VM одновременно. Получалось, что значительную долю времени все несколько vCPU этой VM простаивали. Так поступал ESX версии 2.

Однако ESXi, начиная еще с версии 3, использует механизм под названием «Relaxed coscheduling». Суть его заключается в том, что одновременно получать такты должны не все vCPU одной VM, а лишь те, что «отстали». Это уменьшает потери производительности из-за виртуализации, позволяя более эффективно утилизировать процессорные ресурсы сервера.

Однако панацеей такой механизм не является, поэтому следует стремиться настраивать для виртуальных машин так мало vCPU, как это возможно с точки зрения производительности.

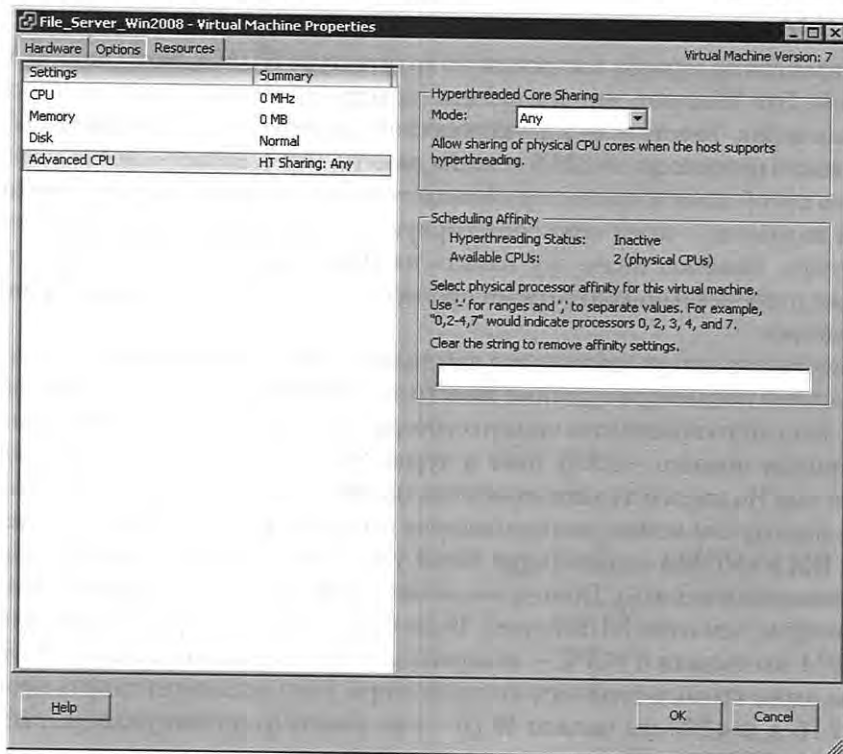
Обратите внимание. Консольная утилита **esxtop** (**resxtop** для vCLI) показывает время ожидания синхронизации в столбце **%CSTP**. Таким образом, эта величина характеризует накладные расходы на виртуализацию процессоров многопроцессорной VM. Если значение этого счетчика превышает пороговое – возможно, VM будет работать быстрее, если ей дать меньше виртуальных процессоров. Или стоит уменьшить число виртуальных процессоров на данном сервере – в смысле выключив или мигрировав на другие сервера другие VM.

Также в свойствах VM на вкладке **Resources** есть строка **Advanced CPU** (рис. 6.24). Здесь вы можете задать настройки **Hyperthreaded Core Sharing** и **CPU Affinity**.

Hyperthreaded Core Sharing управляет тем, как будут использоваться логические процессоры, на которые делится каждое физическое ядро при включении гипертрейдинга. Напомню, что для процессоров с включенным гипертрейдингом каждое физическое ядро порождает два логических, что теоретически позволяет выполнять часть команд в два потока.

Варианты настройки:

- ☐ **Any** – когда виртуальный процессор этой VM работает на каком-то ядре, то на втором логическом ядре этого физического ядра могут работать другие vCPU этой и других виртуальных машин;
- ☐ **Internal** – настройка доступна лишь для многопроцессорных виртуальных машин. Когда VM с несколькими vCPU, то они могут работать на разных логических ядрах одного физического ядра. Для VM с одним процессором такое значение этой настройки эквивалентно значению **None**;
- ☐ **None** – когда vCPU этой VM начинает выполняться на каком-то физическом ядре, то он захватывает его полностью. Второе логическое ядро простаивает. На данном ядре выполняется только этот один vCPU этой одной VM.

Рис. 6.24. Настройки **Advanced CPU**

Важно: эта настройка сбрасывается на значение по умолчанию (Any), когда ВМ в DRS-кластере или при миграции ВМ на другой сервер. Применимость этой настройки невелика – лишь для тонкого тюнинга производительности виртуальных машин на одном сервере. Как правило, ESXi хорошо управляет распределением ресурсов без нашего участия.

Обратите внимание. Нет четких данных по эффективности использования гипертрейдинга для ESXi. Как правило, от включения гипертрейдинга производительность меняется незначительно. Вероятность ухудшения производительности от включения этой функции невелика. Я склоняюсь к рекомендации включать его на современных серверах, однако его выключение – один из шагов диагностического «шаманства» при неочевидных проблемах с производительностью.

Scheduling Affinity – здесь мы можем указать ядра, на которых могут выполняться vCPU этой ВМ. Если по умолчанию гипервизор может расположить процессор ВМ на любом ядре, то с помощью этой настройки мы можем ограничить его выбор.

Если мы изменили данную настройку со значения по умолчанию, то ВМ теряет способность к живой миграции, что значительно снижает применимость данной настройки.

NUMA

Современные сервера построены по архитектуре NUMA, Non Uniform Memory Access. Это означает, что вся память сервера «делится» между процессорами в равных долях. То есть если в двухпроцессорном сервере установлено 64 Гб ОЗУ, то у каждого процессора по 32 Гб «своей» памяти. Что означает «своей»? Это означает, что к этой памяти процессор обращается через свой контроллер памяти. А ко второй половине – через межпроцессорную шину и контроллер памяти второго процессора. Немного иначе эту память называют «локальной» и «удаленной», опять же относительно конкретного процессора. Так вот, к локальной памяти доступ быстрее.

В контексте виртуализации это чем важно – если ВМ выполняется на ядре/ядрах одного процессора и данные памяти этой ВМ расположены в локальной относительно этого процессора памяти сервера – это лучше, чем если это не так.

Хорошая новость – ESXi тоже в курсе, что это лучше, и старается сделать именно так. Но здесь есть пара моментов, на которые стоит обратить внимание.

Во-первых, мы можем немного помочь гипервизору с оптимизацией расположения ВМ в «NUMA-узлах» (один такой узел – это один физический процессор и его локальная память). Помочь мы можем очень просто – не делать ВМ больше размером, чем один NUMA-узел. Если процессоры сервера четырехъядерные, а для ВМ мы выдали 6 vCPU – гипервизор будет вынужден часть vCPU выполнять на ядрах второго физического процессора. Если локальная память процессора – 32 Гб, а для ВМ мы выдали 40 Гб – уже память будет вынужденно распределена между NUMA-узлами.

Однако что делать с теми ВМ, кому действительно требуется большое количество vCPU и/или памяти? Хорошая новость – в пятой версии vSphere появилась поддержка того, что можно назвать «виртуальная NUMA», иногда vNUMA, – гипервизор распределяет такую ВМ поровну между несколькими NUMA-узлами и сообщает гостевой ОС о таком распределении. Таким образом, гостевая ОС имеет возможность понять, что для ее процессоров тоже есть локальная и нелокальная память. И если гостевая ОС/приложение имеет оптимизацию для NUMA, эту оптимизацию она может использовать.

По умолчанию виртуальная NUMA используется для ВМ с 8 и более vCPU. А если вы хотите использовать это и для ВМ меньшего размера, то вам помогут расширенные настройки, см. документацию **vSphere Resource Management** ⇒ **Advanced Attributes** ⇒ **Set Advanced Host Attributes**.

Однако виртуальная NUMA не будет использоваться вообще, если для ВМ включено горячее добавление процессора и памяти (вкладка **Options** ⇒ **Memory/CPU Hotplug**).

Кроме того, избегайте живой миграции таких ВМ на сервера с конфигурацией, отличной от того сервера, где ВМ была включена. После такой миграции vNUMA не будет работать. Если у вас есть кластер DRS с серверами разной конфигурации и большие ВМ (8 и более vCPU), то стоит правилами DRS ограничить работу таких ВМ только на серверах одной конфигурации.

6.2.2. Memory

Вот у нас есть сервер ESXi, для простоты один. У него есть сколько-то оперативной памяти для виртуальных машин, назовем это «Доступная память сервера». На нем работает сколько-то виртуальных машин. Каждой из виртуальных машин выделено сколько-то памяти (назовем ее «Настроенная память ВМ», или ее же в пятой vSphere называют vRAM в контексте лицензирования).

Каждая ВМ какую-то долю от настроенной памяти потребляет («Потребляемая память ВМ»). Что можно рассказать про это?

Несколько общих слов

Доступная память сервера – это все гигабайты памяти сервера минус:

- ❑ память, которую гипервизор тратит на себя. ESXi создает в памяти RAM-диск для своей работы. Виртуальным коммутаторам, iSCSI-инициатору и прочим компонентам также нужны ресурсы для своей работы. Обычно это пренебрежимо небольшое количество ресурсов;
- ❑ накладные расходы. Это память, которую гипервизор тратит для создания процесса виртуальной машины. Overhead, говоря в терминах счетчиков нагрузки. Когда мы создаем виртуальную машину и указываем для нее 1 Гб памяти, гипервизор ей выдает часть или 100% этого гигабайта. И даже в случае стопроцентного выделения еще 70–100 Мб гипервизор тратит на накладные расходы. Притом 70–100 Мб накладных расходов – это для гигабайта настроенной памяти. Если для виртуальной машины настроить 64 Гб памяти, накладные расходы составят примерно 1–1,5 Гб;
- ❑ кластер VMware HA может резервировать сколько-то памяти под свои нужды.

Настроенная память ВМ – это тот объем памяти, который мы указываем в настройках ВМ на вкладке **Hardware**. Именно этот объем видит гостевая ОС. Это максимум, который гостевая ОС может использовать. Именно эта настройка учитывается как «vRAM» с точки зрения лицензирования. Однако гипервизор может выделить для ВМ из реальной оперативки и меньший объем, чем «показал» ей памяти. То, что гостю выделено лишь, например, 400 Мб из показанного гигабайта, изнутри не заметить. По каким причинам гипервизор будет так поступать, поговорим чуть позже.

Потребляемая память ВМ – это сколько реальной оперативной памяти использует виртуальная машина. Или, правильнее сказать, какой объем реальной памяти выделил ей гипервизор. В терминах мониторинга это Consumed.

Memory Overcommitment

При обсуждении работы с памятью разных гипервизоров часто встречается термин «Memory Overcommitment». Притом нередко под ним понимаются разные вещи. Мы же под данным словосочетанием будем понимать ситуацию, когда суммарная «настроенная память» (столбец **Memory Size**) всех включенных ВМ на сервере больше, чем «доступная память сервера» (поле **Capacity**) (рис. 6.25).

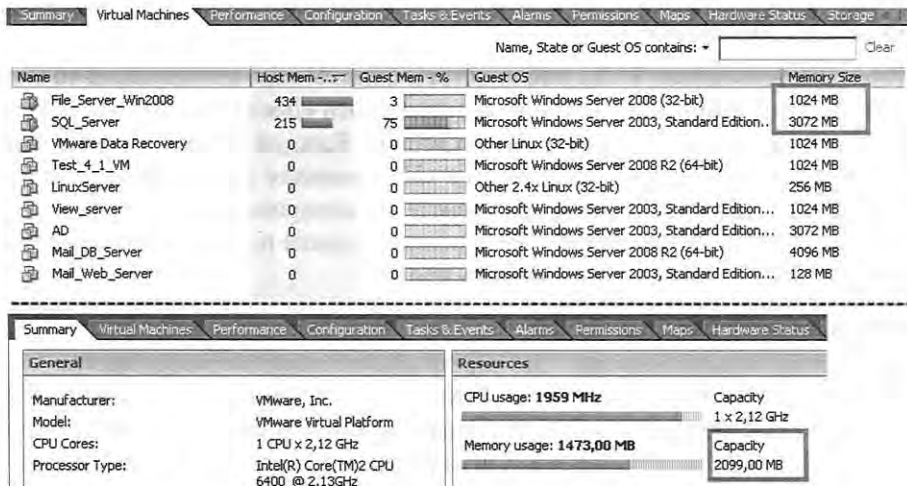


Рис. 6.25. Прикладная иллюстрация Memory Overcommitment

На рис. 6.25 вы можете увидеть МО в действии – на сервере доступны 2099 Мб памяти (внизу), а у запущенных на нем ВМ в сумме 4096 Мб (вверху).

Кстати говоря, и это важно, «потребляемая память» в такой ситуации может быть как меньше (хороший случай), так и больше, чем память «доступная» (плохой случай). Для иллюстрации: обратите внимание на столбец **Host Memory Usage** на предыдущем рисунке. Как видим, при 4096 Мб показанной памяти виртуальные машины реально используют 553 Мб, что вполне помещается в имеющихся у сервера 2099 Мб. Это как раз пример «хорошего» случая memory overcommitment.

Memory Overcommitment достигается на ESXi за счет нескольких технологий. Перечислим их:

- ☐ выделение по запросу;
- ☐ дедупликация оперативной памяти, transparent memory page sharing;
- ☐ механизм баллон, он же balloon driver или vmmemctl;
- ☐ сжатие памяти, memory compression (новинка 4.1);
- ☐ файл подкачки гипервизора, VMkernel swap.

Что это? Каково место этих технологий? Насколько они применимы? Насколько они бесполезны? Давайте поразмышляем.

Вводная к размышлениям.

Мы будем рассуждать о потреблении памяти виртуальными машинами, а точнее гостевыми ОС и приложениями.

Основная идея вот в чем – «показанная» серверу (тут не важно, физическому или виртуальному) оперативная память никогда не загружена на 100%. Почему? У вас загружена именно так? Значит, вы плохо спланировали этот сервер, согласны?

Несколько утверждений, на которых базируется дальнейшее рассуждение.

- ☐ Нам следует стремиться к тому, что виртуальные машины не должны все время потреблять 100% от «показанной» им памяти. Таким образом, про-



чие соображения я буду основывать на том, что у вас именно так. То, что некоторые задачи занимают чем-то (например, кешем) всю свободную память, здесь не учитываем, так как разговор идет в общем.

- ❑ **В разное время суток/дни недели наши приложения потребляют память по-разному.** Большинство задач потребляют максимум ресурсов в рабочие часы в будни, с пиками утром или в середине дня, или <подставьте данные своей статистики>. Однако бывают приложения с нетипичным для нашей инфраструктуры профилем потребления памяти.
- ❑ **В вашей инфраструктуре сделан запас по оперативной памяти серверов, то есть «доступной» памяти.** Этот запас делается из следующих соображений:
 - архитектор опасается промахнуться с необходимым объемом. «Промануться» в смысле «заложить меньше памяти, чем потребуется для оговоренных виртуальных машин»;
 - как запас на случай отказа сервера (или нескольких);
 - как запас на случай роста количества виртуальных машин или нагрузки на существующие виртуальные машины.

Далее.

Рискну предположить, что в наших инфраструктурах можно выделить несколько групп виртуальных машин, по-разному потребляющих память.

Попробую предложить классификацию. Она примерна, потому что тут я ее предлагаю лишь для иллюстрации своих размышлений.

- ❑ **ВМ приложений** – сколько памяти вы выделяете («показываете» в моих определениях тут) своим серверам приложений? При опросах на курсах я слышу цифры 4–8 Гб, редко больше. Вернее, для малого числа ВМ больше. Большинство таких приложений потребляет ресурсы в рабочие часы, однако бывают и исключения (например, сервера резервного копирования, работающие по ночам).
- ❑ **Инфраструктурные ВМ** – всякие DNS, DC и т. п. Обычно гигабайт или два. Потребляют ресурсов мало, пики, если вообще есть, – в рабочие часы.
- ❑ **Тестовые ВМ** – думаю, гигабайт или два в среднем по больнице и больше по требованию, смотря что тестировать будем. Пики в рабочие часы, где-то бывает куча тестовых виртуальных машин, простаивающих подавляюще большую часть времени (как крайний случай – кто-то создал и забросил, а удалить ВМ страшно – вдруг кому нужна).

Давайте теперь рассмотрим эти группы виртуальных машин в контексте механизмов работы с памятью.

Выделение по запросу

Далеко не все ВМ потребляют всю выделенную память все время работы. И часто возникают ситуации, когда для ВМ выделены 10 (например) Гб, а она активно использует лишь 7. Или один.

ESXi умеет определять, сколько памяти ВМ использует, а сколько не использует. И из физической оперативной памяти выделяет каждой ВМ столько, сколько

надо ей. Иными словами, в любой момент времени есть достаточно много *доступной* для машин памяти, но ими *не используемой*.

Виртуальной машине выделили («показали») 2 Гб. Виртуальную машину включили. Что происходит дальше?

Этап 1 – включение. Допустим, в момент включения гость потребляет всю или большую часть доступной памяти. Кто-то забивает ее нулями, кто-то просто много хочет в момент включения. Это самый плохой случай – ведь тогда гостю нужна вся «показанная» ему память. Ок, гипервизор ему всю выдает. Итак, на этапе 1 «потребляемая» память равна «показанной», в самом плохом случае.

Этап 2 – гость стартовал все службы, службы начали работать, создавать нагрузку. Но не 100% памяти потребляется, см. утверждение 1. Например, 1200 Мб из выделенных 2000. То есть гость 800 Мб пометил у себя в таблице памяти как «свободная». По-хорошему гипервизору надо бы ее забрать. Он и забирает (забегая вперед – для этого используется механизм balloon). Итак, баллон раздулся, затем сразу сдулся. Что получилось: гостю 1200 Мб нужно, поэтому они баллоном или не отнялись, или сразу вернулись обратно. Но больше памяти гостю обычно не нужно – и он больше не запрашивает у гипервизора. А раз не запрашивает, гипервизор этой виртуальной машине больше страниц физической памяти и не выделяет.

Итак, если гость потребляет не всю показанную память, гипервизор ему не выделит в реальной оперативке больше, чем он хотя бы раз затребует. То, что однажды было затребовано и выделено, но потом освобождено и не используется, периодически у гостя отбирается. Следовательно, ранее выделенную под это реальную оперативную память гипервизор может использовать под другие ВМ.

Работает этот механизм всегда, когда виртуальная машина потребляет не 100% «показанной» памяти. То есть всегда, кроме первых минут после включения и редких пиков, этот механизм работает и заметно экономит нам память.

Если виртуальная машина не потребляет всю память часто – механизм очень полезен.

Для некоторых тестовых – 100% времени.

Для инфраструктурных – иногда большую часть времени.

Для производственных серверов – как минимум ночью. А если часть серверов нагружается в другое время суток, чем оставшаяся часть, – вообще супер.

Эффект на производительность если и есть негативный, то несущественный.

Transparent Memory Page Sharing

На серверах ESXi работает много виртуальных машин. Часто у многих из них одинаковая операционная система. Иногда на несколько машин установлено одно и то же приложение. Это означает, что для многих ВМ мы вынуждены хранить в оперативной памяти одни и те же данные: ядро операционной системы, драйверы, приложения, dll. Даже внутри одной ВМ есть страницы памяти, занятые одинаковыми данными.

На ESXi работает фоновый процесс для нахождения одинаковых страниц памяти. Одинаковые для нескольких ВМ страницы памяти хранятся в физиче-

ской оперативной памяти в единственном экземпляре (разумеется, доступ к такой странице у виртуальных машин только на чтение).

Гипервизор считает контрольные суммы страниц оперативной памяти. Находит одинаковые (для одной или нескольких виртуальных машин) страницы. И одинаковые страницы из разных «виртуальных таблиц памяти» адресует в одну-единственную страницу в памяти реальной (рис. 6.26).

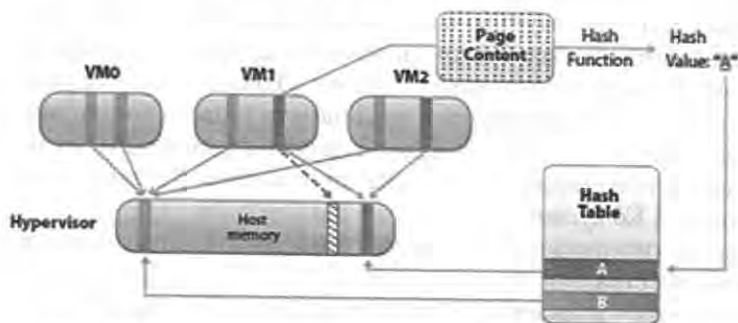


Рис. 6.26. Иллюстрация работы механизма Page Sharing
Источник: VMware

Притом механизм на самом деле двухуровневый – сначала по контрольным суммам находятся кандидаты на идентичность, затем они сравниваются побайтно для гарантии одинаковости.

Получается, на пустом месте гипервизор делает «потребляемую» память меньше, чем она могла бы быть без данного механизма. Ну, про «на пустом месте» я слегка соврал – гипервизор тратит ресурсы процессоров сервера на подсчет контрольных сумм. Но с учетом того, что, как правило, ресурсов процессора у нас в избытке, этот обмен нам очень выгоден.

Насколько часто это применяется к разным группам виртуальных машин?

Сложный вопрос. Сложность – в механизме Large Pages.

Оперативная память используется, адресуется блоками. Их называют «страницы». Классический размер страницы оперативной памяти – 4 Кб. Если мы хотим выделить гостю 4 Гб памяти, то гипервизор для этого гостя выделяет чуть более миллиона страниц.

Гипервизору необходимо обслуживать таблицы трансляции – ведь у гостя эти страницы нумеруются по-своему, а когда процессорные инструкции гостя выполняются на реальных процессорах – гипервизору необходимо «гостевую нумерацию» страниц транслировать в реальную нумерацию. Получается таблица трансляции в миллион записей.

Это задача, актуальная не только для гипервизора, потому что даже когда какая-то обычная ОС работает на физическом сервере – ей необходимо использовать разные таблицы с адресами страниц.

Поэтому сейчас разные ОС, и ESXi в их числе, используют Large Pages, большие страницы, страницы памяти размером 2 Мб. Поэтому для ВМ с 4 Гб

памяти будет выделено не миллион, а порядка двух тысяч страниц. Накладные расходы сократятся. А эффективность дедупликации уменьшится – практика показывает, что найти 4 Кб одинаковых данных в памяти множества ВМ легко, а найти идентичные куски в 2000 Кб уже невозможно (исключая обнуленные страницы).

Таким образом, теория и практика гласят, что эффект от page sharing будет маленьким. Хотя в реальности там довольно сложный алгоритм:

Хотя сам ESXi использует большие страницы, но механизм дедупликации памяти воспринимает их, «как будто» они состоят из маленьких страниц, и считает контрольные суммы этих маленьких страниц. Однако механизм page sharing для маленьких страниц не используется, пока памяти сервера достаточно. А вот если памяти сервера перестает хватать на все виртуальные машины, то, перед тем как включить какой-то из механизмов подкачки, гипервизор начинает делать sharing этих маленьких 4 Кб кусков.

Получается, гипервизор в фоне делает подготовительную работу для дедупликации маленьких страниц, но использует большие, пока памяти в достатке. А если ее стало не хватать – пускает в ход заранее накопленные данные и пытается сэкономить память при помощи этого механизма, пусть и отказом от больших страниц.

Производительность не испытывает негативного эффекта от применения данного механизма. Теория гласит, что это возможно в редких случаях, но мой личный опыт молчит о таких ситуациях.

ESXi знает, что такое архитектура NUMA, и если сервер у нас этой архитектуры, то дедупликация страниц памяти идет внутри каждого одного NUMA узла независимо, чтобы виртуальной машине не приходилось за отдельными, дедуплицированными страницами обращаться в память другого процессора.

Однако в теории накладные расходы имеют место быть: если какая-то ВМ хочет изменить разделяемую с другими страницу памяти, с помощью технологии Copy-on-write делается копия страницы, которая и отдается приватно данной ВМ. Это медленнее, чем просто дать записать в неразделяемую страницу. Насколько заметен эффект в реальной жизни, сказать очень сложно.

Официальные данные по влиянию Page Sharing на производительность следующие (рис. 6.27).

На этом рисунке каждая тройка столбцов – разная задача.

Первый в каждой тройке столбец – производительность задачи в виртуальной машине, когда page sharing для нее не используется.

Второй столбец – производительность при использовании Page Sharing, с настройками по умолчанию.

Третий столбец в каждой тройке – используется Page Sharing, с уменьшенной частотой сканирования памяти.

За единицу выбраны данные тестов с отключенным Page Sharing. Как видно, средний столбец – со включенным Page Sharing и параметрами по умолчанию – не хуже, а иногда лучше по производительности. Улучшение связывают с тем, что memory footprint у виртуальной машины становится меньше и лучше помещается в кеш. В любом случае, разницу по производительности можно назвать несуществ-

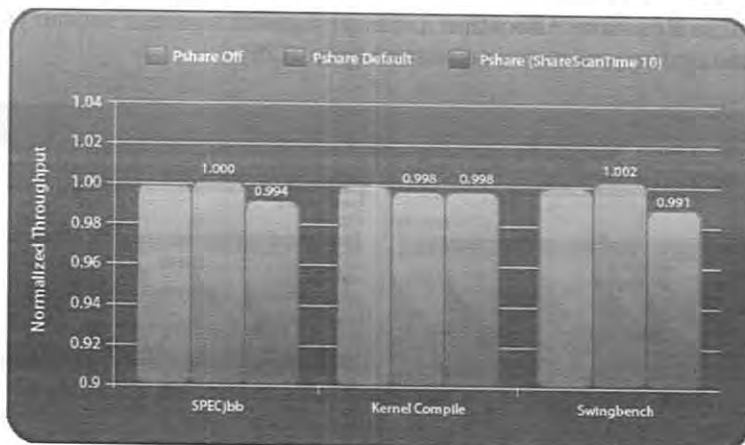


Рис. 6.27. Сравнение производительности трех задач относительно применения к ним Page Sharing

венной. Результаты первой и третьей бенчмарки – число транзакций в секунду, для компилирования ядра – затраченное время.

Эффект от работы данного механизма легко обнаружить на вкладке **Summary** для пула ресурсов (рис. 6.28).

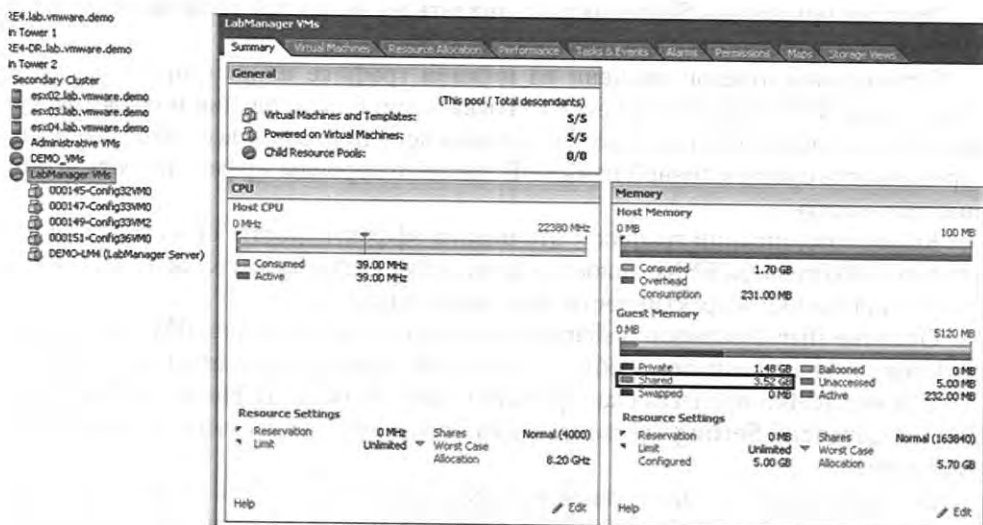


Рис. 6.28. Выделение памяти для пяти VM

В пуле ресурсов с рисунка пять VM. Каждой выделено по 1 Гб памяти. Из 5 Гб выделенной на эти пять VM памяти на 3,5 Гб распространяется механизм Page Sharing.

На вкладке **Resource Allocation** для виртуальной машины можно увидеть эту информацию применительно к ней одной (рис. 6.29).

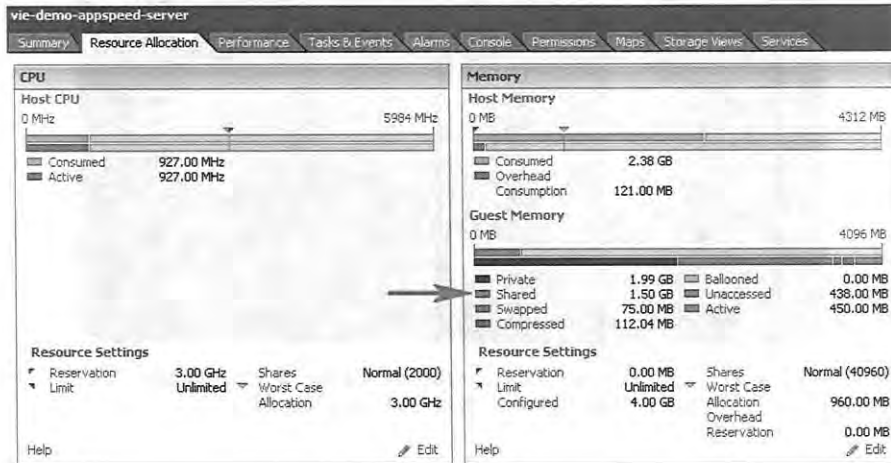


Рис. 6.29. Выделение памяти для одной ВМ

Как видно, из полагающихся ей четырех гигабайт эта виртуальная машина разделяет с другими ВМ полтора гигабайта.

Этот же показатель Shared можно увидеть на вкладке **Performance**, выбрав соответствующий счетчик для памяти (рис. 6.30, вверху).

Отмеченный отрезок времени на нижнем графике иллюстрирует причину того, почему TPS не рекомендуется учитывать при планировании инфраструктуры и почему может быть опасно его активно использовать – все дело в негарантированности сэкономленной памяти. В какие-то периоды эффективность может внезапно падать.

Кроме того, нижний график – это оценка эффективности TPS сразу для нескольких ВМ сервера ESXi. Такое представление информации может быть очень удобно для оценки эффективности этого механизма.

Отличие shared common и shared в следующем: когда у трех ВМ одна общая страница, то shared common – объем этой одной страницы, а shared total – трех.

Для настройки процесса сканирования памяти пройдите **Home** ⇒ **Configuration** ⇒ **Advanced Settings** сервера. Здесь есть возможность указать следующие параметры:

- ☐ Mem.ShareScanTime – как много времени можно тратить на сканирование;
- ☐ Mem.ShareScanGHz – сколько ресурсов можно тратить (если 0, то эта функция не будет работать на данном сервере).

Для отключения этого механизма для отдельной ВМ добавьте в ее файл настроек (*.vmx) параметр

```
sched.mem.pshare.enable = false
```

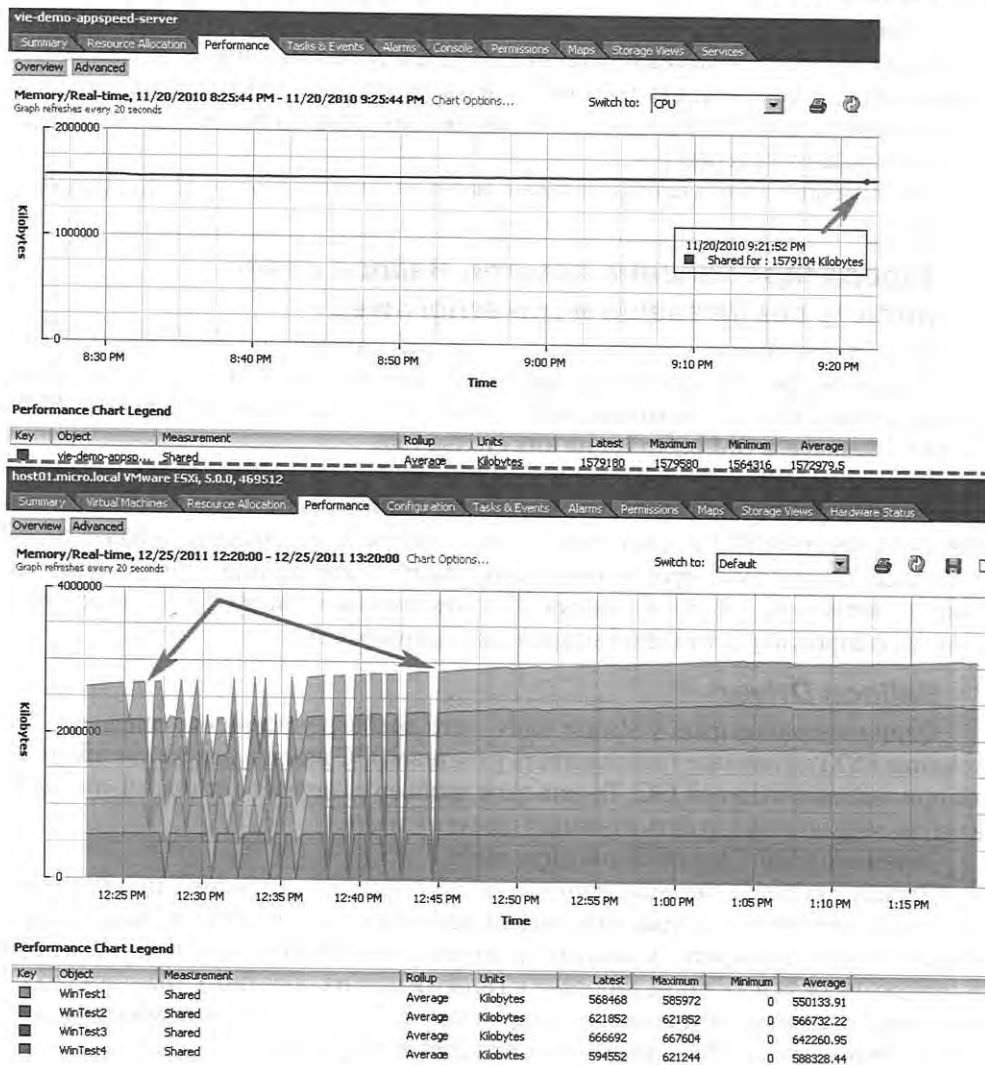



Рис. 6.30. Счетчик Shared

Пробовать отключить этот механизм имеет смысл лишь в случаях аномально высокой загрузки процессоров виртуальной машины или сервера, или для экспериментов по увеличению производительности ВМ.

Однако есть один способ заметно повлиять на page sharing. Этот способ – отключение использования больших страниц. Если мы пройдем в расширенные настройки, **Configuration** ⇒ **Advanced Settings** ⇒ **Mem** ⇒ **Mem.allocGueslargePage**, и заменим единичку на нуль, то после рестарта ВМ или миграции ВМ на этот сервер с других для них уже не будут использоваться большие страницы. По имеющимся у меня отзывам, в течение порядка часа может быть сэкономлено 15–40% памяти.

К сожалению, у меня толком нет данных по негативному влиянию этого отключения. Поэтому рекомендовать его не буду, и если вдруг что – «я вам ничего не говорил» ☺. Впрочем, для тестовых инфраструктур это однозначно архиполезная настройка, позволяющая значительно повысить эффективную плотность виртуальных машин на сервере.

Для справки можете ознакомиться с материалами по ссылке <http://link.vm4.ru/mem>.

Перераспределение памяти. *Balloon driver, memory compression и vmkernel swap*

Когда оперативной памяти много и все работающие виртуальные машины в ней помещаются, все хорошо. Однако не исключены ситуации, когда памяти хватать перестает. Самый явный пример – произошел отказ сервера, серверов ESXi осталось меньше, а ВМ работает на них столько же.

В таких ситуациях нас выручат настройки *limit*, *reservation* и *shares* – с их помощью мы заранее указали, каким ВМ дать больше памяти. Но получается, у менее приоритетных ВМ память потребуется отнять, чтобы отдать ее более приоритетным. Само собой, просто так отнять память у ВМ нельзя – можно часть ее данных вытеснить в файл подкачки. Вот механизмы *vmmemctl* (*balloon driver*), *memory compression* и *vmkernel swap* этим и занимаются.

Balloon Driver

Один из компонентов VMware tools – это драйвер устройства *vmmemctl*. По команде ESXi он начинает запрашивать у гостевой ОС память, так же как это делают приложения гостевой ОС. То есть этот драйвер раздувает тот самый «баллон» внутри, запрашивая (то есть отнимая) память у гостя.

Зачем это надо? Для решения двух задач.

Первая уже описана выше: если гостю были выделены страницы памяти, которые он уже перестал использовать, то гипервизор не может такие свободные страницы отличить и забрать. А раздувшемуся внутри «баллону» гость сам их отдаст в первую очередь и затем не запросит у гипервизора их обратно. Страницы реальной памяти, занятые «баллоном», гипервизор отличит от прочих страниц памяти, выделенных данной ВМ, и перестанет их адресовать для нее.

Посмотрите на рис. 6.31 – страницы памяти, гостем для себя помеченные как «свободные», помечены звездочками слева. А справа, в следующем состоянии описываемого механизма, они уже не занимают железную память сервера.

Вторая задача – когда гостю выделено, например, 2 Гб, а 1 Гб из них надо отнять. Характерный пример – когда памяти сервера перестало хватать резко увеличившемуся числу виртуальных машин. А число их резко увеличилось из-за сбоя одного из серверов и рестарта его ВМ на другом.

Так вот, у виртуальной машины надо отнять память. Гипервизор раздувает баллон, гость начинает использовать свой собственный файл/раздел подкачки. Реальную память, теперь занятую баллоном (с точки зрения гостя), гипервизор отдает другой ВМ.

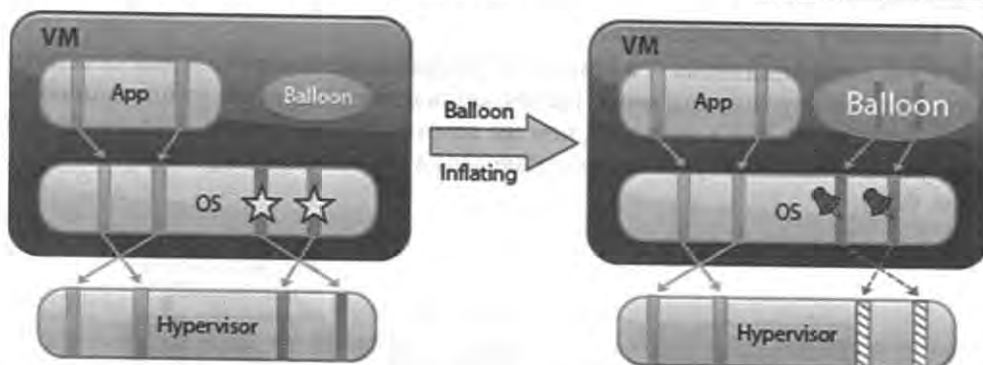


Рис. 6.31. Иллюстрация отъема ранее запрошенной, но неиспользуемой памяти механизмом Balloon

То есть balloon driver – это способ гипервизору задействовать механизм подкачки гостя.

Кстати, в моем примере гипервизор отдаст команду отнять гигабайт. А весь ли гигабайт баллон отнимет? Может быть, и не весь – гость вполне может отказаться ему в памяти, если сочтет, что другим приложениям память нужнее. Если баллон не справился, то гипервизор доотнимет оставшееся с помощью memory compression и vmkernel swap.

Когда механизм баллона может не справиться:

- ☐ когда в VM не установлены VMware tools. vmmemctl – это часть VMware tools;
- ☐ когда vmmemctl не загружен. Например, при старте VM, до того как загрузилась гостевая ОС;
- ☐ когда vmmemctl не способен отнять память достаточно быстро;
- ☐ по умолчанию баллон отнимает не более 65% памяти VM. Кроме того, можно эту настройку поменять индивидуально для VM, прописыванием в ее файле настроек

`sched.mem.maxmemctl` = <разрешенное к отъему количество мегабайт>

Проверить, что для VM недоступен механизм balloon, можно следующим образом: запустите `esxtop`, переключитесь на экран **Memory** нажатием «m», нажмите «f» и отметьте столбец «MCTL». Нажмите **Enter**. Значение «N» в этом столбце для какой-то VM означает, что драйвер vmmemctl в ней не работает.

В ваших интересах, чтобы этот механизм был доступен для всех VM. В частности, потому что он используется не только для отъема и перераспределения памяти в случае нехватки ресурса (как в примере выше). Еще он используется, чтобы вытеснить VM из части физической оперативной памяти, когда та ей уже не используется. Это намного более частая операция, и с использованием balloon она проходит безболезненно и неощутимо для производительности VM.

Memory compression

Суть этого механизма – в том, что гипервизор резервирует некий объем памяти под создание эдакого кеша. Теперь, когда встает задача два из описания баллона – впихнуть ВМ в меньший, чем им хочется, объем памяти, – часть памяти виртуальной машины будет сжата и помещена в ранее зарезервированную область (рис. 6.32).

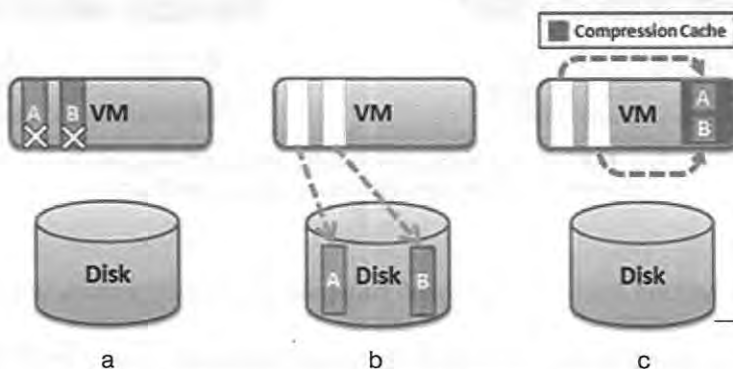


Рис. 6.32. Иллюстрация работы Memory Compression

Состояние «а» – надо выгрузить две страницы памяти из реальной оперативки.

Состояние «b» – решение вопроса «по старинке», механизмами подкачки.

Состояние «с» – современное решение вопроса. Страницы сжаты и помещены в ранее зарезервированную область памяти сервера.

Идея в том, что сжать и записать данные в память, или прочитать и разжать, быстрее, чем без сжатия читать или писать с диска, из файла подкачки.

Обратите внимание на то, что память под этот кеш не резервируется на сервере заранее – это часть памяти самой ВМ. То есть когда для ВМ перестает хватать памяти, гипервизор отнимает у нее еще немного – и в этом «немного» он размещает сжатые страницы ее памяти, которым не хватает места просто так.

Какие-то настройки для данного механизма доступны только через **Advanced Options** для ESXi.

Для того чтобы этот механизм был эффективен, необходимо, чтобы данные в памяти ВМ сжимались вдвое или более. Гипервизор в фоне и заранее пытается обнаружить такую память, и механизм сжатия памяти используется, только если поиски увенчались успехом.

VMkernel swap

Чуть выше я описал `vmxmemctl (balloon)` – механизм для вытеснения части оперативной памяти гостя в файл/раздел подкачки.

Затем рассказал про `memory compression`, промежуточный механизм между использованием механизмов подкачки и просто работой в оперативной памяти.

А теперь поговорим про последний механизм для той же самой цели – отъема реальной памяти у виртуальной машины.

При включении каждой виртуальной машины гипервизор создает файл .vswp – файл подкачки vmkernel для этой ВМ.

Теперь когда гипервизору надо забрать часть памяти у виртуальной машины, он может просто из части страниц скопировать содержимое в данный файл и перенаправлять обращения ВМ к памяти в файл. Гость ничего про это знать не будет.

Чтобы проиллюстрировать отличие между механизмами balloon и vmkernel swap, давайте взглянем на рис. 6.33.

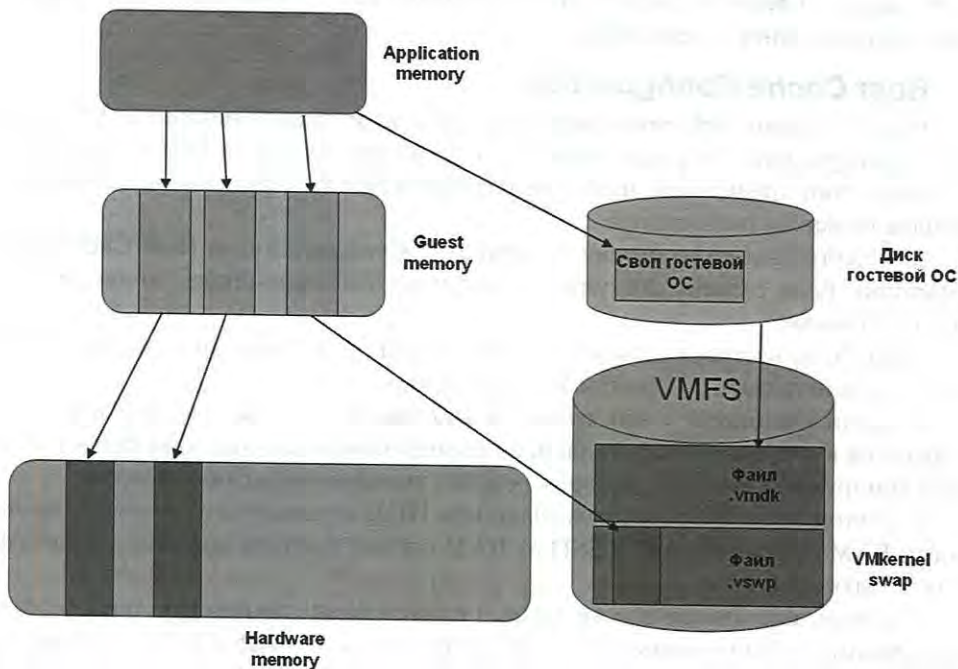


Рис. 6.33. Уровни работы с оперативной памятью

Вы видите три уровня оперативной памяти.

Верхний, Application memory, – это память, которую видят приложения внутри ВМ. Адресоваться эта память может в оперативную память (которую гостевая ОС считает аппаратной) плюс файл подкачки гостевой ОС.

Второй уровень, Guest memory, – это та память, которую показывает гипервизор для гостевой ОС. Адресоваться он может в физическую память сервера и файл подкачки гипервизора. Это тот самый файл, который создается при включении ВМ и размер которого равен hardware memory минус reservation.

Как вы видите, файлов подкачки мы можем задействовать два, на разных уровнях этой схемы. А какой лучше? Лучше тот, который является файлом подкачки гостевой ОС. Потому что если используется он, то гостевая ОС может сама

выбрать страницы памяти, которые пойдут первыми в файл подкачки, и выберет она наименее важные. В то время как гипервизор в свой файл подкачки помещает какие-то страницы памяти ВМ, у гипервизора нет возможности определить, что в них и насколько они критичны.

Так вот, balloon – это способ вытеснить гостя из реальной оперативной памяти в его внутренний файл подкачки, а vmkernel swap – вытеснить гостя из реальной оперативной памяти во внешний файл подкачки, который гипервизор создал для этой виртуальной машины.

VMkernel swap можно задействовать всегда – гипервизору нужно лишь в таблице памяти поменять ссылки с памяти физической на файл подкачки – это основной плюс данного механизма.

Host Cache Configuration

В пятой версии vSphere появилась функция оптимизации механизма VMkernel swap. Его суть в том, что если серверу доступно хранилище на SSD-накопителях, то часть этого хранилища можно использовать под кэширование используемых файлов подкачки гипервизора.

Для настройки этого механизма пройдите **Configuration** ⇒ **Host Cache Configuration**. Если серверу доступно подходящее SSD-хранилище, то вы увидите его/их в списке.

Зайдя в его настройки, вы определяете объем места этого хранилища, которое можно задействовать под файлы VMkernel swap.

Обратите внимание – это именно кэширование. Файлы vswp, как и раньше, создаются в каталоге ВМ (или на явно определенном для них хранилище). А на SSD копируются те из них, которые реально используются гипервизором.

С учетом того, что производительность HDD отличается от производительности RAM на два порядка, а SSD от RAM отстает всего на порядок – выигрыш может быть заметным.

Впрочем, мое мнение все же такое – планирование инфраструктуры должно быть таким, чтобы механизм VMkernel swap не использовался 100% времени. Но если вы все же допускаете, что он может использовать в заметном объеме (или хотите быть к этому готовым), то, может быть, вам будет оправданно добавить к конфигурации ваших серверов 1–2 SSD-накопителя для использования их в описанном качестве.

Насколько часто три описанных механизма применяются к разным группам виртуальных машин?

У механизма balloon задач, можно сказать, две. Первая задача – отнятие незанятой памяти. Эта задача стабильно актуальна для виртуальных машин любой группы.

Вторая задача, общая для всех трех механизмов, – перераспределение используемой памяти между виртуальными машинами. Данная задача мне видится мало актуальной – при адекватном планировании инфраструктуры. Давайте разберем ее слегка подробнее.

Итак, из чего может взаться ситуация, когда у одной ВМ память надо отнять, чтобы отдать другой? Я вижу несколько вариантов.

Когда у нас мемору `overcommitment` и сразу у многих ВМ наступили пики нагрузки, что привело к загрузке сервера на 100% по памяти. Это, кстати говоря, причина пользоваться мемору `overcommitment` аккуратно. Впрочем, бездумно совсем отказываться от мемору `overcommitment`, по моему мнению, тоже не стоит.

Когда у нас ломается один из серверов кластера HA. Например, у нас 10 серверов, все загружены по памяти (днем, в будни) процентов на 70. Однако после отказа одного из серверов есть вероятность, что один или несколько оставшихся окажутся загруженными на 100% – HA хоть и выбирает для включения каждой из упавших виртуальных машин наименее загруженный сервер, но гарантий достаточности ресурсов не дает.

Далее ситуация разветвляется. Если у вас есть DRS, то он быстренько разнесет ВМ по разным серверам, и перераспределять память не придется.

А вот если DRS нет, или он **не** в автоматическом режиме – вот тут мы приходим к нехватке на все ВМ физической памяти (на одном или нескольких серверах). И гипервизор отнимет память у части машин с помощью баллона или оставшихся двух механизмов.

Таким образом, если в вашей инфраструктуре выполняются следующие условия:

- ☐ ресурсов серверов достаточно для всех виртуальных машин, с учетом их пиков и с учетом плановых и неплановых простоев части серверов. Это самое главное условие;
- ☐ у вас есть кластер DRS, и он настроен на автоматический режим балансировки нагрузки.

Так вот, если эти условия выполняются, вероятность того, что перераспределять память не потребуется, стремится к 100%.

Вывод: наличие механизмов перераспределения памяти – это способ значительно сократить негативный эффект от недостатков планирования инфраструктуры.

Хочется явно отметить, что смысл баллона, сжатия памяти и подкачки гипервизора – в том, чтобы запустились все виртуальные машины, пусть они и будут испытывать замедление из-за свопирования того или иного рода. Правда, какие-то виртуальные машины могут «остаться на коне» за счет опускания других ВМ в еще большее «свопирование» (в настройке приоритета виртуальных машин нам помогут настройки `reservation` и `shares`).

И по большому счету, это и есть эффект от работы и сам смысл существования описываемых функций перераспределения памяти. Напомню, что это не относится к выделению памяти по запросу и TPS.

Отсюда вывод: наличие `Balloon driver`, `memory compression` и `vmkernel swap` – это несомненный плюс для инфраструктуры, но не панацея от нехватки памяти. Лучше планировать инфраструктуру, нагрузку на нее и увеличение доступных ресурсов так, чтобы эти механизмы не пригождались вообще, – иначе часть виртуальных машин будет испытывать нехватку ресурсов.

Нехватка памяти на всех – какой механизм будет использован?

У ESXi есть счетчик – процент свободной памяти.

Его пороговые значения. Судя по документации, это:

- ☐ >6% = high. Нормальное состояние сервера;
- ☐ >4% = soft. Памяти осталось меньше, чем считается нормальным, но острой нехватки нет;
- ☐ >2% = hard. Ситуация с памятью становится критичной;
- ☐ >1% = low. Все, приехали.

Однако, по некоторым данным, эти константы слегка другие:

- ☐ High = свободно свыше 64% памяти;
- ☐ Soft = свободно от 64% до 32%;
- ☐ Hard = свободно от 32% до 16%;
- ☐ Low = свободно менее 16%.

Как ESXi реагирует на смену состояния этого счетчика:

- ☐ состояние **High**. Задействуется только page sharing;
- ☐ состояние **Soft**. Свободной памяти меньше, чем хотелось бы. Начинаем использовать **balloon**;
- ☐ состояние **Hard**. Свободной памяти мало. Используем еще и **compression**, и **vmk swap**;
- ☐ состояние **Low**. Гипервизор использует все доступные механизмы и плюс к тому может остановить выполнение виртуальных машин, требующих память сверх выделенной в данный момент.

Обратите внимание. Если для какой-то виртуальной машины следует отключить работу механизмов Balloon driver и VMkernel swap, то простой способ это сделать – указать значение reservation для памяти равным значению Hardware memory этой виртуальной машины. В пятой версии ESXi для этого появился специальный флажок на вкладке **Resources** ⇒ **Memory** в свойствах VM.

Balloon, memory compression и vmkernel swap делают одну и ту же работу. Но balloon делает ее более оптимально – позволяет гостю самому выбрать, что помещать в своп. Данные тестов приведены на следующих рисунках, первый из которых – рис. 6.34.

Столбцами отображается объем отбираемой у виртуальной машины памяти. Красная линия с ромбами – скорость компиляции, зеленая с квадратами – только vmkernel swap. Как видно, когда из 512 Мб памяти у гостя оставалось только 128, при использовании для отъема памяти balloon скорость выполнения бенчмарки практически не снижалась.

Специфика данного теста, компиляции, – в том, что самому процессу память особо не нужна – основной объем занят под кеш данных. Так вот, в случае balloon гость понимал, что в своп лучше положить сначала данные, чем ядро ОС или ядро приложения. А в случае vmkernel swap такой выбор сделать нельзя, и в своп идет «что-то».

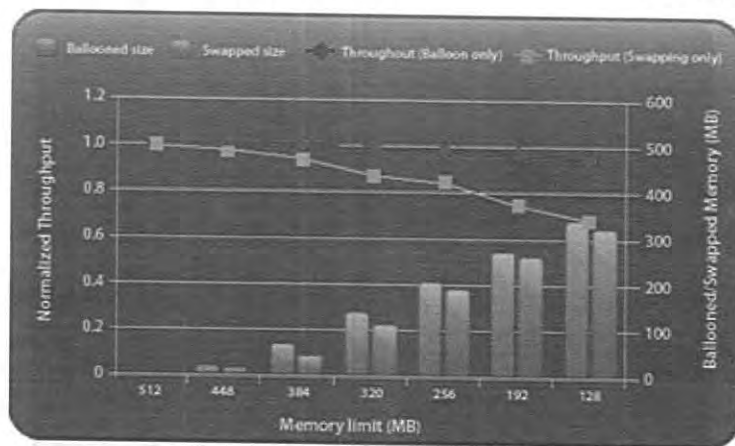


Рис. 6.34. Время выполнения компиляции при отъеме памяти с помощью balloon или vmkernel swap

А вот данные при похожих условиях для базы данных Oracle, нагруженной соответствующей утилитой (рис. 6.35).

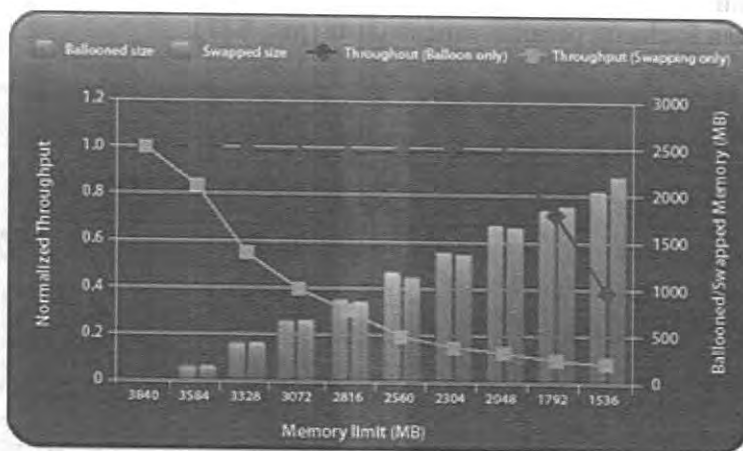


Рис. 6.35. Количество транзакций в секунду для БД Oracle при отъеме памяти с помощью balloon или vmkernel swap

Обратите внимание: пока balloon отнимал меньше, чем 1792 Мб из 3840, производительность практически не падала. Это опять же специфика приложения, но приложения характерного.

А вот для каких-то приложений разницы не будет (рис. 6.36).

И в начальном диапазоне vmkernel swap даже меньше негатива оказывает на производительность ВМ. Впрочем, процент приложений, вот так использующих память, мал в общем случае. Здесь использовалась бенчмарка SPECjbb – утилита

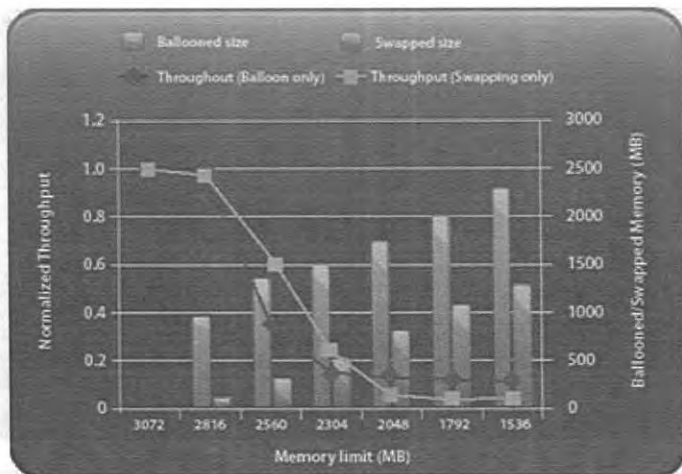


Рис. 6.36. Количество транзакций в секунду для SPECjbb при отъеме памяти с помощью balloon или vmkernel swap

для нагрузочного тестирования, имитирующая нагрузку на серверную часть Java-приложений.

А вот для Exchange разница кардинальна (рис. 6.37).

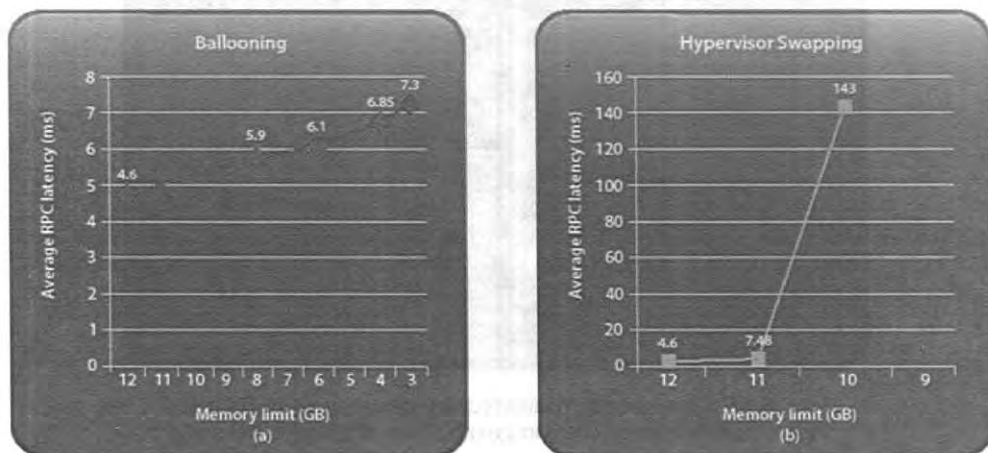


Рис. 6.37. Задержки при обработке почты сервером Exchange при отъеме памяти с помощью balloon или vmkernel swap

Если отнять 9 из 12 Гб памяти баллоном, то это даст удвоение latency, в то время как отнятие 2 Гб из 12 при помощи vmkernel swap – утридцатие (!). Опять же Exchange использует всю доступную память для кеширования данных с целью минимизации обращения к дискам.

Таким образом, если использование подкачки неизбежно, balloon – это меньшее зло.

Однако еще разок замечу: по моему мнению, вам редко придется оказаться в ситуации, когда balloon будет работать из-за нехватки памяти. И даже если такая ситуация случится, balloon даст снижение производительности, просто почти всегда меньшее снижение, чем использование vmkernel swap.

А теперь сравним это еще и с compression.

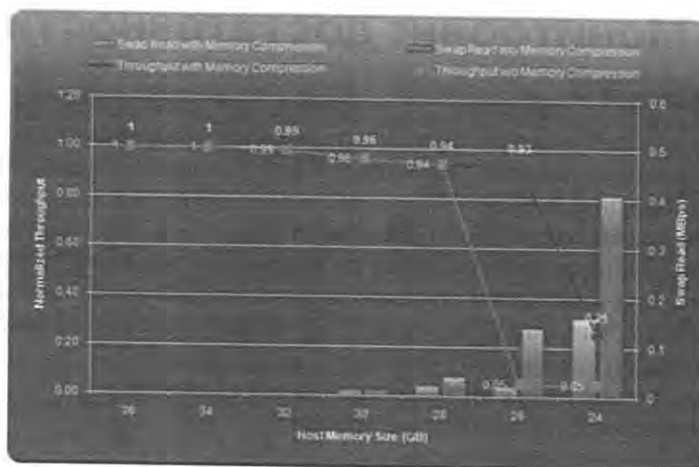


Рис. 6.38. Отъем памяти у сервера SharePoint механизмами memory compression и vmkernel swap

Как видим, при использовании механизма balloon довольно безболезненно можно отобрать 8 из 32 Гб памяти, а с memory compression – еще 2 Гб.

Как видно, compression – не панацея, но если уж памяти жестко не хватает, то с compression лучше, чем без него (лучше, чем только с vmkernel swap сразу после balloon, если быть точным). Нагрузка на процессоры увеличивается на 1–2%, что неощутимо.

Первоисточник данных тестирования – документ «Understanding Memory Resource Management in VMware ESXi 5» (<http://www.vmware.com/resources/techresources/10206>).

6.2.3. Disk

Одним из механизмов управления ресурсами дисковой подсистемы у ESXi можно рассматривать настройки Multipathing. Подробно о них рассказывалось в главе 3.

Еще в версии 4.1 появился механизм Storage IO Control – о нем рассказывалось в разделе 6.1.4.

Также оказывает влияние на производительность механизм Storage DRS, появившийся в пятой версии vSphere. Ему посвящен отдельный раздел.

6.2.4. Net

Из механизмов управления ресурсами сетевой подсистемы у ESXi есть только поддержка группировки контроллеров (NIC Teaming) и ограничения пропускной способности канала (traffic shaping). Подробно о них рассказывалось в главе 2.

Кроме того, в версии 4.1 появился механизм Network IO Control, о нем рассказано в разделе 6.1.5.

6.3. Мониторинг достаточности ресурсов

Для того чтобы правильно настраивать распределение ресурсов, необходимо правильно оценивать достаточность их для виртуальной машины. Как правило, довольно легко понять, что для ВМ не хватает ресурсов, – время отклика от приложения больше, чем хотелось бы. Но какого ресурса не хватает? Как настроить автоматическое оповещение о нехватке ресурсов? Про все это поговорим здесь.

Итак, у вас есть подозрение, что виртуальной машине не хватает какого-то ресурса. Что делать, понятно: обратиться к счетчикам производительности и определить узкие места. Разобьем эту задачу на три подпункта:

1. Где искать значения счетчиков производительности?
2. Какие именно счетчики нас интересуют?
3. Какие значения указывают на узкое место?

Поговорим последовательно.

Сразу упомяну о профильных документах: «Performance Troubleshooting for VMware vSphere 4» (<http://communities.vmware.com/docs/DOC-10352>) и «Performance Best Practices for VMware vSphere 5.0» (<http://www.vmware.com/resources/techresources/10199>).

6.3.1. Источники информации о нагрузке

Важным моментом анализа нагрузки на виртуальную среду, анализа ресурсов виртуальных машин является тот, что далеко не всегда здесь применим опыт такого анализа физических серверов. Вернее, он не применим напрямую. Почему так, я подробнее расскажу в следующих разделах. Вкратце – потому что управлением доступа к ресурсам обладают не гостевые ОС, а гипервизор, и именно гипервизор способен показать истинную картину. Поэтому нас интересуют способы доступа к данным гипервизора.

Их четыре:

- клиент vSphere, в первую очередь вкладка **Performance**;
- утилита esxtop в локальной командной строке, или resxtop в vSphere CLI. Плюс вспомогательные средства для анализа полученной через эти утилиты информации. esxtop – не единственная, но основная утилита анализа нагрузки;
- для Windows ВМ можно получить доступ к некоторым счетчикам гипервизора «изнутри», с помощью Perfmon;

- ☐ сторонние средства. По понятным причинам, сторонние средства здесь рассматриваться не будут.

Вкладка Performance и другие источники информации через клиент vSphere

Вкладка **Performance** доступна для объектов разных типов, в первую очередь серверов и виртуальных машин. На примере виртуальной машины: выделяем интересующую ВМ в клиенте vSphere ⇒ вкладка **Performance** ⇒ кнопка **Overview**. Здесь мы видим графики с самыми важными счетчиками для этой ВМ. В выпадающем меню **Time Range** можем выбирать интересующий нас период времени. А в выпадающем меню **View** – переключиться на информацию по утилизации места этой ВМ на хранилищах.

Если на вкладке **Performance** нажать кнопку **Advanced** ⇒ ссылка **Chart Options**, то можно будет выбрать для просмотра произвольный набор счетчиков. Откроется окно настроек (рис. 6.39). Выбираем CPU и нужный отрезок времени в левой части.

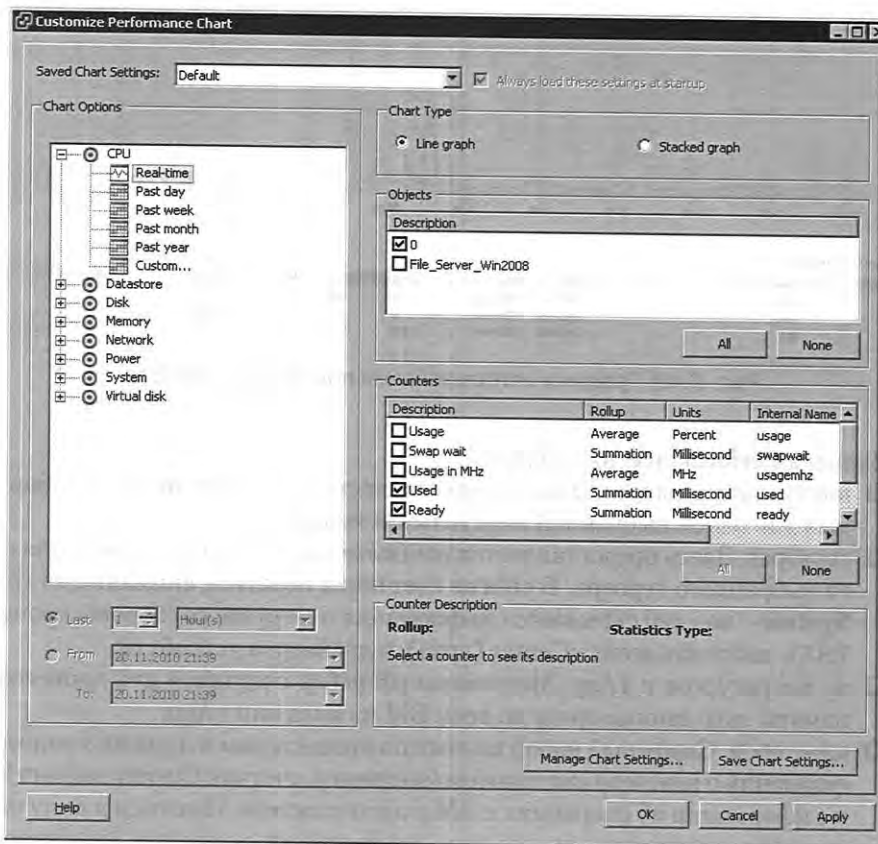


Рис. 6.39. Выбор счетчиков загрузки процессора

В правой части наверху выбираем объект, чьи счетчики хотим смотреть. Здесь «File_Server_Win2008» – это вся виртуальная машина (имеется в виду – сразу все процессоры этой ВМ), а «0» – это первый (и в данном случае единственный) ее процессор. В левой части внизу выбираем конкретный счетчик.

Обратите внимание. Некоторые счетчики доступны только для всей ВМ целиком, а некоторые – только для отдельных ее виртуальных процессоров. Это относится и к некоторым счетчикам для других подсистем.

Нажимаем **ОК** и видим графики (рис. 6.40).

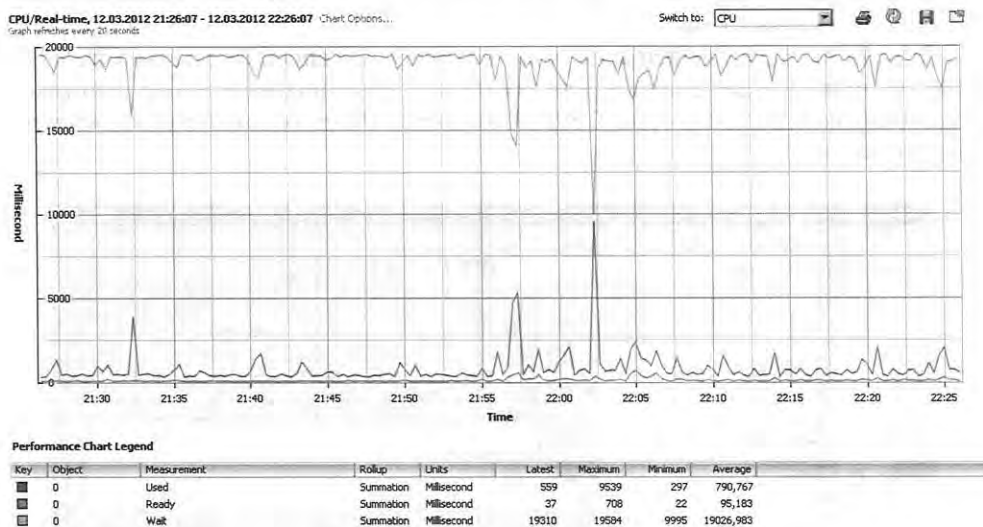


Рис. 6.40. Графики использования процессора для ВМ

Вкладка **Performance** доступна для:

- ☐ виртуальных машин. Здесь предоставляется наиболее полный набор данных для одной выбранной виртуальной машины;
- ☐ серверов. Здесь предоставляется наиболее полный набор данных для одного выбранного сервера. В списке счетчиков обратите внимание на группу **System** – под ней скрывается информация о загрузке со стороны процессов ESXi, таких как агент vCenter (vpxa), и драйверов устройств;
- ☐ пулов ресурсов и vApp. Минимальный набор счетчиков для процессора и памяти, зато данные сразу по всем ВМ из пула или vApp;
- ☐ кластеров. С скромный набор данных по процессорам и памяти, немного информации о кластере как таковом (например, счетчик Current failover level) и информации об операциях с ВМ в этом кластере. Имеются в виду операции вида включения, выключения, перезагрузки, миграций, клонирования и многие другие. А выводимая о них информация – их количество;

- ❑ **datacenter.** Только информация о количестве разнообразных операций с виртуальными машинами;
- ❑ **хранилищ (Home ⇒ Inventory ⇒ Datastores).** Доступна информация об утилизации места виртуальными машинами с разбивкой по их дискам, снимкам состояния (snapshot), файлам подкачки.

Кроме вкладки **Performance**, есть еще некоторые источники информации об использовании ресурсов серверов.

Для пулов ресурсов и серверов мы можем оценить процент использования их ресурсов процессора и памяти. Для хранилищ и серверов мы можем оценить количество свободного места на хранилищах.

Пулы ресурсов

Для оценки потребления ресурсов пула нам пригодятся его вкладки **Summary** и **Virtual machines**. Давайте взглянем на них.

Вкладка **Summary**, рис. 6.41.

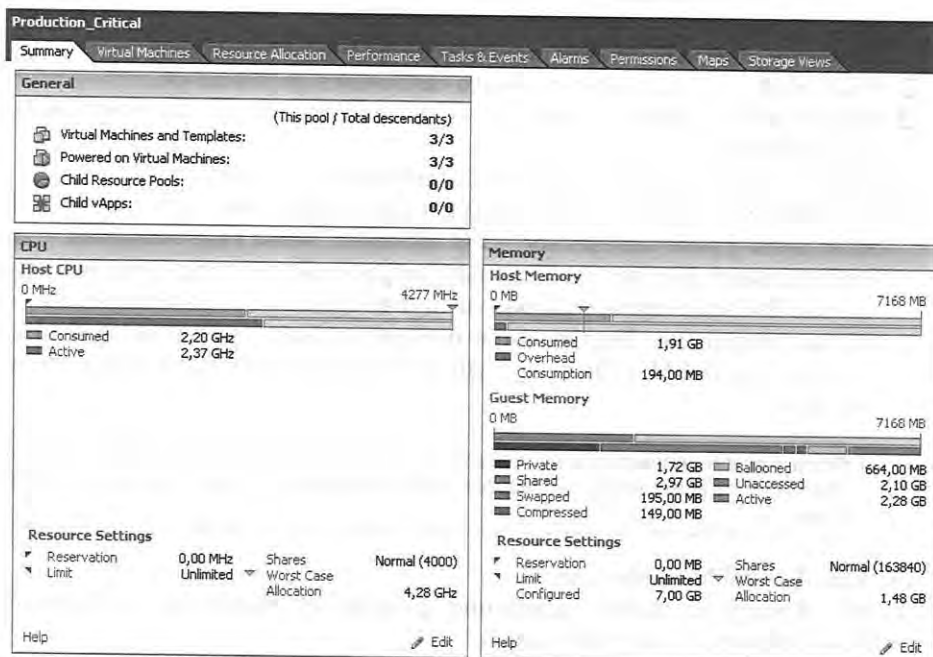


Рис. 6.41. Summary для пула ресурсов

Для процессора здесь отображаются показатели:

- ❑ **Consumed** – текущее потребление ресурсов процессора ВМ этого пула;
- ❑ **Active** – максимальное количество ресурсов, которое может быть выделено для ВМ в этом пуле. Если для пула настроен limit для процессора, то Active не будет больше limit;

- ❑ **Resource Settings** – здесь указаны настройки *limit*, *reservation* и *shares* для этого пула. Самое интересное – это значение **Worst case allocation** – приблизительный подсчет того, сколько ресурсов будут потреблять все включенные ВМ этого пула, если они начнут потреблять по максимуму из того, что им разрешено. Учитываются настройки *limit*, *reservation* и *shares* на уровне каждой ВМ, а также доступные физические ресурсы сервера и пула ресурсов.

Для памяти:

- ❑ **Private** – столько мегабайт выделено для ВМ из физической оперативной памяти, и эти страницы памяти не общие;
- ❑ **Shared** – столько мегабайт памяти выделено для ВМ из физической оперативной памяти, но эти страницы одинаковые хотя бы для двух ВМ, и одинаковые страницы для разных ВМ адресуются в одну страницу в физической памяти. Это экономия памяти механизмом *Transparent Memory Page sharing*. Обратите внимание: если у трех ВМ одинаково по 10 Мб в памяти, то *shared* для них будет равно 30 Мб, хотя этими совпадающими данными реально в памяти сервера будет занято 10 Мб;
- ❑ **Swapped** – столько памяти переадресуется в *VMkernel swap*;
- ❑ **Ballooned** – столько памяти занято баллоном в гостевых ОС;
- ❑ **Unaccessed** – столько памяти сервера не выделено ни для одной ВМ, то есть свободно;
- ❑ **Active** – столько памяти активно задействуется гостевыми ОС;
- ❑ **Resource Settings** – здесь указаны настройки *limit*, *reservation* и *shares* для этого пула. Самое интересное – это значение **Worst case allocation** – приблизительный подсчет того, сколько ресурсов будут потреблять все включенные ВМ этого пула, если они начнут потреблять по максимуму из того, что им разрешено. Учитываются настройки *limit*, *reservation* и *shares* на уровне каждой ВМ, а также доступные физические ресурсы сервера и пула ресурсов.

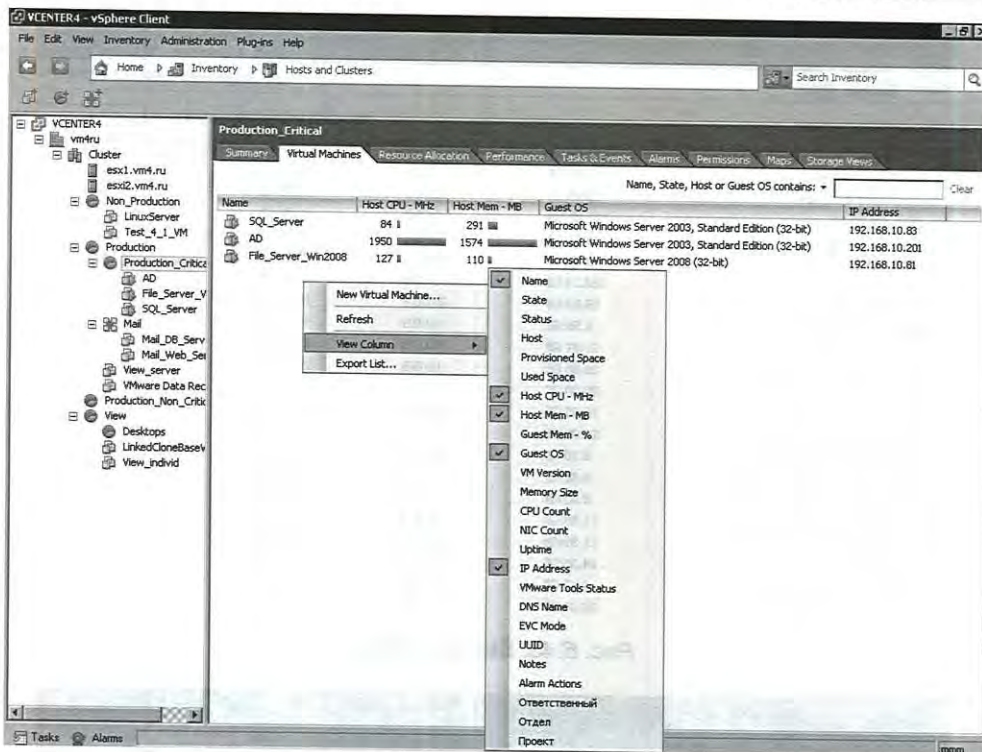
Обратите внимание: если пул ресурсов создан в DRS-кластере, то он свою долю отсчитывает от всех ресурсов кластера, от суммы мегагерц и мегабайт всех серверов в нем.

Вкладка **Virtual Machines**, рис. 6.42.

Здесь нам доступна разнообразная информация – особо обратите внимание на пункт **View Column** контекстного меню.

К ресурсам непосредственно относятся столбцы *Host CPU*, *Host Mem* и *Guest Mem*:

- ❑ **Host CPU** – сколько мегагерц гипервизор выделяет ВМ сейчас;
- ❑ **Host Mem** – сколько мегабайт выделяется ВМ сейчас плюс накладные расходы памяти на нее. Величину накладных расходов можно посмотреть на вкладке **Summary** для ВМ ⇒ **Memory Overhead**;
- ❑ **Guest Mem** – сколько процентов от выделенной памяти активно использует ВМ. Входит в предыдущий пункт.

Рис. 6.42. Вкладка **Virtual Machines**

Хранилища, **Storage Views**

Для любого объекта в иерархии vCenter доступна вкладка **Storage Views** (рис. 6.43).

Здесь вам доступна самая разнообразная информация – обратите внимание на выпадающее меню (рис. 6.44).

То есть вы можете просматривать разного рода информацию для тех или иных объектов. Плюс к тому обратите внимание на пункт **View Columns** контекстного меню здесь (рис. 6.45).

Наконец, обратите внимание на кнопку **Maps** вкладки **Storage Views** (рис. 6.46).

Здесь можно просмотреть взаимосвязи между относящимися к хранилищам объектами.

esxstop и resxstop

Еще один путь для получения данных о нагрузке на сервер со стороны виртуальных машин – воспользоваться командной строкой; нам доступны два варианта одной и той же утилиты – **esxstop** и **resxstop**.

PRI-HA-DRS-Cluster

DRSResource AllocationPerformanceTasks & EventsAlarmsPermissionsMapsProfile ComplianceStorage ViewsServicesvShield

View:ReportsMaps

Last Update Time: 11/20/2010 9:17:40 PM

Show all Virtual Machines

VM or Multipathing Status contains:

VM	Space Used	Snapshot Space	Provisioned Space
VMware Data Recovery	17.00 GB	0.00 B	17.00 GB
vie-demo-win7P-template	10.00 GB	0.00 B	10.75 GB
vie-demo-win7P-03	773.09 MB	0.00 B	10.75 GB
vie-demo-win7-template	8.38 GB	0.00 B	16.00 GB
vie-demo-win7-individual-1	13.12 GB	43.00 B	21.00 GB
vie-demo-win7-02	183.24 KB	0.00 B	25.00 GB
vie-demo-win7-01	15.00 GB	20.00 B	16.00 GB
vie-demo-win2k8-R2-template	6.58 GB	0.00 B	44.00 GB
vie-demo-w7-template	11.81 GB	0.00 B	21.00 GB
vie-demo-w2k8-r2-TechExp-04	20.00 GB	0.00 B	21.00 GB
vie-demo-w2k8-r2-TechExp-03	20.00 GB	0.00 B	21.00 GB
vie-demo-w2k8-r2-TechExp-02	20.00 GB	0.00 B	21.00 GB
vie-demo-w2k8-r2-TechExp-01	20.00 GB	0.00 B	21.00 GB
vie-demo-w2k8 r2 datacenter hot add	6.70 GB	43.00 B	44.00 GB
vie-demo-w2k3-template	6.02 GB	0.00 B	9.00 GB
vie-demo-vShield-Manager	8.00 GB	0.00 B	8.00 GB
vie-demo-view-win7-desktop-template	11.88 GB	0.00 B	21.00 GB
vie-demo-view-win7-desktop	11.89 GB	74.27 KB	41.00 GB
vie-demo-viewmgr	14.33 GB	0.00 B	44.00 GB
vie-demo-view-goldenmaster-win7	9.49 GB	60.94 KB	31.00 GB
vie-demo-vCloud-Director-Cell-2	23.01 GB	7.01 GB	40.01 GB

Рис. 6.43. Storage Views

PRI-HA-DRS-Cluster

DRSResource AllocationPerformanceTasks & EventsAlarmsPermissionsMapsProfile ComplianceStorage ViewsServicesvShield

View: ReportsMaps

Last Update Time: 11/20/2010 9:17:40 PM

Show all Virtual Machines

VM or Multipathing Status contains:

	Space Used	Snapshot Space	Provisioned Space
☒ Show all Virtual Machines			
Show all Datastores	7.00 GB	0.00 B	17.00 GB
Show all Hosts	0.00 GB	0.00 B	10.75 GB
Show all Resource Pools	1.09 MB	0.00 B	10.75 GB
Show all SCSI Volumes (LUNs)	8.38 GB	0.00 B	16.00 GB
Show all SCSI Paths	13.12 GB	43.00 B	21.00 GB
Show all SCSI Adapters	183.24 KB	0.00 B	25.00 GB
Show all SCSI Targets (Array Ports)	15.00 GB	20.00 B	16.00 GB
Show all NAS Mounts	6.58 GB	0.00 B	44.00 GB
	11.81 GB	0.00 B	21.00 GB

Рис. 6.44. Доступная информация на вкладке Storage Views

esxtop мы можем запустить из командной строки ESXi, локально или через SSH.

resxtop мы можем использовать из vSphere CLI под Linux, загрузив и установив их. Или загрузив и запустив на vSphere виртуальную машину с vMA, где эти vSphere CLI уже предустановлены. На примере последнего варианта – подключаемся к vMA по SSH и выполняем команду `resxtop --server <имя сервера ESXi>`.

Независимо от того, работаем ли мы с esxtop или resxtop, интерфейс и возможности практически идентичны.

После запуска утилиты мы увидим примерно следующее (рис. 6.47).

PRI-HA-DRS-Cluster

DRS Resource Allocation Performance Tasks & Events Alarms Permissions Maps Profile Compliance Storage Views Services vShield

View: Reports Maps Last Update Time: 11/20/2010 9:17:40 PM

Show all Virtual Machines VM or Multipathing Status contains:

VM	Space Used	Snapshot Space	Provisioned Space
VMware Data Recovery	17.00 GB	0.00 B	17.00 GB
vie-demo-winXP-template	10.00 GB	0.00 B	10.75 GB
vie-demo-WinXP-03	773.09 MB	0.00 B	10.75 GB
vie-demo-win7-template	8.38 GB		
vie-demo-win7-individual-1	13.12 GB		
vie-demo-win7-02	183.24 KB		
vie-demo-win7-01	15.00 GB		
vie-demo-win2K8-R2-template	6.58 GB		
vie-demo-w2K8-r2-TechExp-04	11.81 GB		
vie-demo-w2K8-r2-TechExp-03	20.00 GB		
vie-demo-w2K8-r2-TechExp-02	20.00 GB		
vie-demo-w2K8-r2-TechExp-01	20.00 GB		
vie-demo-w2K8 r2 datacenter hot add	6.70 GB		
vie-demo-w2K3-template	6.02 GB		
vie-demo-vShield-Manager	8.00 GB		
vie-demo-view-win7-desktop-template	11.88 GB		
vie-demo-view-win7-desktop	11.89 GB		
vie-demo-viewmgr	14.33 GB	0.00 B	44.00 GB

Refresh View Column Export List...

VM Multipathing Status contains:

- ☒ VM
- ☐ Host
- ☐ Resource Pool
- ☐ Multipathing Status
- ☒ Space Used
- ☒ Snapshot Space
- ☐ Virtual Disk Space
- ☐ Swap Space
- ☐ Other VM Space
- ☐ Shared Space
- ☒ Provisioned Space
- ☐ Uncommitted Space
- ☐ Disks
- ☐ Datastores

Рис. 6.45. Доступная информация на вкладке **Storage Views**

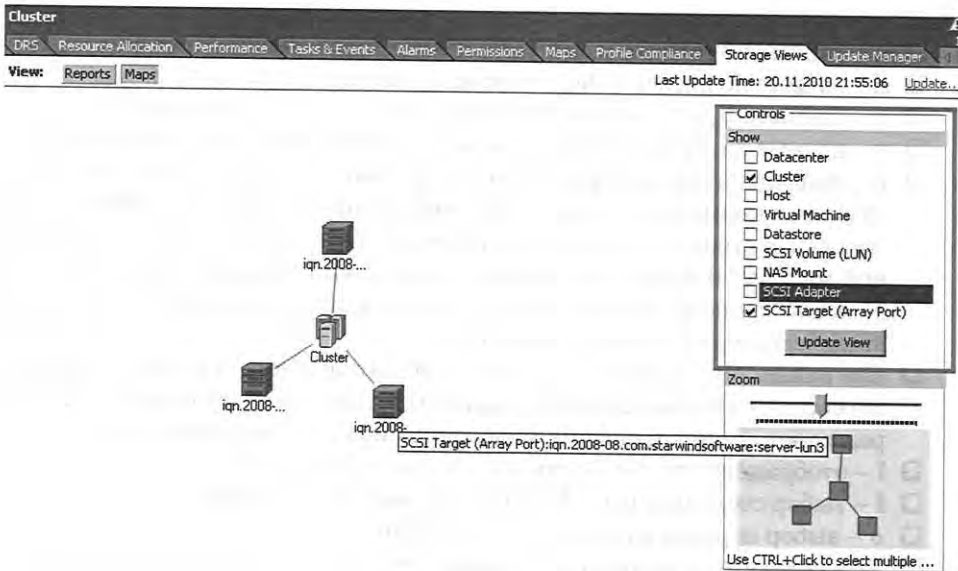


Рис. 6.46. **Maps** на **Storage Views**

Какие именно значения нас интересуют, я сообщу чуть позднее. Пока же немного остановлюсь на интерфейсе и возможностях.

Для того чтобы оставить на экране данные только по виртуальным машинам, нажмите **Shift+v**.

```

3:40:01pm up 22:16, 126 worlds; CPU load average: 0.16, 0.17, 0.17
PCPU USED(%): 22 AVG: 22
PCPU UTIL(%): 24 AVG: 24
CCPU(%): 1 us, 8 sy, 90 id, 0 wa ; cs/sec: 323

```

ID	GID	NAME	NWLD	%USED	%RUN	%SYS	%WAIT	%RDY
1	1	idle	1	76.99	81.52	0.00	0.00	18.38
11	11	consoles	1	10.41	10.00	0.00	0.00	2.00
70	70	File Server Win	4	4.69	4.54	0.32	392.60	2.73
73	73	SQL Server	4	3.70	3.95	0.07	393.55	2.32
7	7	helper	75	0.67	0.69	0.00	7490.68	2.62
47	47	storageRM.4230	1	0.52	0.57	0.01	99.40	0.03
51	51	net-lbt.4234	1	0.46	0.46	0.00	99.26	0.22
54	54	vmkiscsid.4236	2	0.03	0.03	0.00	199.81	0.09
8	8	drivers	10	0.01	0.01	0.00	999.60	0.06
2	2	system	7	0.01	0.01	0.00	699.14	0.12
19	19	vmkapiomod	6	0.00	0.00	0.00	599.80	0.00
9	9	vmotion	4	0.00	0.00	0.00	399.86	0.00
44	44	FT	1	0.00	0.00	0.00	99.98	0.00
45	45	dhclient-uw.422	1	0.00	0.00	0.00	99.99	0.00
46	46	vobd.4224	6	0.00	0.00	0.00	599.99	0.00
50	50	net-cdp.4233	1	0.00	0.00	0.00	99.99	0.00
55	55	vmware-vmkauthd	1	0.00	0.00	0.00	99.98	0.00

Рис. 6.47. esxtop

По умолчанию нам демонстрируется информация о процессоре. Чтобы переключиться на другие подсистемы, нажмите:

- ☐ **m** – данные по памяти;
- ☐ **n** – данные по сети;
- ☐ **d** – данные по дисковой подсистеме, а именно контроллерам;
- ☐ **u** – данные по дисковой подсистеме, а именно устройствам (LUN);
- ☐ **v** – данные по дисковой подсистеме, данные по виртуальным машинам;
- ☐ **c** – данные по процессорам, именно они демонстрируются по умолчанию. Обратите внимание, что на экране информации о процессорной подсистеме каждая строка – это группа процессов, относящихся к одной виртуальной машине. В частности, каждый виртуальный процессор порождает отдельный процесс. Чтобы увидеть данные каждого процесса, нажмите **e** и введите номер группы (столбец ID);
- ☐ или нажимайте цифры **2** и **8** – это позволит подсветить строку, перемещая курсор вверх и вниз. Цифра **4** удалит выделенную строку с экрана. Цифра **6** развернет группу процессов выделенной в данный момент строки;
- ☐ **l** – отображение только процесса с указываемым GID;
- ☐ **f** – выбор счетчиков (столбцов) по текущей подсистеме;
- ☐ **o** – выбор порядка расположения столбцов;
- ☐ **W** – сохранить сделанные изменения в конфигурационный файл esxtop;
- ☐ **#** – количество выводимых строк;
- ☐ **s** и цифра – позволят поменять частоту обновления данных на экране. По умолчанию – раз в 5 секунд;
- ☐ **?** – помощь.

Разумеется, это не все ключи. Список источников дополнительной информации см. чуть ниже.

Обратите внимание. При помощи esxtop возможно принудительно завершать процессы виртуальных машин. Для этого на экране данных процессора (доступен по нажатию **c**) следует нажать **f**, добавить в отображаемое поле **LWID (Leader World Id)** ⇒ **Enter**. Затем нажать **k (kill)** и указать значение **LWID** для той ВМ, которую необходимо принудительно остановить.

В данных esxtop достаточно много нюансов. Что означает (например) %USED в показаниях esxtop? %USED = («время CPU used в предпоследний момент снятия данных» минус «время CPU used в последний момент снятия данных») делить на (время между последним и предпоследним моментом снятия данных).

Рассмотрение всех здесь мне кажется неоправданным, в следующем разделе мы разберем лишь самые важные. Рекомендую ознакомиться с документацией:

- ☐ встроенной справкой, доступной по нажатию **h** в окне esxtop;
- ☐ <http://communities.vmware.com/docs/DOC-7390>;
- ☐ <http://communities.vmware.com/docs/DOC-3930>;
- ☐ <http://communities.vmware.com/docs/DOC-11812>.

Анализ информации от (r)esxtop

В описанном варианте запуска (r)esxtop показывает данные в реальном времени. А если необходимы данные для анализа? Тогда запустите команду следующего вида:

```
esxtop -a -b -d 10 -n 1080 > /tmp/esxtopout_esxi2.csv
```

Параметры задают следующие настройки:

- ☐ **a** – выгрузка всех параметров (можно выгружать лишь часть, для снижения размера итогового файла);
- ☐ **b** – пакетный режим;
- ☐ **d** – размер задержки в секундах, то есть данные снимаются каждые **d** секунд;
- ☐ **n** – количество итераций.

Таким образом, эта команда выгрузит в файл с указанным именем все данные по нагрузке на сервер за **d × n** секунд, начиная с момента запуска команды.

Полученный на выходе csv файл можно загрузить в perfmon или утилиту под названием esxplot для удобного анализа собранных данных. Для копирования этого файла на свой компьютер удобно использовать утилиту WinSCP.

Запустите perfmon: **Пуск** ⇒ **Выполнить** ⇒ **perfmon**.

Дальнейшая инструкция для системного монитора Windows Server 2003 и Windows XP.

В открывшемся окне вас интересует иконка **Загрузка данных из журнала** (View Log Data), затем вкладка **Источник** (Source), на ней выберите **Файлы журнала** (Log files) ⇒ кнопка **Добавить** (Add) ⇒ выберите csv-файл.

Затем вкладка **Данные** (Data) ⇒ удалите все вхождения ⇒ кнопка **Добавить** (Add) ⇒ выпадающее меню **Объект** (Performance Object) ⇒ выберите интересующую группу счетчиков ⇒ нижний левый список, выберите интересующий счет-

чик ⇒ нижний правый список, выберите интересующий объект или объекты ⇒ нажмите **Добавить** (Add). После добавления всех интересующих счетчиков нажмите **Заккрыть**. На рис. 6.48 показано окно **Perfmon** с единственным открытым счетчиком CPU ready для виртуальной машины File_Server_Win2008.

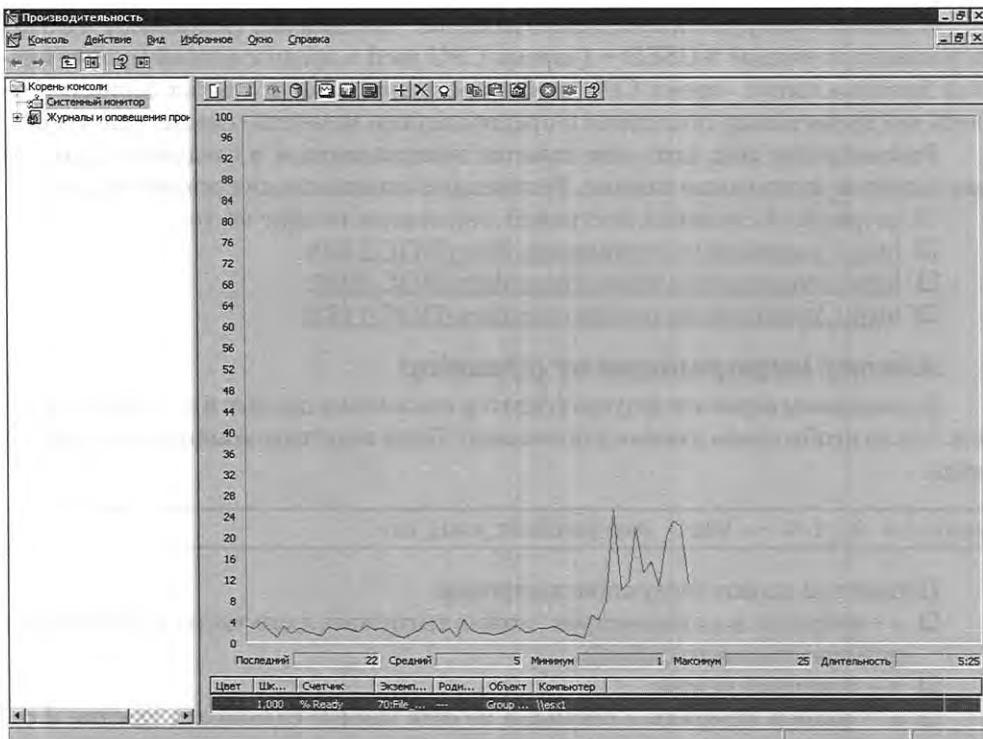


Рис. 6.48. Данные esxtop в **Perfmon**

Альтернатива perfmon – утилита esxplot, собственная разработка одного из инженеров VMware. Ссылку на нее можно найти на <http://labs.vmware.com/>. После загрузки запустите исполняемый файл, меню **File** ⇒ **Import** ⇒ **Dataset** ⇒ укажите csv-файл с данными от esxtop. Затем в нижнем левом поле ищите интересующую группу счетчиков. Справа отобразится график (рис. 6.49).

Я рекомендую попробовать оба средства и выбрать более удобное лично для вас. Мне удобнее esxplot.

Perfmon «внутри» гостевой ОС

В состав VMware tools для Windows входят dll, которые позволяют к некоторым из счетчиков гипервизора получать доступ изнутри виртуальной машины. Запустите perfmon, запустите окно добавления счетчиков. В списке будут группы VM Processor и VM Memory (рис. 6.50).

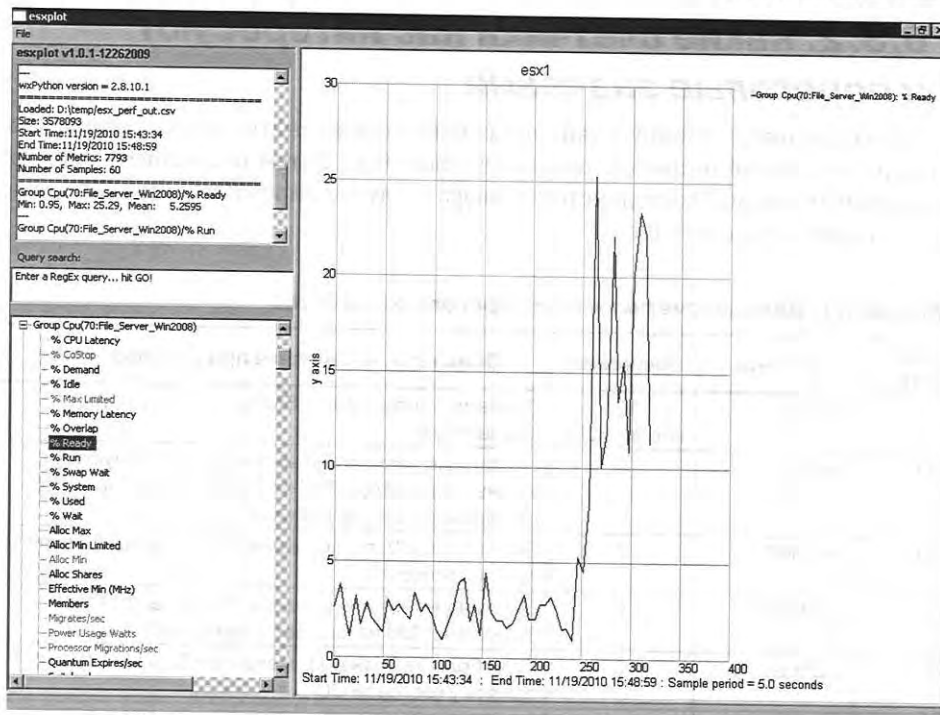


Рис. 6.49. Данные esxtop в esxplot

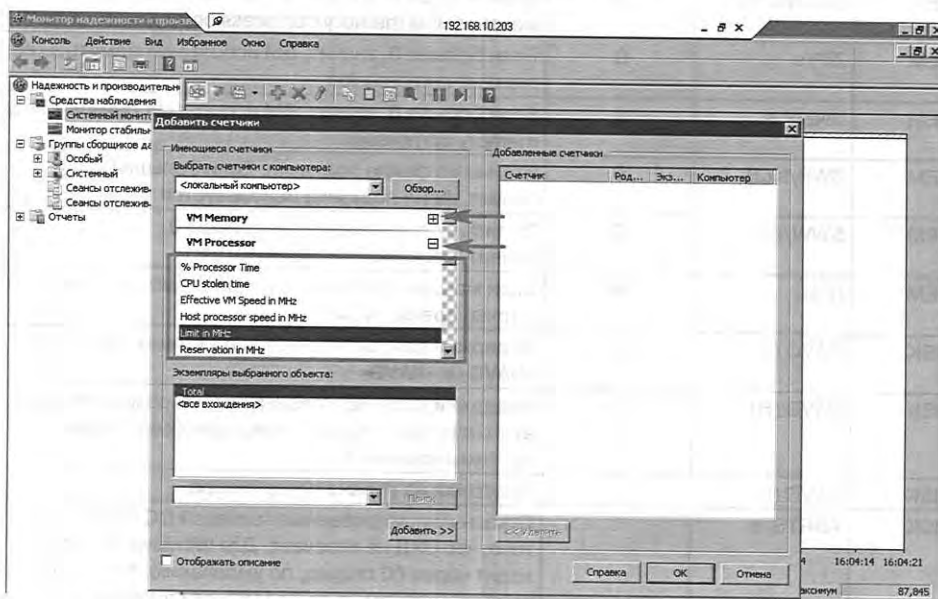


Рис. 6.50. Добавление счетчиков гипервизора в perfmon гостевой ОС

6.3.2. Какие счетчики нас интересуют и пороговые значения

Сначала приведу сводную таблицу самых важных счетчиков, пороговых значений их показаний и краткое описание (табл. 6.1). Затем последовательно коснусь нюансов каждой из подсистем сервера в случае виртуализации и более подробного описания счетчиков.

Таблица 6.1. Важные счетчики и их пороговые значения

Под-система	Счетчик	Значение	Описание, возможная причина проблемы
CPU	%RDY	10* кол-во vCPU	Очередь к процессору. VM не хватает ресурсов процессора
CPU	%CSTP	3	Накладные расходы на многопроцессорную VM. Уменьшите число vCPU этой VM, или число vCPU на сервере, где она работает
CPU	%MLMTD	0	Если больше 0, то, возможно, VM уперлась в Limit по процессору
CPU	%SWPWT	5	VM ожидает чтения страниц из файла подкачки. Возможно, VM не хватает физической памяти
MEM	MCTLSZ (I)	0	Если больше 0, значит, механизм Balloon отнимает память у гостевой ОС
MEM	SWCUR (J)	0	Если больше 0, значит, часть памяти VM в файле подкачки VMkernel
MEM	CACHEUSD	0	Если больше 0, значит, часть памяти VM отнимается механизмом memory compression.
MEM	ZIP/s	0	Если больше 0, значит, VM использует память в кеше memory compression
MEM	UNZIP/s	0	Если больше 0, значит, VM использует память в кеше memory compression
MEM	SWR/s (J)	0	Чтение из файла подкачки. Если больше 0, значит, файл подкачки используется
MEM	SWW/s (J)	0	Запись в своп-файл. Если больше 0, значит, своп-файл используется
MEM	N%L	80	Если меньше 80, значит, VM работает неоптимально с точки зрения NUMA
DISK	GAVG (H)	25	Задержки для гостевой ОС. Это сумма счетчиков «DAVG» и «KAVG»
DISK	DAVG (H)	25	Задержки устройства хранения. Высокое значение этого счетчика говорит о задержках со стороны системы хранения
DISK	KAVG (H)	2	Задержки на уровне гипервизора
DISK	ABRTS/s	1	Отказы инициированные гостевой ОС (VM) из-за того, что СХД не отвечает. Для Windows это происходит через 60 секунд, по умолчанию. Могут быть вызваны отказом путей к LUN, или проблемами в работе multipathing

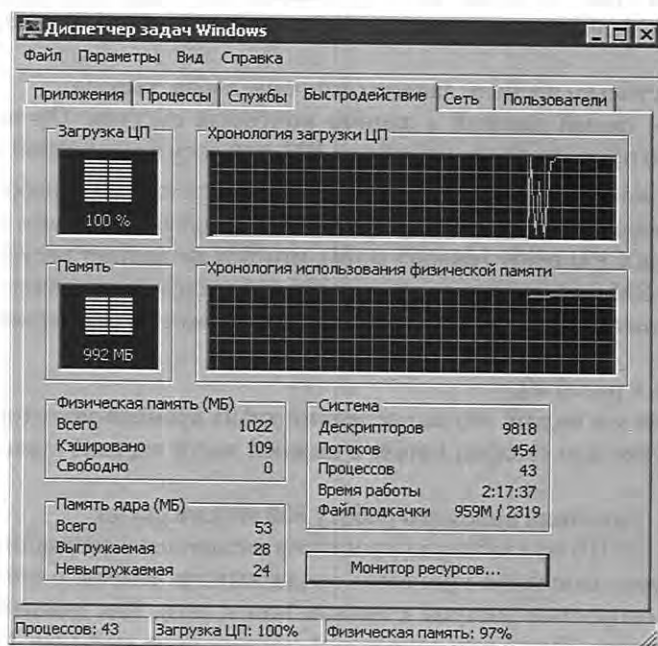
Таблица 6.1. Важные счетчики и их пороговые значения (окончание)

Под-система	Счетчик	Значение	Описание, возможная причина проблемы
DISK	QUED (F)	1	Превышена величина очереди команд
DISK	RESETS/s (K)	1	Число сброса SCSI команд в секунду
DISK	CONS/s	20	Число конфликтов SCSI reservation в секунду. Если происходит слишком много конфликтов, производительность СХД может деградировать из-за потерь времени вследствие резервирования LUN то одним, то другим сервером
NET	%DRPTX	0	Отброшенные передаваемые пакеты. Возможно сеть перегружена
NET	%DRPRX	0	Отброшенные принимаемые пакеты. Возможно сеть перегружена

Перечисленные в этой таблице значения являются пороговыми, в том смысле что показания выше (ниже) приведенных ненормальны и могут говорить о нехватке ресурсов той или иной подсистемы.

CPU

Итак, мы столкнулись с недостаточной производительностью сервера. Не в процессоре ли дело? Если речь идет о физическом сервере, то самый простой путь – это открыть Task Manager и посмотреть на загрузку процессора (рис. 6.51).

**Рис. 6.51. Менеджер процессов нагруженного сервера**

Однако что означает 100%-ная загрузка процессора?

Формально она означает следующее: на этом процессоре нет свободных тактов, на которых работает процесс `idle` (бездействие системы). В случае работы ОС на физическом сервере это обычно и означает, что ресурсов процессора недостаточно.

В случае же виртуальной машины гостевая ОС не имеет доступа к управлению ресурсами физических процессоров. Управляет доступом к ним гипервизор, и именно гипервизор выделяет или не выделяет процессорное время для ВМ.

Таким образом, 100%-ная загрузка процессора внутри ВМ означает, что гостевая ОС не видит свободных тактов процессора. Но не факт, что их нет на физическом процессоре, – может быть, их гипервизор не показывает, а выдает ровно столько процессорного времени, сколько готова задействовать гостевая ОС.

Определять же, хватает ли процессорных тактов ВМ или же гипервизор ей достаточно не выдает, нужно по-другому. Основных путей два – воспользоваться вкладкой **Performance** для ВМ в клиенте `vSphere` или консольной утилитой `esxtop` (`resxtop` в случае использования `vSphere CLI`).

Какие счетчики нам доступны? Фактически ВМ для гипервизора – это как процесс для обычной операционной системы. И счетчики очень похожие. Основные здесь:

- `usage` – сколько процессорных ресурсов ВМ задействовала. Выполненная работа. Не с точки зрения гостевой ОС, а с точки зрения гипервизора. Доступен в разных единицах измерения, смотря где и как мы смотрим показания этого счетчика. Может показываться в мегагерцах, процентах (от производительности одного LCPU) и миллисекундах (процессорного времени);
- `wait` – столько времени заняло ожидание ввода/вывода;
- `ready` – самый важный в данном контексте счетчик. Он показывает, на сколько процессорного времени у ВМ осталось несделанной работы.

В каждый момент времени ВМ готова выполнить какую-то работу. Эта работа делится на сделанную и несделанную, или `CPU usage` и `CPU ready`. Именно высокий показатель `CPU ready` говорит о том, что ВМ не хватает ресурсов процессора, – работа у ВМ есть, но гипервизор под нее не выделяет процессор.

Теперь поговорим про то, почему `CPU ready` может быть высокий и как с этим бороться.

Вернитесь к рис. 6.40.

На графике мы видим, что за последний квант времени мониторинга (правая точка на графике или столбец **Latest** в нижней части экрана) процессор номер 0 этой ВМ:

- **Used** – выполнял полезную работу 559 миллисекунд;
- **Wait** – 19 310 миллисекунд процессор находился в состоянии `wait`, то есть ожидания окончания процессов ввода/вывода. Вывод: у этой ВМ проблемы со скоростью доступа к дискам или к сети. Как мониторить диски и сеть – далее. Также есть отдельный счетчик `swap wait time`. Он показывает ожидание операции с файлом подкачки `VMkernel`. Это время входит в `wait`;



- **Ready** – гостевая ОС хотела выполнить работу, но гипервизор не дал достаточно процессорного времени на это – еще 37 миллисекунд. Вывод – гипервизор не ограничивает эту ВМ в процессорных ресурсах.

Обратите внимание. В CPU Ready показывается сумма этого показателя для всех процессоров виртуальной машины. То есть CPU Ready = 20 для четырехпроцессорной ВМ может означать очередь CPU Ready = 5 для каждого из процессоров, что является нормальным. Но возможно, что проблемы у какого-то одного виртуального процессора. Чтобы определить это, просмотрите данные по каждому виртуальному процессору этой ВМ.

Напомню, что означают эти миллисекунды. Один квант измерений – 20 секунд, или 20 000 миллисекунд (это просто факт). Таким образом, в примере выше из последних 20 000 мс времени, которые могли быть потрачены на полезную работу, процессор ВМ:

- 559 мс выполнял полезную работу (примерно 2,8% времени);
- 19 310 мс ждал отклика от дисков/сети (примерно 96,5% времени);
- 37 мс ВМ еще могла занять полезной работы, но гипервизор поставил эту работу в очередь (примерно 0,2%). CPU Ready редко равен 0 из-за погрешностей, значения менее 10% (2000 мс) на один vCPU считаются в пределах нормы.

Высокий CPU Ready – это плохо. Почему CPU Ready может быть высоким? Вариантов, в общем-то, несколько.

Во-первых, возможно, что ресурсов процессора сервера в принципе недостаточно на все ВМ. Наша ВМ претендует на какую-то долю в соответствии с настройками reservation и shares, но этой доли ей не хватает.

Решение – увеличить количество доступных ресурсов. Это можно сделать:

- выключив или перенеся на другие сервера часть ВМ;
- перенеся на более свободный сервер эту ВМ;
- увеличив ее reservation или shares;
- понизив reservation или shares других ВМ.

Частный случай предыдущего пункта – когда на самом сервере свободные ресурсы есть, но не хватает ресурсов в пуле ресурсов, где работает эта ВМ.

Решение:

- те же варианты, что и в предыдущем списке;
- увеличить долю ресурсов этого пула;
- понизить долю ресурсов других пулов.

Высокий CPU Ready отображает ситуацию «ВМ думает, что она может использовать сколько-то ресурсов, у нее есть работа для значительной доли этих мегагерц, но гипервизор ее искусственно ограничивает».

Но возможна другая проблемная ситуация – когда гипервизор отдает ВМ все, что может, но этого мало. А сколько это – «все, что может»?

Напомню, что один vCPU работает на одном и только на одном LCPU. Это значит, что однопроцессорная ВМ как максимум может задействовать ресурсы одного ядра сервера. Двухпроцессорная – двух ядер и т. д. Получается, что если ВМ,

например, однопроцессорная и приложению не хватает той производительности, что может дать одно ядро, вы получите нехватку ресурсов. Индикатором этого служит высокий показатель CPU Usage, близкий к 100% от максимума. Максимум, повторяюсь, равен количеству виртуальных процессоров этой ВМ, умноженному на тактовую частоту физического процессора.

Решение такой проблемы – дать ВМ больше виртуальных процессоров. Какие нюансы могут здесь нас поджидать:

- ❑ в одной ВМ может быть не больше 32 vCPU, когда ESXi имеет Enterprise Plus лицензию. И не больше восьми во всех других, включая бесплатную лицензию, случаях;
- ❑ приложение в ВМ должно быть многопоточным, чтобы получилось воспользоваться вторым и далее vCPU;
- ❑ используемая в ВМ ОС или приложение могут иметь собственные ограничения на количество поддерживаемых процессоров. Возможна ситуация, когда с точки зрения ESXi мы можем выдать 8 (например) процессоров для ВМ, а ПО внутри не может их задействовать из-за своих технических или лицензионных ограничений. Иногда это можно обойти с помощью изменения числа ядер виртуального процессора. Например, если мы выдали 8 потоков как 8 одноядерных vCPU, то Windows Server Standard Edition не сможет использовать все 8. А вот если их выдать как один восьмиядерный (или два четырехядерных) vCPU – сможет прекрасно;
- ❑ еще одна потенциальная причина того, что ВМ не выделяются такты, хотя они и есть свободные, – это настройка limit для процессоров ВМ (на уровне ВМ или ее пула ресурсов). Решение – увеличить limit.

Из общих рекомендаций следует отметить следующую – всегда настраивайте ВМ на использование как можно меньшего количества виртуальных процессоров. Только если вы однозначно уверены, что не хватает именно производительности одного физического ядра, то выдавайте ВМ второй (и т. д.) виртуальный процессор. Убедитесь, что производительность действительно увеличилась. Эта рекомендация является следствием того, что с ростом числа vCPU для ВМ уменьшается эффективность их использования и растут накладные расходы.

Приведенные здесь рекомендации по решению проблем с производительностью для процессора являются самыми простыми. Существуют и более глубокие методики, типа настройки ОС и приложений на использование больших страниц памяти или снижение частоты timer interrupt для гостевой ОС. Здесь я их не привожу в силу их большей зависимости от конкретного случая, конкретной гостевой ОС и прочего, так что см. документацию. Например, рекомендую обратиться к документам «Performance Troubleshooting for VMware vSphere 4» (<http://communities.vmware.com/docs/DOC-10352>) и «Performance Best Practices for VMware vSphere 5.0» (<http://www.vmware.com/resources/techresources/10199>).

MEMORY

Для оперативной памяти показателем недостаточности ресурсов является вытеснение виртуальной машины из памяти сервера в файл подкачки. Главный нюанс заключается в том, что использование VMkernel Swap, memory compression и



balloon изнутри ВМ определить невозможно. Поэтому опять же необходимо смотреть «снаружи», со стороны гипервизора.

Выделите ВМ ⇒ вкладка **Performance** ⇒ **Chart Options**. В левой части открывшегося окна выберите **Memory** и интересующий период времени. Справа доступны счетчики:

- ☐ **Consumed** – столько памяти для ВМ выделено из физической памяти сервера;
- ☐ **Granted** – это количество памяти сервер выделил для ВМ. Страница не выделяется, пока со стороны ВМ не будет хотя бы одного запроса для нее. Выданные страницы могут быть отображены механизмами `vmmemctl` и `VMkernel swap`;
- ☐ **Active** – с такого объема памятью ВМ активно работала в момент последнего опроса. Выделенная, но не активная память может быть отображена у ВМ без ущерба для ее производительности;
- ☐ **Balloon** – столько памяти отнимается у ВМ механизмом balloon;
- ☐ **Zipped memory** – столько памяти находится в сжатом виде, в буфере механизма `memory compression`;
- ☐ **Swapped** – столько памяти помещено в `VMkernel swap`;
- ☐ **Swap in rate** и **Swap out rate** – активность использования файла подкачки. Даже большие значения в `balloon` и `swapped` могут не означать проблемы производительности конкретной ВМ – этими механизмами гипервизор может отнимать у ВМ память, которую та в данный момент не использует. А нужна ли ВМ та память, что ESXi этими механизмами отбирает, показывают данные счетчики.

Таким образом, свидетельством нехватки памяти являются стабильно высокие показатели `balloon`, `zipped` и `swapped` вкупе с высокими показателями `Swap in rate` и `Swap out rate`.

Как бороться? Убедитесь, что ВМ не ограничена `limit` на ее уровне или на уровне пула ресурсов. Убедитесь, что на сервере есть свободная память, – обеспечьте ее, если это не так. Наконец, можно зарезервировать часть или весь объем памяти для этой ВМ.

DISK

С производительностью дисковой подсистемы в случае виртуальных машин все обстоит примерно так же, как и для машин физических. Специфика работы с дисковой подсистемой в случае виртуализации состоит в том, что от гостевой ОС до непосредственно дисков задействовано много этапов:

1. Гостевая ОС общается с драйвером SCSI контроллера (виртуального).
2. Он передает команды SCSI-контроллеру, виртуальному.
3. Они перехватываются гипервизором, гипервизор формирует очередь команд из обращений к диску своих ВМ и передает ее драйверу контроллера. Это может быть драйвер HBA, или служба NFS, или программный iSCSI-инициатор.
4. В зависимости от варианта эти команды быстрее или чуть медленнее попадают на контроллер, HBA или NIC.

5. Запрос уходит на систему хранения и затем в обратном порядке возвращается к гостевой ОС.

Обычно узким местом становится этап № 4 или 5. В каких-то ситуациях мы можем упереться в пропускную способность канала передачи данных. В каких-то – в количество операций ввода/вывода, которые способна обработать система хранения.

Если вернуться к таблице важных счетчиков, то для дисковой подсистемы:

- ☐ GAVG – это этапы со второго по пятый;
- ☐ KAVG – это этапы три и четыре;
- ☐ DAVG – это этап пять.

Данные по нагрузке на дисковую подсистему нам могут предоставить клиент vSphere, esxtop и утилита vscsiStats. Последняя предоставляет более детальные и низкоуровневые данные. Подробная информация о ней приведена по ссылке <http://communities.vmware.com/docs/DOC-10095>.

В крупных инфраструктурах часто имеет смысл обращаться к статистике нагрузки со стороны системы хранения. СХД высокого уровня имеют соответствующие возможности.

В инфраструктурах любого размера внимательно изучайте документацию по настройке имеющейся СХД под ESXi – проблемы производительности из-за неправильной настройки чрезвычайно обидны.

Еще плохими показателями для системы хранения являются высокие показатели латентности и дисковой очереди.

Network

Для сети нас в первую очередь интересует количество отброшенных пакетов. Это счетчики droppedRx и droppedTx в графическом интерфейсе, или %DRPTX и %DRPRX в esxtop.

Высокие показатели этих счетчиков свидетельствуют о плохой производительности сети.

Однако в большинстве случаев проблемы с производительностью сети вызваны внешней относительно серверов ESXi частью инфраструктуры.

Разве что упомяну, что мне приходилось слышать о проблемах в виде роста числа отброшенных пакетов при использовании балансировки нагрузки по методу ip hash. Но вообще это нетипичная ситуация, и обычно такого не бывает.

6.4. Механизм Alarm

ESXi предоставляет большое количество информации о состоянии инфраструктуры. Это и разнообразные счетчики производительности, и данные по состоянию серверов, наличие сигналов пульса (heartbeat) от VMware tools внутри ВМ, журнал событий (events) и др. Однако часто бывает полезной не только сама по себе возможность посмотреть данные, но и получить автоматически оповещение или иную реакцию.

Для этого в vCenter предусмотрен механизм Alarms – обратите внимание на одноименную вкладку для датацентров, серверов, кластеров, виртуальных машин, пулов ресурсов и папок (рис. 6.52).

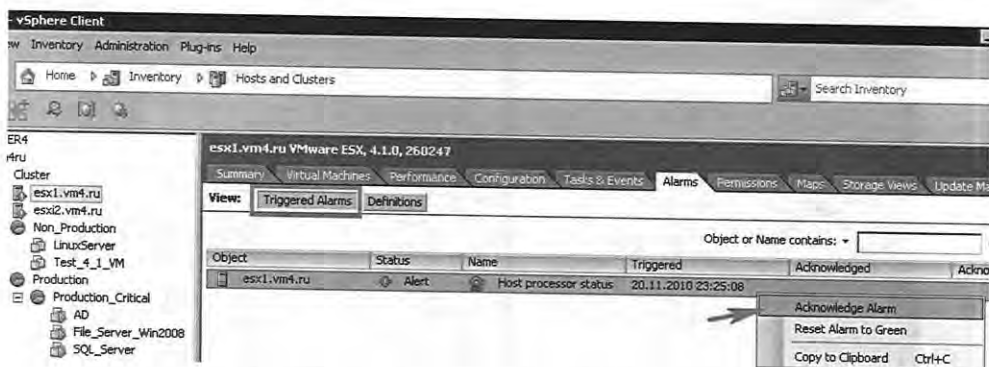


Рис. 6.52. Сработавший alarm для сервера

Каждый alarm – это суть триггер, отслеживающий событие или состояние счетчика нагрузки. Alarm могут мониторить счетчики для объектов разных типов – ВМ, серверов, сетей, хранилищ. Притом отслеживаться может как показатель нагрузки (например, процент загрузки процессора или свободное место на диске), так и состояние (включена ли ВМ, доступен ли сервер по сети).

Кроме того, alarm может отслеживать событие (event) с указанными параметрами.

Для создания Alarm перейдите на желаемый уровень иерархии. Например, если я хочу создать alarm для мониторинга виртуальных машин, то выберу Datacenter в случае, когда хочу отслеживать сразу все ВМ в нем. Если же я хочу отслеживать только группу ВМ, то перейду на соответствующий пул ресурсов, vApp или каталог для виртуальных машин.

Затем следует перейти на вкладку **Alarms** ⇒ кнопка **Definitions**. Там мы увидим все актуальные для этого уровня иерархии alarm. В столбце Defined In указано, с какого объекта они наследуются. Выбрав пункт **New Alarm** в контекстном меню пустого места этой вкладки, мы запустим мастер создания нового alarm.

На первой вкладке вводим имя alarm и что он должен мониторить (рис. 6.53).

Вариантов типов объектов, как вы видите, много. Также здесь мы указываем, что мы хотим мониторить, – состояние какого-либо счетчика или состояние объекта, или событие, произошедшее с объектом.

На вкладке **Triggers** мы указываем условия срабатывания alarm. Их может быть несколько, alarm может срабатывать как при выполнении любого, так и сразу всех условий. Кроме того, мы можем указать пороговые значения, при которых alarm меняет статус объекта на **Warning** (Предупреждение) или **Error** (Ошибка).

На вкладке **Reporting** мы указываем параметры срабатывания alarm.

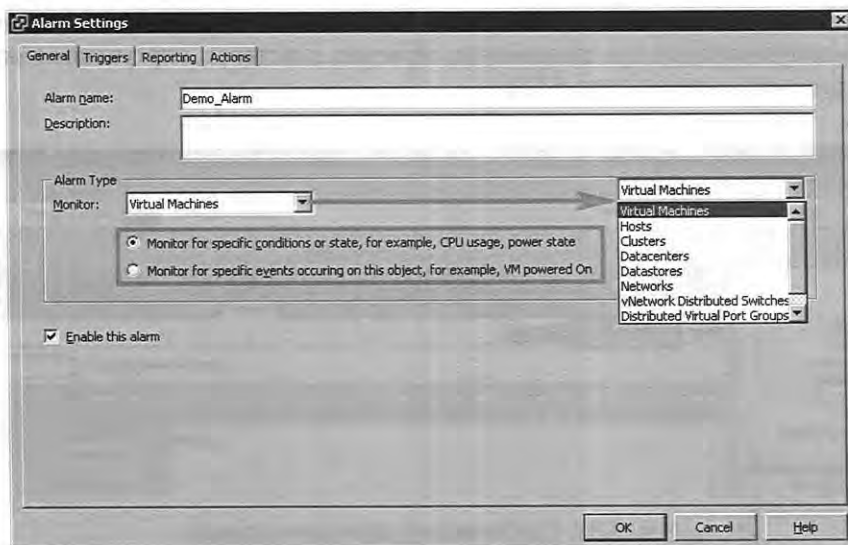


Рис. 6.53. Создание alarm

Range – диапазон, при выходе из которого alarm меняет статус. То есть «наступление порогового значения» плюс «диапазон» = сработавший alarm. Например, если вы указали срабатывание alarm при 70%-ной загрузке процессора, а range = 5, то alarm сработает при загрузке выше 75% ($= 70 + 5$) и вернется в нормальное состояние при загрузке ниже 65% ($= 70 - 5$).

Trigger Frequency – в течение этого количества минут после срабатывания alarm не сработает еще раз.

Что бы ни отслеживал тот или иной alarm, ключевым следствием этого является возможность реакции на событие. При создании аларма мы указываем условия смены статуса объекта, за которым наблюдаем. На последней вкладке мы указываем действие, которое необходимо предпринять при смене статуса.

Среди действий мы встретим (в зависимости от типа аларма и за кем он наблюдает):

- ❑ оповещение по электронной почте. Действие доступно для объектов всех типов. Письмо будет отправлять vCenter, поэтому в меню **Home** ⇒ **vCenter Server Settings** ⇒ **Mail** нужно указать настройки почтового сервера;
- ❑ оповещение по SNMP. Действие доступно для объектов всех типов. Трассообращения будет рассылать сервер vCenter, поэтому в меню **Home** ⇒ **vCenter Server Settings** ⇒ **SNMP** нужно указать адреса получателей и строки community;
- ❑ запуск произвольной команды. Действие доступно для объектов всех типов. В столбце **Configuration** указываются путь и параметры запускаемой программы. Например:

```
c:\windows\system32\cmd.exe /c c:\tools\cmd.bat
```

или просто

```
c:\a.bat
```

А в этом командном файле может быть указано выполнение сценария PowerShell:

```
%SystemRoot%\system32\windowspowershell\v1.0\powershell.exe -psc "C:\Program Files (x86)\VMware\Infrastructure\vsphere PowerCLI\vim.psc1" -command "c:\posh_script.ps1"
```

Также в качестве параметров можно передавать некоторые поля alarm. Например:

```
c:\tools\sendsms.exe AlarmName targetName
```

Список доступных полей я не обнаружил в документации к vSphere 5, но он доступен в документации к vSphere 4: **vSphere 4 – ESXi Installable and vCenter Server ⇒ vSphere Basic System Administration ⇒ System Administration ⇒ Working with Alarms ⇒ Alarm Actions ⇒ Running Scripts as Alarm Actions.**

Указанная команда будет выполнена на сервере vCenter отдельным от службы vCenter потоком;

- ☐ для виртуальных машин – включение, выключение, пауза, миграция, перезагрузка;
- ☐ для серверов – ввод в режим обслуживания, вывод из режима обслуживания, отключение от vCenter, перезагрузка, выключение.

Аларм, созданный на одноименной вкладке на каком-то уровне иерархии vCenter, мониторит объекты этой ветки иерархии. Например, аларм мониторинга ВМ, созданный на уровне Datacenter, мониторит все ВМ. А созданный на уровне каталога для ВМ – все ВМ этого каталога.

Обратите внимание: в контекстном меню большинства объектов иерархии vCenter есть пункт **Alarm ⇒ Disable Alarms Actions** (рис. 6.54).

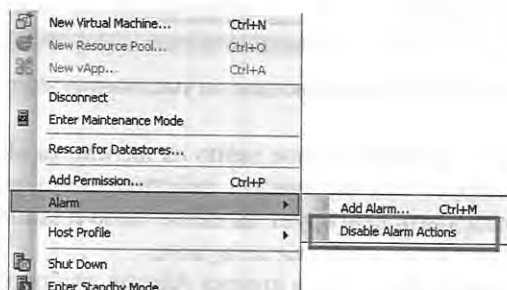


Рис. 6.54. Выключение реакции alarm

Этот пункт нужен для отключения реакции (но не факта срабатывания) alarm для данного объекта. Опять же, востребовано обычно во время каких-либо плановых процедур с инфраструктурой, которые могут вызвать нежелательные сраба-

тывания alarm. Срабатывание автоматических реакций alarm отключается, пока их не включите обратно, из того же самого меню.

Существующие по умолчанию alarms созданы для самого верхнего уровня иерархии. Найти их, чтобы посмотреть свойства, поменять свойства или удалить, можно, выбрав vCenter в иерархии ⇒ вкладка **Alarms** ⇒ кнопка **Definitions** (рис. 6.55).

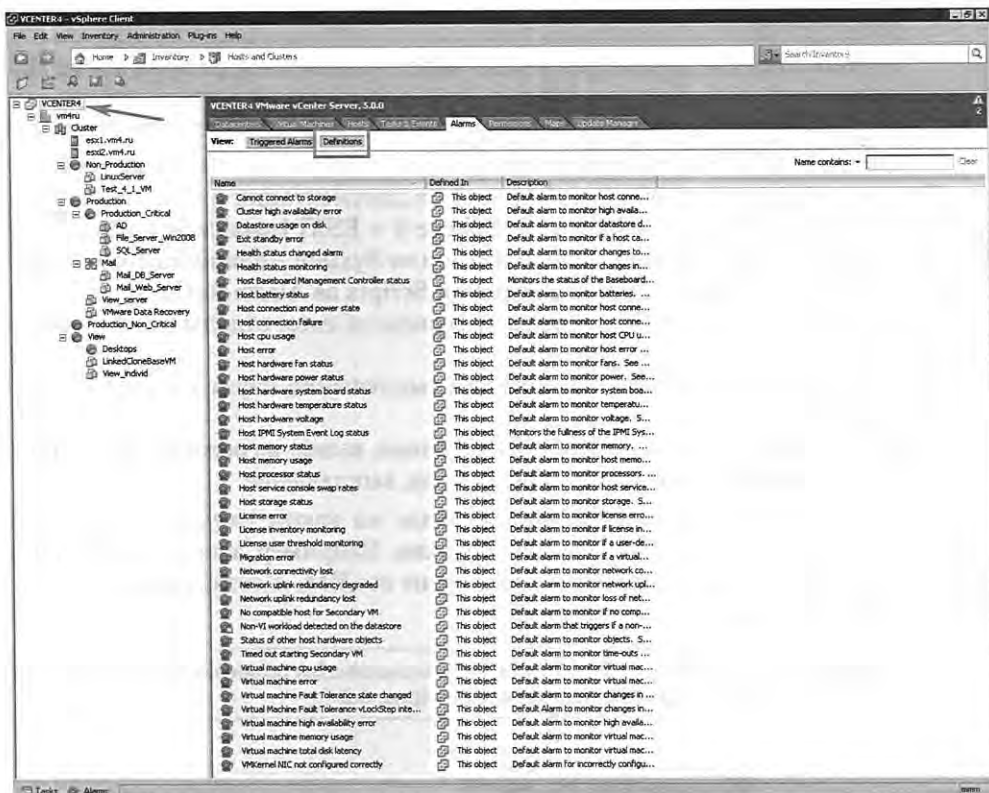


Рис. 6.55. Существующие по умолчанию alarms

Обратите внимание на контекстное меню на активном alarm. Пункт acknowledge (рис. 6.40) блокирует автоматическое действие alarm, не сбрасывая его статус. Это полезно, когда alarm реагирует на какое-либо плановое событие, о котором вы знаете и реагировать на которое не нужно.

Также в нижней части экрана есть кнопка **Alarms**, выбор которой покажет все активные на данный момент alarm (рис. 6.56).

Приведу пример применения этого механизма. Допустим, перед нами поставили задачу – эвакуировать виртуальные машины с сервера, на котором произошел отказ компонента. Из тех соображений, что отказ компонента, не приведший

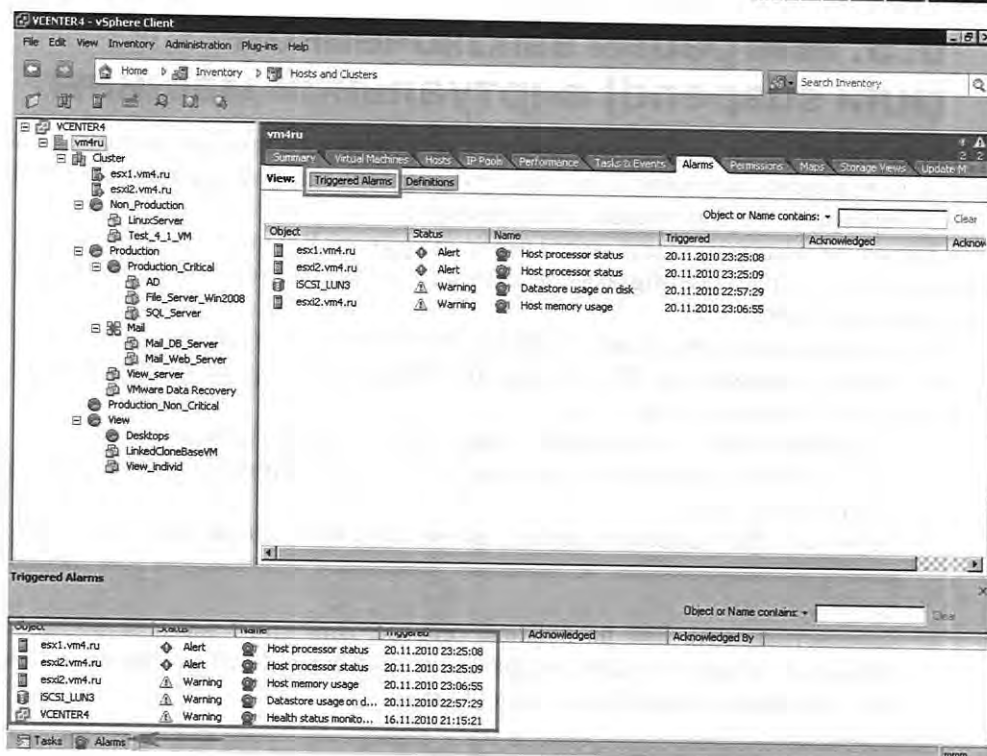


Рис. 6.56. Просмотр сработавших alarms

к отказу сервера, все равно снижает его надежность. Например, если отказал один из двух блоков питания. Для этого в клиенте vSphere пройдите **Home** ⇒ **Inventory** ⇒ **Hosts and Clusters** и в левом дереве выберите объект высокого уровня иерархии, например корень, Datacenter или кластер, – чтобы все наши сервера были ниже по иерархии.

Вас интересует вкладка **Alarms** ⇒ кнопка **Definitions**. Контекстное меню пустого места ⇒ **New Alarm**. Мониторить будем сервера (выпадающее меню **Monitor**), для серверов мониторить будем состояние (**Monitor for specific events...**).

На вкладке **Triggers** добавьте условие **Hardware Health Changed**. А на вкладке **Actions** добавьте реакцию – **Enter maintenance mode**. Также при необходимости добавьте оповещение по SNMP или e-mail.

Теперь в случае ухудшения в работе компонентов сервера vCenter переведет его в Maintenance-режим, что вызовет миграцию виртуальных машин с этого сервера на прочие (если сервер в кластере DRS) и воспрепятствует запуску ВМ на данном сервере.

Больше информации о возможностях механизма alarm вы можете подчерпнуть по ссылке <http://communities.vmware.com/docs/DOC-12145>.

6.5. Миграция выключенной (или suspend) виртуальной машины

Если виртуальная машина выключена или находится в состоянии паузы (suspend), то ее можно мигрировать как между серверами, так и между хранилищами, между серверами и хранилищами одновременно.

Запуск миграции осуществляется перетаскиванием ВМ на нужный сервер или хранилище, или пунктом **Migrate** контекстного меню ВМ. Условий ни на ВМ, ни на сервера не налагается.

Если необходимо мигрировать ВМ без участия vCenter (клиент vSphere подключен напрямую к серверу ESXi), то пункт **Migrate** или перетаскивание вам недоступно. Тогда можно сделать так:

1. Выключить ВМ или перевести в состояние паузы (suspend).
2. Удалить ВМ из иерархии объектов ESXi – пункт **Remove from Inventory** контекстного меню.
3. Перенести файлы ВМ на другое хранилище, если необходимо. Для этого можно воспользоваться встроенным файловым менеджером или любым другим.
4. Зарегистрировать ВМ на нужном сервере. Для этого через встроенный файловый менеджер найти ее файлы и выбрать **Add to inventory** в контекстном меню ее файла настроек (*.vmx).

Если есть необходимость перенести ВМ с сервера на сервер или на другое хранилище с минимальным простоем, но vMotion/Storage vMotion недоступен, то можно сделать так:

1. Для работающей ВМ создать снимок состояния (snapshot).
2. Скопировать все ее файлы, кроме файлов последнего снимка состояния, на новое хранилище.
3. Перевести ВМ в состояние паузы (suspend).
4. Перенести на другое хранилище оставшиеся файлы ВМ (это файлы последнего снимка и файл с памятью ВМ в состоянии паузы).
5. Через встроенный файловый менеджер найти скопированные файлы на другом сервере и выбрать **Add to inventory** в контекстном меню ее файла настроек (*.vmx). Если копирование на другое хранилище этого же сервера, то тогда исходную ВМ нужно удалить пунктом **Remove from inventory**, а скопированную добавить на этом же сервере.
6. Включить ВМ на новом месте, удалить снимок состояния, удалить исходную ВМ.

6.6. Storage vMotion – живая миграция файлов VM между хранилищами

Storage vMotion, иногда SvMotion – это перенос файлов VM с хранилища на хранилище без ее выключения. Поддерживаются хранилища любых типов, включая локальные диски. Поддерживается перенос как всей VM, так и только одного или нескольких ее файлов vmdk.

Кроме переноса файлов VM с хранилища на хранилище, Storage vMotion пригодится для:

- ☐ конвертации диска VM между форматами thin и thick;
- ☐ копирования RDM в файл vmdk. Обратный процесс, таким образом, невозможен.

Суть процесса в следующем:

1. Когда администратор инициирует Storage vMotion, начинается копирование ее файлов на новое место. Разумеется, больше всего проблем с файлами дисков – они большие, копировать их долго, и работающая VM их изменяет по ходу копирования. В пятой версии vSphere применен следующий подход для решения проблемы с копированием изменяемого файла – все изменения зеркалируются, то есть записываются и в исходный, и в новый файл-диск.
2. Основной объем информации (файлы vmdk) копируется. Притом требуется только одна итерация, потому что все данные, которые были изменены во время самой процедуры переноса, сразу были записаны и в новую копию.
3. В последний миг миграции, когда были скопированы последние из старых данных, гипервизор отправляет запросы VM уже к новым файлам. Старые файлы удаляются.

Для запуска этого процесса выберите **Migrate** в контекстном меню VM и **Change Datastore** на первом шаге мастера.

Или перейдите **Home** ⇒ **Datastore** ⇒ хранилище с VM ⇒ вкладка **Virtual Machines** и перетащите нужную VM на другое хранилище.

В мастере будут шаги:

1. **Select Datastore** – здесь вы выберете, на какое хранилище переносить VM. А если нажать кнопку **Advanced**, то можно указать миграцию только отдельных дисков этой VM;
2. **Disk Format** – здесь можно указать тип диска для VM на новом хранилище. Thick, Thin или тот же, что и сейчас. Если у VM есть RDM-диски, то при выборе здесь Same format as source они останутся неизменными, скопируется лишь vmdk-ссылка на RDM. При выборе thin или thick содержимое RDM LUN скопируется в файл vmdk на указанном хранилище.

Условий на тип хранилища нет – возможен перенос ВМ между системами хранения любых типов, включая локальные диски и DAS. Но еще раз обращаю ваше внимание на то, что процесс Storage vMotion оставляет ВМ на том же самом сервере.

Для виртуальных машин условий нет.

Для сервера условие, в общем-то, одно – чтобы была лицензия на Storage vMotion.

6.7. vMotion – живая миграция ВМ между серверами

vMotion – это название процесса живой миграции. То есть переезда виртуальной машины с одного сервера на другой без перерыва в работе. Живая миграция пригодится вам:

- когда необходим плановый простой сервера. ВМ с требующего выключения сервера можно убрать на другие сервера и спокойно выключить его;
- для балансировки нагрузки. С загруженного сервера можно мигрировать виртуальные машины на более свободные сервера.

Как только мы запускаем миграцию, vCenter проверяет выполнение условий для виртуальной машины и серверов, между которыми она мигрирует. Об условиях далее.

Затем гипервизор начинает копировать содержимое оперативной памяти этой виртуальной машины на другой ESXi. Но ведь виртуальная машина продолжает работать, и за время копирования содержимого оперативной памяти это содержимое успеет поменяться. Поэтому гипервизор начинает вести список адресов измененных блоков памяти перемещаемой виртуальной машины. ВМ продолжает работать, но адреса измененных страниц в ее памяти фиксируются.

Итак, основной объем памяти передается на другой ESXi через интерфейс VMkernel, который мы задействуем под vMotion.

Как только вся память передалась – ВМ блокируется полностью, и на второй ESXi передаются измененные страницы памяти. Так как их объем будет небольшой – время, в течение которого ВМ полностью блокируется, также невелико. Весьма невелико. А если объем измененной памяти будет больше некоего порогового значения, значит, ESXi просто повторит итерацию. Благодаря этому область памяти, для передачи которой ВМ полностью заморозится, обязательно будет очень небольшой, пусть и через несколько итераций.

На этом этапе мы имеем два идентичных процесса, две идентичные ВМ на обоих серверах ESXi. Теперь ВМ на исходном сервере убивается, по сети идет оповещение, что машина с этим MAC-адресом доступна уже на другом порту физического коммутатора. Все. В подавляющем большинстве случаев переключение между ВМ на старом и новом сервере занимает менее одной секунды.

Если на каком-то этапе идет сбой, то ВМ просто не убивается на исходном сервере и никуда не уезжает, но падения ВМ из-за неудавшегося vMotion не происходит.

Для того чтобы какую-то VM можно было мигрировать без простоя между двумя серверами, должны быть выполнены некоторые условия для этой виртуальной машины и этих серверов.

Основное условие vMotion, налагаемое на сервера, – процессоры серверов должны быть совместимы с точки зрения vMotion. Дело в том, что процессор – единственная подсистема сервера, которую виртуальные машины (гостевые ОС) видят такой, какая она есть физически. Не имеет значения, если у процессоров этих серверов окажутся разные тактовые частоты, размер кеш-памяти, количество ядер. А имеет значение набор поддерживаемых инструкций, таких как SSE 3, SSE 4.1, NX/XD и др. Если на двух разных серверах разные процессоры, а приложение использует какую-то из инструкций, что была доступна до, но недоступна после переезда, – то приложение упадет.

Дабы не допустить этого, vCenter не позволит начать vMotion между серверами с несовместимыми процессорами. Кстати, не забудьте, что часть функций процессора управляется BIOS, так что и сервера с идентичными процессорами могут быть непригодны для горячей миграции между ними VM, если настройки BIOS отличаются. Иногда проще всего сбросить их на значения по умолчанию.

В идеале процессоры у вас совместимы для vMotion (подробности доступны в базе знаний VMware). Если нет, но живая миграция очень нужна, вам могут помочь следующие варианты:

- ❑ редактирование так называемой CPUID Mask (свойства VM ⇒ **Options** ⇒ **CPUID Mask**). Суть в том, что для конкретной VM мы можем «спрятать» те процессорные инструкции, что мешают миграции. Подробные инструкции вы найдете в базе знаний VMware (<http://kb.vmware.com/kb/1993>);
- ❑ в принципе, отключение самой проверки на одинаковость процессоров, которую делает vCenter. Действие не поддерживаемое, но работающее. Конечно, данное решение имеет смысл использовать, лишь если вы уверены, что приложения в ваших VM не задействуют тех инструкций, которыми отличаются процессоры серверов. Для отключения проверки необходимо вставить строки

```
<migrate>
  <test>
    <CpuCompatible>>false</CpuCompatible>
  </test>
</migrate>
```

в файл %AllUsersProfile%\Application Data\VMware\VMware VirtualCenter\vpzd.cfg и перезапустить службу vCenter;

- ❑ Enhanced vMotion Compatibility, EVC. Что это такое, можно прочитать в начальных разделах книги, как это включается – в разделе про DRS. Однако EVC включается и для кластера, в котором DRS не включен. По факту имеет смысл пользоваться только этим механизмом, про остальные я упомянул лишь для справки.

Вернемся к условиям, которые налагаются на VM и сервера из-за vMotion.

Все ресурсы, которые задействует ВМ, должны быть доступны на обоих серверах. Это:

- ❑ разумеется, файлы самой ВМ. vmx, vmdk и прочие (за исключением файла подкачки vswp). Если резюмировать – ВМ должна быть расположена на общем хранилище. Какого именно типа – FC, iSCSI или NFS, не важно. RDM также поддерживается, но LUN, подключенный как RDM, должен быть виден обоим серверам;
- ❑ для виртуальных SCSI-контроллеров настройка **SCSI Bus Sharing** не должна быть выставлена в значение, отличное от **None**. Это означает, что виртуальные машины-узлы кластера Майкрософт не могут быть мигрированы с помощью vMotion;
- ❑ к ВМ могут быть подключены образы CD-ROM и Floppy. Эти файлы также должны быть доступны с обоих серверов;
- ❑ к ВМ не должен быть подключен CD-ROM сервера, то есть настройка «Host Device» в свойствах виртуального CD-ROM. Те же условия верны для FDD;
- ❑ к ВМ не должны быть подключены физические COM- и LPT-порты сервера;
- ❑ у ВМ не должно быть настроено CPU Affinity – указание конкретных ядер, на которых должна работать эта ВМ;
- ❑ группы портов, к которым подключена ВМ, должны существовать на обоих серверах. Проверяется только совпадение имен с точностью до регистра;
- ❑ ВМ не должна быть подключена к виртуальному коммутатору без привязанных физических сетевых контроллеров. Однако проверку на это можно отключить. Для этого вставьте строки

```
<migrate>
  <test>
    <CompatibleNetworks>
      <VMOnVirtualIntranet>false</VMOnVirtualIntranet>
    </CompatibleNetworks>
  </test>
</migrate>
```

в файл %AllUsersProfile%\Application Data\VMware\VMware VirtualCenter\vpzd.cfg и перезапустите службу vCenter;

- ❑ в процессе vMotion между серверами передается содержимое оперативной памяти ВМ. Это содержимое передается между интерфейсами VMkernel, так что мы должны создать эти интерфейсы. Обратите внимание, что любой интерфейс VMkernel может одновременно использоваться еще и для NFS, iSCSI, Fault Tolerance, и для управляющего трафика. Однако для vMotion рекомендуется выделить отдельный гигабитный интерфейс и, как следствие, отдельный интерфейс VMkernel. В свойствах виртуального сетевого контроллера VMkernel, который вы планируете задействовать под vMotion, необходимо поставить флажок «Use this port group for vMotion».

Обратите внимание. vCenter проверяет только факт наличия интерфейсов VMkernel с флажком **Use this port group for vMotion** в свойствах, но не наличие связи между этими интерфейсами разных серверов. Чтобы убедиться, что в конфигурации сети нет ошибок, на одном из серверов выполните команду `ping <IP vMotion-интерфейса VMkernel второго сервера>`.

Для проверки выполнения большей части условий из списка выше хорошо подойдет вкладка **Maps** для виртуальной машины (рис. 6.57).

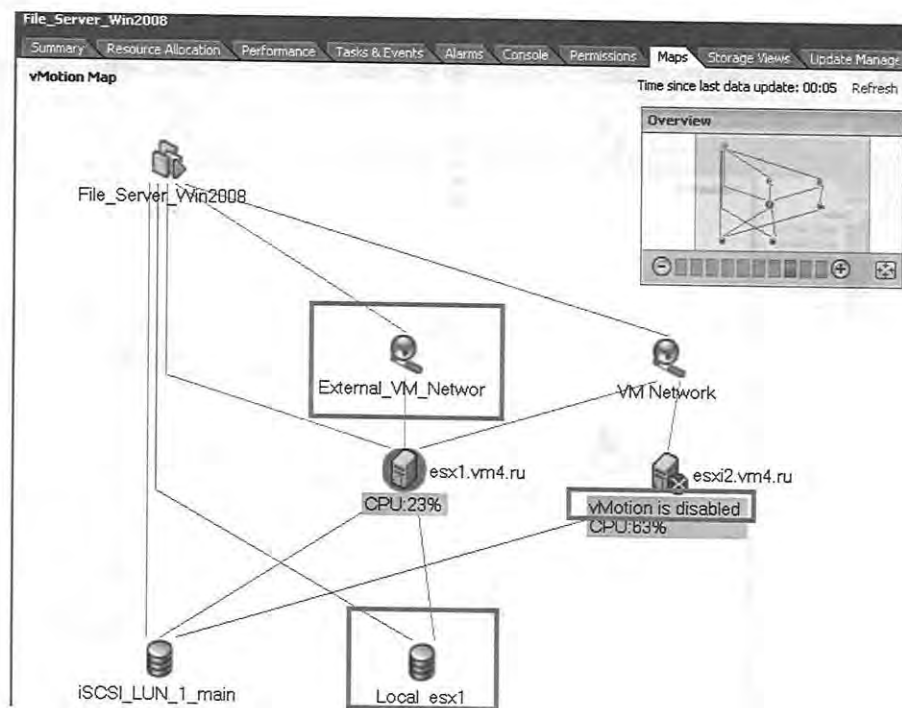


Рис. 6.57. **Maps** для VM, которая не может мигрировать на горячую

Обратите внимание на рисунок – на нем показана схема виртуальной инфраструктуры в контексте vMotion одной VM. Сейчас виртуальная машина «File_Server_Win2008» работает на esx1.vm4.ru. Мною обведены те объекты, на которые стоит обратить внимание, так как они препятствуют vMotion этой VM на второй сервер:

- ❑ это группа портов «External_VM_Networ» – она есть на одном сервере, но на сервере esxi2.vm4.ru ее нет. Решение – создать группу портов с тем же именем на сервере esxi2. Или перестать ее задействовать для данной VM;
- ❑ данная VM задействует два хранилища – «iSCSI_LUN_1_main» и «Local_esx1». Притом второе недоступно с сервера esxi2. Решение – или сделать «Local_esx1» доступным со второго сервера (не всегда возможно), или

перенести ВМ или ее диск, расположенные на «Local_esx1», в другое хранилище, доступное обоим серверам;

- ❑ «vMotion is disabled» на esxi2 означает, что на этом сервере нет ни одного интерфейса VMkernel, в свойствах которого разрешен трафик vMotion. Решение – создать такой интерфейс или поставить соответствующий флажок в свойствах существующего интерфейса.

Для того чтобы понять, какие группы портов и какие хранилища доступны с обоих серверов, поможет **Home** ⇒ **Maps**. На рис. 6.58 вы видите пример такой карты.

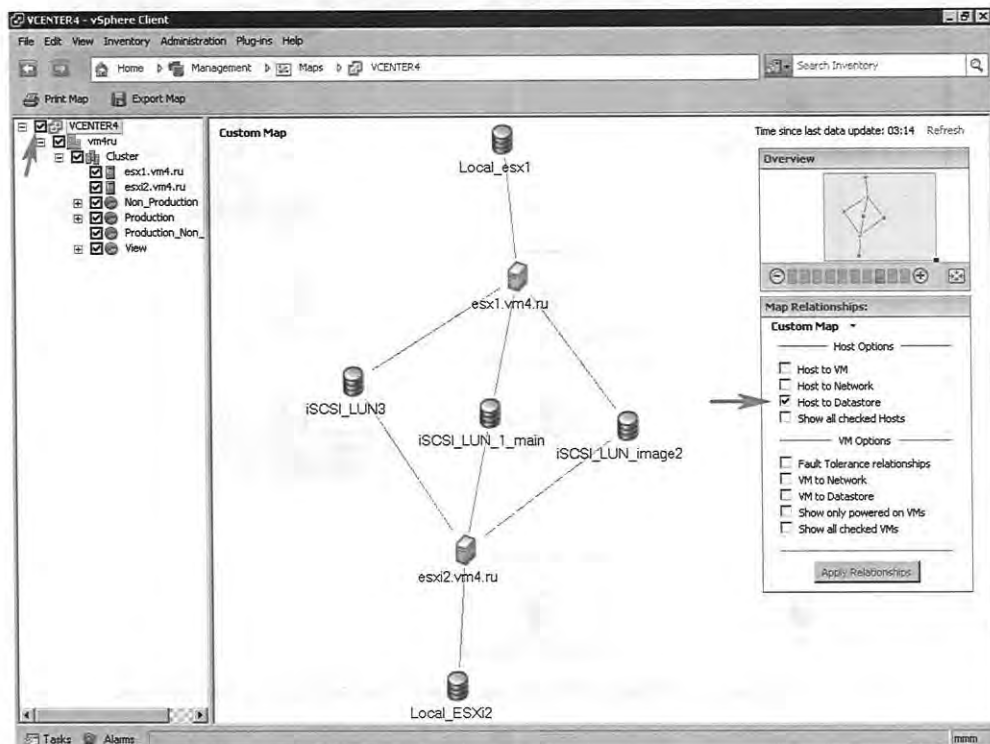


Рис. 6.58. Карта связей хранилищ с серверами

Если все проблемы решены, то карта vMotion должна выглядеть примерно так, как показано на рис. 6.59.

Обратите внимание, что через **Maps** не отображается информация о RDM-дисках ВМ. Виртуальную машину с RDM мигрировать с помощью vMotion возможно, однако подключенный как RDM LUN должен быть презентован обоим серверам. Проверить это через **Maps** нельзя. Впрочем, если к ВМ будет подключен RDM LUN, видимый лишь текущему серверу, об этом вам скажут на первом шаге мастера vMotion.

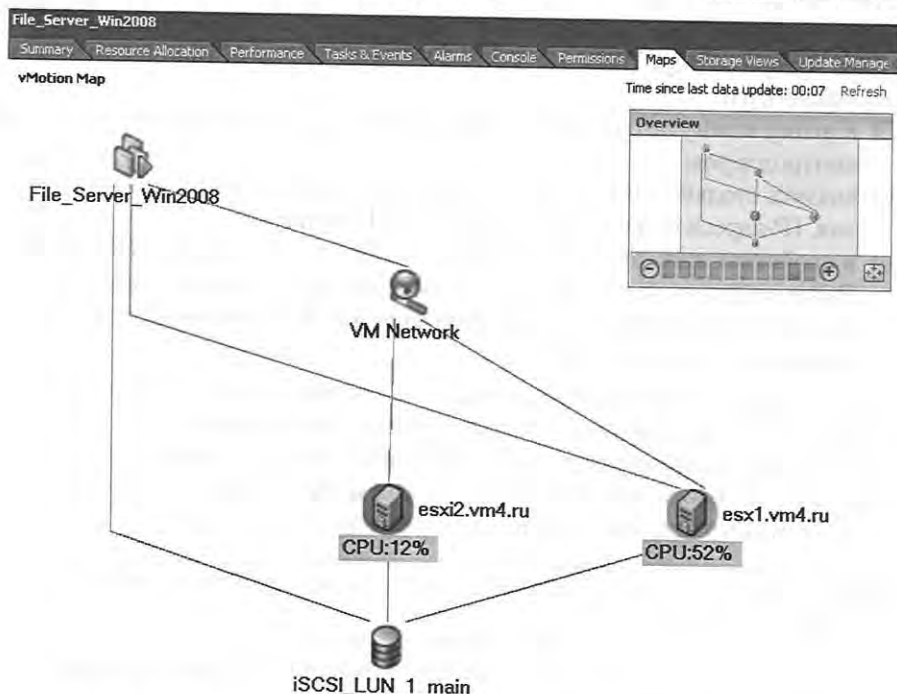


Рис. 6.59. **Maps** для ВМ, которая может мигрировать на горячую

Единственным файлом ВМ, который не обязан лежать на общем хранилище, является файл подкачки (VMkernel swap). Даже если он расположен на приватном для первого сервера ресурсе, при миграции ВМ будет создан файл подкачки на диске, видимом второму серверу, и содержимое будет перенесено. Но на том диске, где второй сервер располагает файлы подкачки работающих на нем ВМ, должно быть достаточно свободного места, иначе vMotion не произойдет.

Запустить сам процесс миграции можно, выбрав **Migrate** в контекстном меню ВМ, затем выбрав **Change host** на первом шаге мастера. Чаше проще перетащить ВМ на нужный ESXi – тогда в мастере vMotion будет меньше вопросов. Останутся лишь:

- ☐ **Select Resource Pool** – помещать ли ВМ в какой-то пул ресурсов, и если да, то в какой;
- ☐ **vMotion Priority** – резервировать ли ресурсы под перемещаемую ВМ на сервере назначения (выбрано по умолчанию). Или не резервировать. Второй вариант позволит мигрировать ВМ в условиях недостатка ресурсов, пусть даже сама миграция займет больше времени.

В пятой версии vSphere была реализована возможность миграции ВМ сразу по нескольким сетевым контроллерам – это может значительно повысить скорость миграции ВМ с большими объемами памяти при использовании гигабитных сетевых соединений.

Для реализации этой возможности следует определить виртуальный коммутатор, на котором будут (или уже) созданы интерфейсы VMkernel для vMotion. Дальнейшие шаги:

- ☐ к этому коммутатору следует подключить несколько физических сетевых контроллеров;
- ☐ создать столько же интерфейсов VMkernel с флажком vMotion в настройках. IP-адреса нужно указать из одной IP-подсети;
- ☐ в свойствах каждой группы портов VMkernel на вкладке **NIC Teaming** ⇒ **Failover Order** следует выбрать только один физический интерфейс как активный. Разумеется, выбрать надо отдельный физический интерфейс для каждого интерфейса VMkernel.

Когда эти настройки будут выполнены на каждом сервере, миграция ВМ между ними будет происходить сразу по нескольким интерфейсам.

Для проверки корректности работы такой конфигурации ознакомьтесь с файлом журнала гипервизора vmkernel.log – в нем вы должны обнаружить записи о том, что для миграции ВМ установлено несколько соединений:

```
cpu1:10496)MigrateNet: 1155: 1328264432389309 S: Successfully bound connection to
vmknics '192.168.111.4'
cpu1:10496)VMotionUtil: 3118: 1328264432389309 S: Stream connection 1 added.
cpu1:10496)MigrateNet: 1155: 1328264432389309 S: Successfully bound connection to
vmknics '192.168.111.14'
cpu1:10496)VMotionUtil: 3118: 1328264432389309 S: Stream connection 2 added.
```

Обратите внимание. Имеется в виду именно и только vMotion, для Storage vMotion и миграции выключенных ВМ данная информация не применима.

6.8. Кластер DRS. DPM

Живая миграция ВМ – это полезная штука. Она позволяет мигрировать ВМ между серверами для:

- ☐ балансировки нагрузки между серверами;
- ☐ освобождения сервера, когда нужно его перезагрузить. Это пригодится при установке обновлений, аппаратном обслуживании.

Однако при мало-мальски большой инфраструктуре выполнять эти операции вручную для администратора становится затруднительно. На помощь в этом ему может прийти VMware DRS, Distributed Resource Scheduler.

Зачем нужен DRS:

- ☐ для балансировки нагрузки (по процессору и памяти) между серверами;
- ☐ для автоматического vMotion виртуальных машин с сервера в режиме обслуживания (maintenance mode). Этим режимом администратор или VMware Update Manager помечает сервер, который надо освободить от виртуальных машин перед операциями обслуживания.

Для решения этих задач DRS умеет осуществлять всего две вещи:

- ☐ запускать vMotion для виртуальных машин и выбирать, откуда, куда и какую ВМ лучше мигрировать;
- ☐ при включении ВМ выбирать сервер, на котором она включится (это называется Initial Placement).

Для создания DRS-кластера необходимо создать объект «Cluster» в иерархии vCenter. В контекстном меню Datacenter выберите **Add new Cluster**, укажите имя создаваемого кластера и поставьте флажок **DRS** (не имеет значения, стоит ли флажок HA). Нажмите **OK**.

Итак, кластер как объект vCenter вы создали. Осталось два шага:

1. Настроить DRS-кластер.
2. Добавить в него сервера ESXi.

Впрочем, включить DRS можно и для уже существующего кластера.

Если для кластера активирован DRS, то вам доступны следующие группы настроек (рис. 6.60):

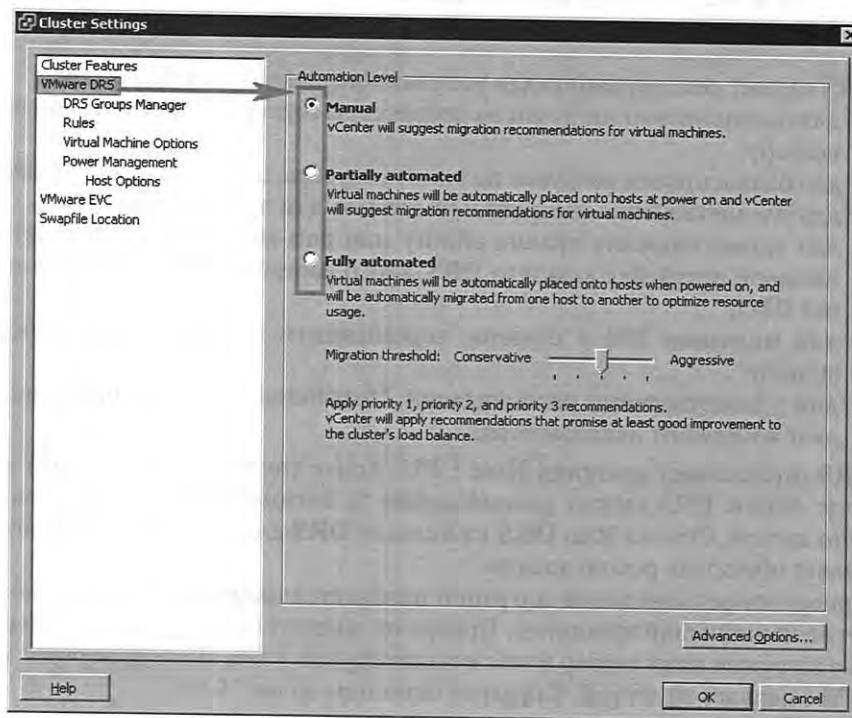


Рис. 6.60. Настройки кластера DRS

По порядку.

VMware DRS. Здесь мы можем указать базовую настройку – уровень автоматизации. Напомню, что DRS делает всего две вещи – инициирует vMotion и вы-

бирает, где включить ВМ. Эти настройки указывают, что из этого следует делать автоматически, а что лишь предлагать для одобрения администратору. Вариантов три:

- ☐ **Manual** (ручной) – и выбор, где включить, и vMotion будет лишь предлагаться;
- ☐ **Partially automated** (полуавтомат) – выбор, где включить ВМ, DRS будет делать сам, а вот vMotion – лишь предлагать;
- ☐ **Fully automated** (полностью автоматический) – и выбор, где включить, и vMotion DRS будет делать сам.

Обратите внимание на бегунок **Migration Threshold** – его положение указывает агрессивность работы кластера DRS. Чем его положение ближе к **Conservative**, тем только более важные рекомендации будет выдавать и исполнять DRS. Чем бегунок ближе к **Aggressive**, тем менее важные рекомендации будут выдаваться.

Обратите внимание. Если DRS работает в ручном (Manual) режиме, то непривилегированные пользователи (у которых есть право включать ВМ, например с ролью Virtual Machine User) не смогут увидеть окно выбора сервера при включении этой ВМ (рис. 6.66), и ВМ не включится.

DRS создает рекомендацию для миграции по следующим причинам:

- ☐ для балансировки нагрузки на процессоры сервера, или reservation по процессору;
- ☐ для балансировки нагрузки на память сервера, или reservation по памяти;
- ☐ для удовлетворения настройки reservation для пулов ресурсов;
- ☐ для удовлетворения правил affinity или anti-affinity (речь идет об одноименной настройке кластера DRS, а не о настройке CPU affinity в свойствах ВМ);
- ☐ для миграции ВМ с сервера, переходящего в режим maintenance или standby;
- ☐ для удовлетворения рекомендаций Distributed Power Management, если этот компонент используется.

DRS отслеживает счетчики Host CPU: Active (включая run и ready) и Host Memory: Active. DRS выдает рекомендации за пятиминутные интервалы, изменить это нельзя. Ссылка **Run DRS** на вкладке **DRS** для кластера принудительно заставляет обсчитать рекомендации.

Приоритет рекомендаций миграции измеряется цифрами от 1 до 5, где 5 указывает на наименьший приоритет. Приоритет зависит от нагрузки на сервера: чем больше загружен один сервер и чем меньше другой – тем выше приоритет миграции ВМ с первого на второй. Обратите внимание на рис. 6.61.

Самое главное здесь – это:

- ☐ **Migration Threshold** – настройка, указывающая на уровень приоритета, рекомендации с которым выполняются автоматически. От этой настройки зависит значение «Target host load standard deviation». Таким образом, данная настройка указывает на то, насколько кластер может быть несбалансированным;

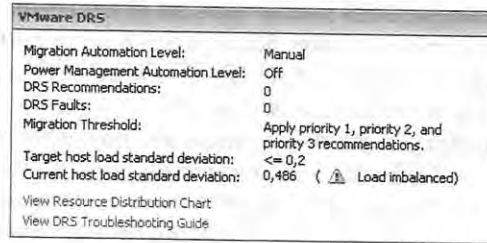


Рис. 6.61. Текущая информация о статусе DRS-кластера

- ❑ **Target host load standard deviation** – стандартное отклонение нагрузки, при достижении которого требуется балансировка;
- ❑ **Current host load standard deviation** – стандартное отклонение текущей нагрузки на сервера.

В расчете стандартного отклонения используется расчет нагрузки на каждый сервер по следующей формуле:

$$\frac{\text{sum(нагрузка от VM на одном сервере)}}{(\text{ресурсы сервера})}$$

Приоритет миграции высчитывается по формуле

$$6 - \text{ceil}(\text{Current host load standard deviation} / 0.1 \times \sqrt{\text{Количество серверов в кластере}}),$$

где $\text{ceil}(x)$ – наименьшее целое, не меньшее x .

Впрочем, формула может меняться в следующих версиях vSphere.

Также высокий приоритет имеют миграции, вызванные правилами affinity и anti-affinity, о которых позже. Наконец, если сервер поместить в режим Maintenance, то DRS сгенерирует рекомендацию мигрировать с него все VM, и у этих рекомендаций приоритет будет максимальный.

Таким образом, если бегунок стоит в самом консервативном положении, выполняются только рекомендации от правил affinity и anti-affinity и миграция с серверов в режиме обслуживания. Если в самом агрессивном режиме – даже небольшая разница в нагрузке на сервера будет выравниваться. Рекомендуемыми настройками являются средняя или более консервативные.

Чтобы выбрать VM, которую или которые лучше всего мигрировать, DRS просчитывает миграцию каждой VM с наиболее загруженного сервера на каждый из менее загруженных серверов, и высчитывается стандартное отклонение после предполагаемой миграции. Затем выбирается наилучший вариант. Притом учитывается «анализ рисков» – стабильность нагрузки на VM за последнее время.

Начиная с версии 4.1, DRS учитывает не только нагрузку на процессоры и память серверов, но и число виртуальных машин. Теперь не будет ситуаций, когда на одном из двух серверов кластера 20 маленьких виртуальных машин, а на втором – 2 большие. Получалось слишком много яиц в первой корзине.

Для версии 5 в кластере DRS может быть до 32 серверов и до 3000 виртуальных машин. На каждом сервере может быть до 512 виртуальных машин. Эти цифры не зависят от типа кластера (HA/DRS/оба) и не зависят от числа серверов в кластере (как это было в предыдущих версиях).

DRS Groups Manager – этот пункт настроек появился только в версии 4.1. Его суть в том, что для DRS-кластера мы можем обозначить группы серверов и виртуальных машин, а затем создать правила соотношения между ними. Правила вида «группа виртуальных машин "test" не должна работать на серверах группы "new_servers"» или «группа виртуальных машин "CPU Intensive" должна работать на серверах группы "Servers_with_new_CPU"» (рис. 6.62).

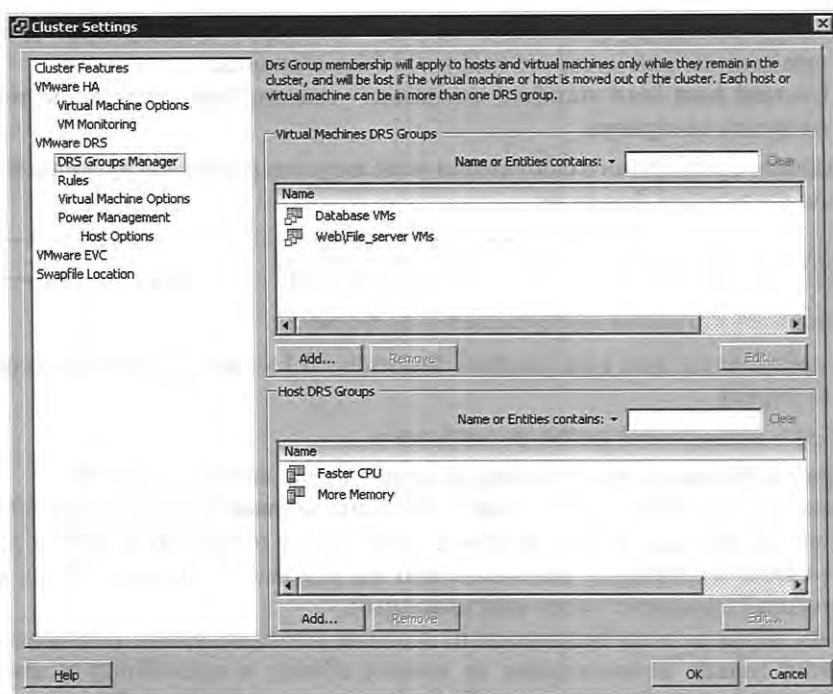


Рис. 6.62. Группы серверов и виртуальных машин для кластера DRS

Эта возможность может быть интересна нам из следующих соображений:

- ☐ чтобы однотипные виртуальные машины работали на одних и тех же серверах. В таком случае механизм transparent page sharing будет более эффективно дедуплицировать оперативную память этих виртуальных машин;
- ☐ чтобы виртуальные машины с более высокими требованиями к производительности работали на более производительных серверах кластера (где более новые процессоры или больший объем памяти);
- ☐ если сервера в кластере разной конфигурации, то самые тяжелые ВМ (от 8 vCPU) лучше не мигрировать на сервера другой конфигурации, чем на

которых такие ВМ включились. Это связано с появлением vNUMA и невозможностью работы этой функции при миграции большой ВМ на сервер другой конфигурации;

- ❑ если приложение в виртуальных машинах лицензируется таким образом, что лицензирование зависит от числа серверов, на которых может заработать виртуальная машина. Ограничив число таких серверов, мы можем удешевить лицензирование или гарантировать соблюдение текущей лицензии;
- ❑ если у части серверов есть единая точка отказа (например, в кластере сервера из двух шасси блейд-серверов), то можно дублирующие друг друга ВМ (такие как сервера DNS или контроллеры домена) явно привязать к серверам только одного шасси. Соответственно, часть серверов DNS старается работать на серверах одного шасси, а часть – на серверах другого шасси;
- ❑ чтобы HA использовал для включения после сбоя некоторых, например тестовых, виртуальных машин только некоторые сервера ESXi.

Rules – здесь мы можем создавать так называемые правила affinity и anti-affinity, а также указывать правила принадлежности групп виртуальных машин группам серверов.

При создании правила anti-affinity мы указываем несколько виртуальных машин, которые DRS должен разносить по разным серверам. Обратите внимание: каждая ВМ одной группы anti-affinity должна располагаться на отдельном сервере. Например, такое правило имеет смысл для основного и резервного серверов DNS или контроллеров домена. Таким образом, в одном правиле не сможет участвовать виртуальных машин больше, чем в кластере серверов.

В правиле affinity мы указываем произвольное количество ВМ, которые DRS должен держать на одном сервере. Например, это бывает оправдано для сервера приложений и сервера БД, с которым тот работает. Если такая пара будет работать на одном сервере, трафик между ними не будет нагружать физическую сеть и не будет ограничен ее пропускной способностью. Если какие-то правила взаимоисключающие, то у имени, созданного последним, нельзя поставить флажок (то есть правило нельзя активировать). С каким другим правилом оно конфликтует, можно посмотреть по кнопке **Details** (рис. 6.63).

Если какое-то правило в данный момент нарушено, получить информацию об этом можно по кнопке **Faults** на вкладке **DRS** для кластера.

Обратите внимание. Доступна возможность создать alarm, который будет отслеживать событие конфликта с правилом. Для этого создайте alarm для виртуальной машины или их группы типа Event based и на вкладке **Trigger** выберите **VM is violating VM-Host Affinity Rule**.

В версии 4.1 появилась возможность разделить виртуальные машины и сервера по группам и указать правила их связи друг с другом. Доступны следующие варианты (рис. 6.64):

- ❑ **Must run on hosts in group** – указанная группа виртуальных машин обязана работать на указанной группе серверов. Если подходящего сервера не будет доступно – виртуальная машина не будет включена или мигрирована;

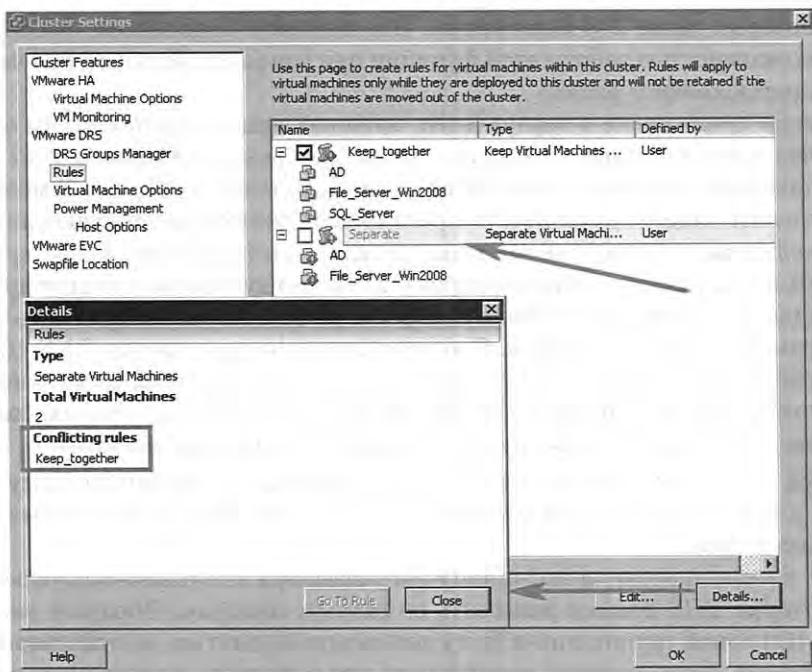


Рис. 6.63. Подробности правила для DRS

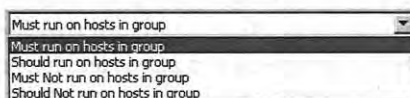


Рис. 6.64. Варианты привязки групп виртуальных машин к группам серверов ESXi

- ☐ **Should run on hosts in group** – указанная группа виртуальных машин должна стараться работать на указанной группе серверов. Если подходящих серверов не будет – виртуальная машина будет работать на каком-то из прочих;
- ☐ **Must Not run on hosts in group** – указанная группа виртуальных машин обязана работать не на указанной группе серверов;
- ☐ **Should Not run on hosts in group** – указанная группа виртуальных машин должна стараться работать не на указанной группе серверов.

Хотя это разделение делается в настройках DRS, «жесткие» правила оказывают влияние также на HA и DPM.

Для этого механизма следует упомянуть о следующих правилах работы:

- ☐ для правил нет возможности указывать приоритет. Если одна виртуальная машина подпадает сразу под пару правил, она будет работать только на серверах, удовлетворяющих обоим правилам;

- ☐ если происходит конфликт правил для одной виртуальной машины, приоритет имеет правило, созданное позднее;
- ☐ если в кластере присутствуют группы с жесткими связями вида «Must run..» или «Must Not run..», то даже администратор не сможет запустить или мигрировать виртуальную машину на сервер, не удовлетворяющий правилу. Более того, DRS, DPM и HA не будут осуществлять операции, противоречащие этим правилам. То есть при конфликте с таким правилом:
 - DRS не будет мигрировать виртуальные машины с сервера, переводимого в режим обслуживания;
 - DRS не будет включать или мигрировать для балансировки нагрузки виртуальную машину на неподходящий сервер;
 - HA не включит виртуальную машину, если подходящих по правилу серверов не будет доступно;
 - DPM не выключит серверов, которые не сможет освободить из-за влияния правила;
- ☐ с правилами типа «Should..», мягкими, таких ограничений нет – именно потому они являются рекомендуемыми в общем случае;
- ☐ правила типа «Should..», мягкие, DRS может не брать в расчет при дисбалансе нагрузки на процессоры или память серверов кластера. Они являются рекомендуемыми, а не обязательными. DRS даже не будет мигрировать виртуальную машину для удовлетворения мягкого правила, а лишь выдаст предупреждение о том, что правило не выполняется.

Обратите внимание. Правила «must..» будут выполняться, даже если функция DRS выключена(!). Возможна ситуация, когда DRS был временно включен, правила созданы, затем DRS выключен. К примеру, включен он был после установки vSphere и ее работы с пробной лицензией, а выключен после установки приобретенной лицензии без функции DRS. Но созданные правила будут оказывать влияние на возможность ручного перемещения VM и работу HA.

Virtual Machine Options – это следующий пункт настроек DRS. Здесь можно включить или выключить возможность индивидуальной настройки уровня автоматизации для каждой VM. Часто имеет смысл для тяжелых VM выставить эту настройку в ручной или полуавтоматический режим, чтобы минимизировать влияние на такие VM со стороны DRS. Балансировка нагрузки будет осуществляться за счет менее критичных виртуальных машин.

Обратите внимание. В случае DRS-кластера рекомендуется для VM с vCenter настроить DRS в ручной режим и всегда держать эту VM на одном выбранном сервере (например, первом). Иначе если VM с vCenter не работает, то искать ее придется на всех серверах по очереди перебором. Или настроить правило, привязывающее vCenter к первому или первым двум серверам.

Power Management и Host Options – здесь настраивается DPM. О нем и о его настройках позднее.

VMware EVC или Enhanced vMotion Compatibility – так как DRS использует vMotion для выполнения своих функций, то для виртуальных машин и серверов

в DRS-кластере должны выполняться условия для vMotion. Одно из них – одинаковость процессоров по поддерживаемым инструкциям. Самый эффективный способ это сделать – включить функцию EVC.

Суть ее – в том, что все сервера в DRS-кластере со включенным EVC «приводятся к единому знаменателю» по функциям процессора, путем отключения тех функций, которых нет хотя бы на одном сервере кластера.

EVC способна обеспечить совместимость не любых процессоров и даже не любых поколений процессоров даже одного вендора. Информацию по группам совместимости EVC (рис. 6.65) можно найти в базе знаний VMware (статьи 1003212 и 1005764).

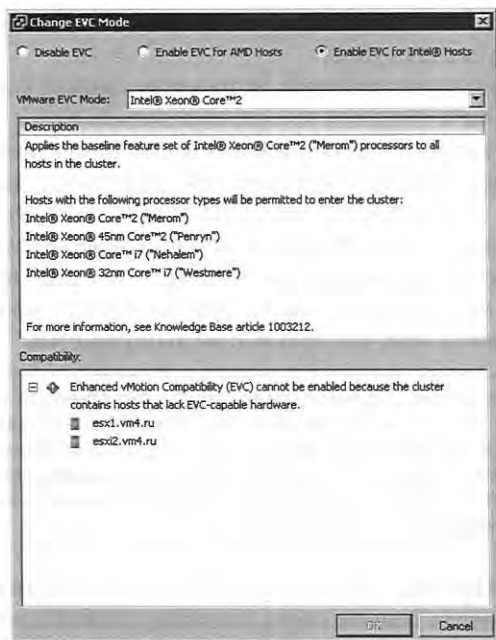


Рис. 6.65. Настройка EVC для кластера DRS

Обратите внимание: включение EVC может потребовать выключения виртуальных машин. В идеале включать EVC следует в момент внедрения виртуальной инфраструктуры, придерживаясь следующей последовательности шагов:

1. Создать кластер DRS (на нем также может быть включена и функция HA, это не имеет значения в данном контексте).
2. Включить EVC.
3. Добавить в кластер сервера.
4. После этого начать создавать и включать виртуальные машины.

Если DRS-кластер уже существует, то порядок включения EVC будет следующим:

1. На тех серверах в кластере, что имеют больше функций, чем в используемом шаблоне EVC, не должно остаться ВМ. Их нужно выключить.
2. Включить EVC.
3. Включить ВМ, выключенные на шаге 1.

Swapfile location – где ВМ по умолчанию будут хранить свой `vmkernel swap`-файл. Обратите внимание, что файл подкачки может быть расположен не на общем хранилище. Например, если мы указали хранить файлы подкачки на локальных дисках серверов, виртуальные машины все равно смогут мигрировать с помощью `vMotion`.

Обращаю ваше внимание, что употребляемый здесь термин «кластер» означает тип объектов в иерархии `vCenter`. По сути, этот «кластер» – не что иное, как контейнер для серверов, группа серверов. На этой группе мы можем включить функции DRS и/или HA. Однако настройки EVC и `Swapfile location` мы можем задавать и для кластера, где ни DRS, ни HA не включены. Следовательно, мы можем задавать эти настройки, даже когда лицензий на HA и DRS у нас нет.

Иллюстрация работы DRS – обратите внимание на рис. 6.66.

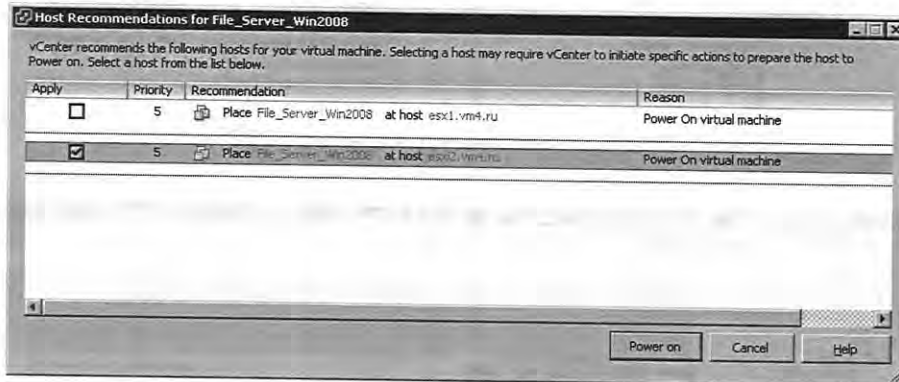


Рис. 6.66. DRS предлагает сервер, на котором лучше включить ВМ

Это окно выбора сервера для включения ВМ. Такое окно вы увидите в `Manual`-режиме DRS-кластера при включении ВМ. Обратите внимание, что при клике на имена ВМ и серверов вы попадете в их свойства, а не выберете миграцию на тот или иной сервер. Смотреть здесь надо на значения в столбце **Priority** – чем больше число, тем лучше (по мнению DRS) запустить ВМ на сервере из этой строки.

Выделив кластер, на вкладке **Summary** вы увидите базовую информацию по нему. Обратите внимание на **DRS Recommendations** (рис. 6.67).

Перейдя по этой ссылке, вы попадете на вкладку **DRS**. По кнопке **Recommendations** на ней вы увидите примерно такую картинку (рис. 6.68).

Вот так выглядят рекомендации к миграции от DRS. В этом примере нам предлагают мигрировать ВМ `File_Server_Win2008` на сервер `esx1.vm4.ru`. Если вы хотите выполнить эту рекомендацию, нажмите кнопку **Apply Recommendation**. Если DRS предлагает миграцию сразу нескольких ВМ, а вы не хотите мигрировать сразу все – поставьте флажок **Override DRS recommendations** и флажками в столбце **Apply** выбирайте рекомендации, которые будут выполнены по нажатию **Apply Recommendation**.

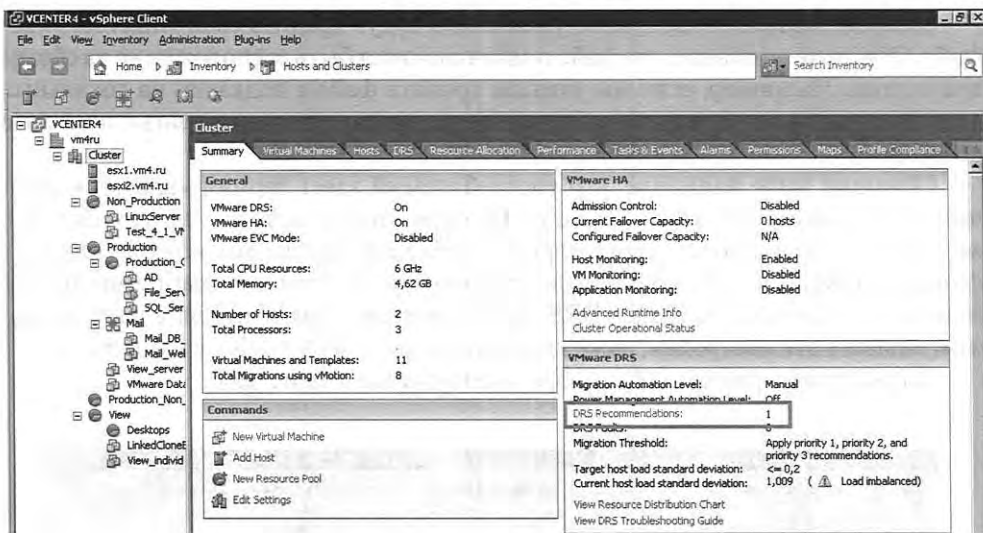


Рис. 6.67. Summary для кластера DRS

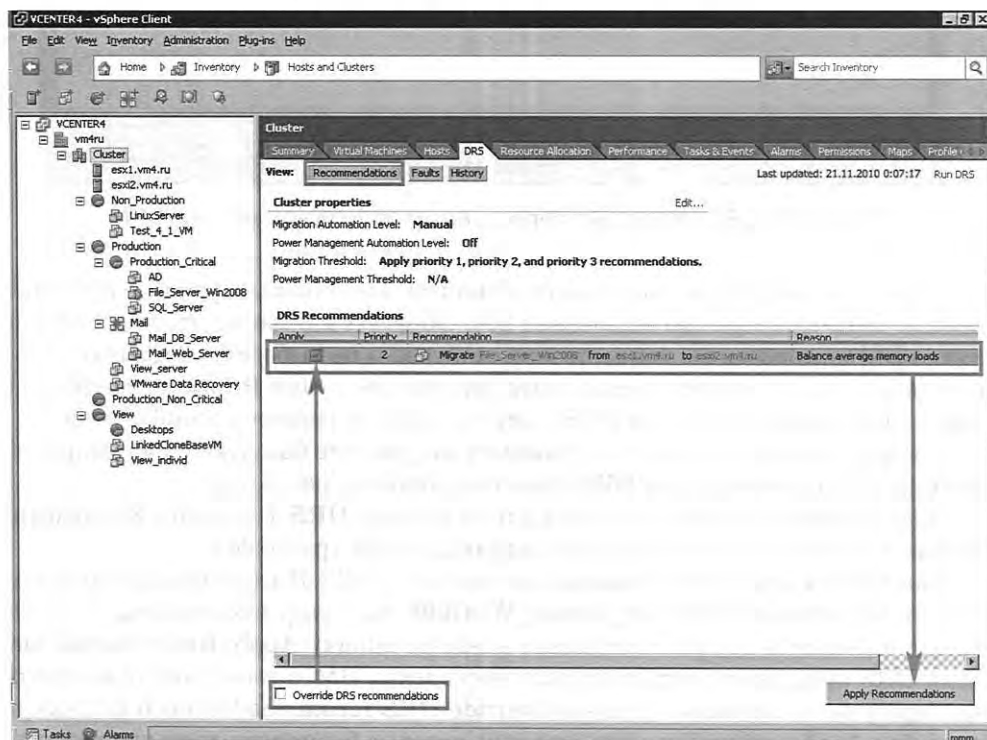


Рис. 6.68. Рекомендация vMotion от DRS

По кнопке **History** на вкладке **DRS** для кластера доступна история успешных миграций. Данные на этой странице сохраняются в течение четырех часов и доступны и при открытии новой сессии клиента vSphere.

По кнопке **Faults** на вкладке **DRS** для кластера доступна история неуспешных миграций. Показывается, кто и куда не может (для режима Manual) или не смог (для Automatic) мигрировать и по какой причине.

Обычно рекомендации DRS следуют из сильно различающейся нагрузки на сервера, которую можно увидеть на вкладке **Hosts** для кластера (рис. 6.69).

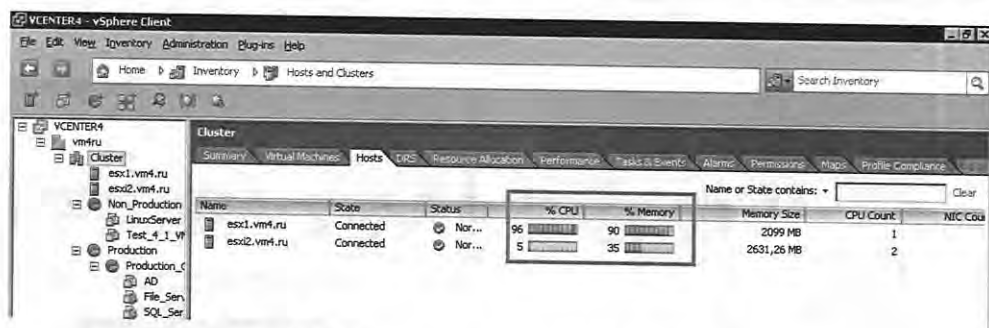


Рис. 6.69. Нагрузка на сервера в DRS-кластере

Вернемся к рис. 6.54. На вкладке **Summary** для кластера DRS доступна информация о его настройках и о ресурсах кластера. В моем примере сообщается, что совокупные ресурсы кластера равняются 6 ГГц для процессора и 4,62 Гб для памяти. Обратите внимание: это не означает, что я могу выделить одной VM 4,62 Гб физической памяти, потому что эта память разнесена по нескольким серверам. То есть правильно читать эту информацию следующим образом: все VM в данном кластере в совокупности могут задействовать 6 ГГц и 4,62 Гб физической памяти. То есть для распределения ресурсов присутствует некоторая дискретность. Впрочем, обычно ресурсы сервера значительно превосходят ресурсы для самой требовательной VM, и это предупреждение неактуально.

DMP, Distributed Power Management

Distributed Power Management (DPM) – это дополнительная функция кластера DRS. Заключается она в том, что DRS анализирует нагрузку на сервера и, если для работы всех виртуальных машин достаточно лишь части серверов, может перенести все VM на эту часть. А оставшиеся сервера перевести в standby-режим. Разумеется, в тот момент, когда в их производительности возникнет нужда, DPM включит их обратно. Анализ DPM основан на тех данных, что собирает и анализирует DRS. DRS выполняет свою работу с пятиминутным интервалом, таким образом, DRS пересчитывает свою аналитику раз в пять минут. При анализе DPM руководствуется данными за 40-минутный интервал.

По умолчанию DPM считает нормальной нагрузку на сервер $63 \pm 18\%$. Когда нагрузка на сервера превышает 81%, начинается включение серверов. Когда нагрузка

падает ниже 45%, DPM начинает консолидировать ВМ и выключать сервера. Эти настройки можно изменять в **Advanced Settings** для DRS-кластера, о чем ниже.

Таким образом, если наша инфраструктура имеет запас по серверам, то лишние в данный момент сервера могут быть выключены. Или когда нагрузка по ночам значительно падает, опять же часть серверов может не греть воздух впустую.

Для включения серверов DPM может использовать два механизма: BMC/IPMI или Wake-On-LAN (WOL). Для задействования BMC/IPMI необходимо для каждого сервера указать необходимые параметры доступа к BMC (Baseboard Management Controller). Под BMC понимаются контроллеры, работающие по протоколу IPMI, например HP iLO, Fujitsu iRMC, IBM RSA, Dell DRAC (рис. 6.70).

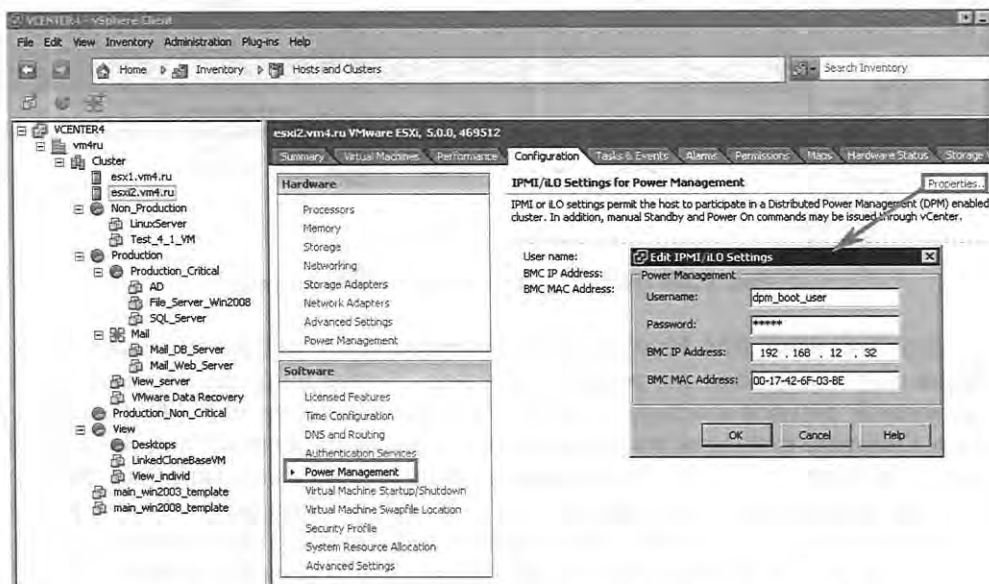


Рис. 6.70. Настройка параметров доступа к IPMI/iLo для ESXi 5

Само собой, должны быть сделаны необходимые настройки – на BMC должен быть создан пользователь, имеющий право включать сервер, IP-адрес BMC/IPMI должен быть доступен с сервера vCenter. Подробности по настройке этих компонентов следует искать в документации к серверу.

Суть BMC/IPMI-контроллера – в том, что эти контроллеры являются, по сути, компьютерами в миниатюре, со своим сетевым портом. Эти контроллеры имеют доступ к управлению оборудованием «большого» сервера. Работают они независимо от работоспособности сервера. Таким образом, если сервер выключен, то BMC/IPMI остается доступен по сети, и по сети можно инициировать, в частности, функцию включения сервера. Этим и пользуется DPM.

Если для сервера не указаны настройки BMC/IPMI, то vCenter будет пытаться включать их с помощью механизма Wake-On-Lan. Обратите внимание, что

пакеты WOL будут посылаться на те физические сетевые контроллеры сервера, к которым подключен vMotion-интерфейс VMkernel. Посылаться они будут не сервером vCenter, а одним из включенных серверов ESXi по команде vCenter. Таким образом, необходимо выполнение следующих условий:

- ☐ на каждом сервере должны быть интерфейсы VMkernel с разрешенным vMotion через них;
- ☐ эти интерфейсы должны быть в одной подсети, без маршрутизаторов между ними;
- ☐ тот физический сетевой контроллер, через который будет работать этот vMotion-интерфейс, должен поддерживать WOL. Проверить это можно, пройдя **Configuration** $\Rightarrow$ **Network Adapters** для сервера. В столбце **Wake On LAN Supported** должно стоять **Yes**.

Обратите внимание, что многие сетевые контроллеры поддерживают WOL не во всех режимах работы. Часто только 100 или 10 Мбит/с. Таким образом, может потребоваться настройка портов физического коммутатора, к которому подключены эти сетевые контроллеры серверов. Эти порты должны быть настроены на автосогласование (auto negotiate) скорости, чтобы, когда сервер выключен, сетевые карты могли работать на требуемой для WOL скорости.

Еще до включения функционала DPM следует протестировать работоспособность этих механизмов. В контекстном меню работающего сервера есть пункт **Enter Standby Mode**, а в меню выключенного сервера – **Power on**. Если включение сервера через IPMI/iLO не заработало, то удалите настройки из пункта **Power Management** для этого сервера. Когда IPMI/iLO для сервера не настроено, DPM будет использовать WOL.

Для настройки DPM зайдите в свойства кластера DRS, вас интересует пункт **Power Management**. Там три варианта настройки:

- ☐ **Off** – DPM выключен;
- ☐ **Manual** – DPM включен, но будет лишь показывать рекомендации по выключению серверов. Эти рекомендации будут доступны на вкладке **DRS** для кластера;
- ☐ **Automatic** – включение и выключение серверов, а также связанные с этим миграции ВМ будут происходить автоматически. Бегунок указывает на то, рекомендации с каким приоритетом будут выполняться автоматически. Приоритет обозначается цифрами от 1 (максимальный) до 5 (минимальный).

Обратите внимание, что уровень автоматизации можно указывать индивидуально для каждого сервера. Делается это на пункте **Host Options**. Если какие-то сервера невозможно включить через IPMI/iLO/WOL, будет хорошей идеей для них функцию DPM отключить.

Рекомендации DRS и DPM не зависят друг от друга. Уровни автоматизации DRS и DPM не связаны друг с другом.

Для автоматизации мониторинга действий как DRS, так и DPM можно использовать механизм vCenter alarms. Для кластера или выше него в иерархии

vCenter надо создать alarm, который мониторит события (events). Например, для DPM можно мониторить следующие события:

- ☐ **DrsEnteringStandbyModeEvent** – инициация выключения сервера;
- ☐ **DrsEnteredStandbyModeEvent** – успешное выключение сервера;
- ☐ **DrsExitingStandbyModeEvent** – инициация включения сервера;
- ☐ **DrsExitedStandbyModeEvent** – успешное включение сервера;
- ☐ **DrsExitStandbyModeFailedEvent** – неуспешное включение сервера.

С мониторингом может быть связан еще один нюанс – у вас может использоваться какое-либо из средств мониторинга, такое как BMC Performance Manager, HP System Insight Manager, CA Virtual Performance Management, Microsoft System Center Operations Manager, IBM Tivoli. Эта система может среагировать на выключение сервера, инициированное DPM. Могут потребоваться коррективы правил мониторинга, чтобы на такие штатные выключения серверов эти системы не реагировали.

Достаточно естественной является идея включать сервера по утрам, немного ДО того, как их начнут нагружать приходящие на работу сотрудники. Начиная с версии 4.1, эта возможность реализуется штатными средствами. Для этого через планировщик vCenter (**Home** ⇒ **Scheduled Tasks** ⇒ **New**) создайте задание **Change Cluster Power Settings**.

Обратите внимание. vCenter привязывает шаблоны ВМ к какому-то серверу, даже когда они расположены на общем хранилище. Если DPM выключит тот сервер, за которым числится шаблон, то шаблоном невозможно будет воспользоваться. Поэтому в идеале тот сервер, за которым числятся шаблоны, не должен выключаться DPM. Если же шаблон оказался недоступен, то его надо удалить из иерархии vCenter (пункт **Remove from Inventory** контекстного меню шаблона) и затем добавить обратно через встроенный файловый менеджер (пункт **Browse Datastore** контекстного меню хранилища, где расположен шаблон).

Удаление DRS

С удалением DRS-кластера связан следующий нюанс – удаление DRS уберет все пулы ресурсов, существующие в нем. Кроме того, пропадут все правила affinity, anti-affinity и правила привязки групп виртуальных машин к группам хостов. Поэтому, если вам необходимо лишь на время отключить функционал DRS, часто лучше перевести его в режим **Manual** и проверить, что для каждой ВМ также используется **Manual**-режим.

Advanced Settings

Обратите внимание на рис. 6.71.

Здесь вы видите область расширенных настроек кластера DRS и DPM.

В данном примере значение 75 настройки DemandCapacityRatioTarget говорит о том, что DPM должен обращать внимание на сервера не по формуле $63 \pm 18\%$, а по формуле $75 \pm 18\%$.

За списком расширенных настроек обратитесь по ссылке <http://www.vmware.com/resources/techresources/1080>.

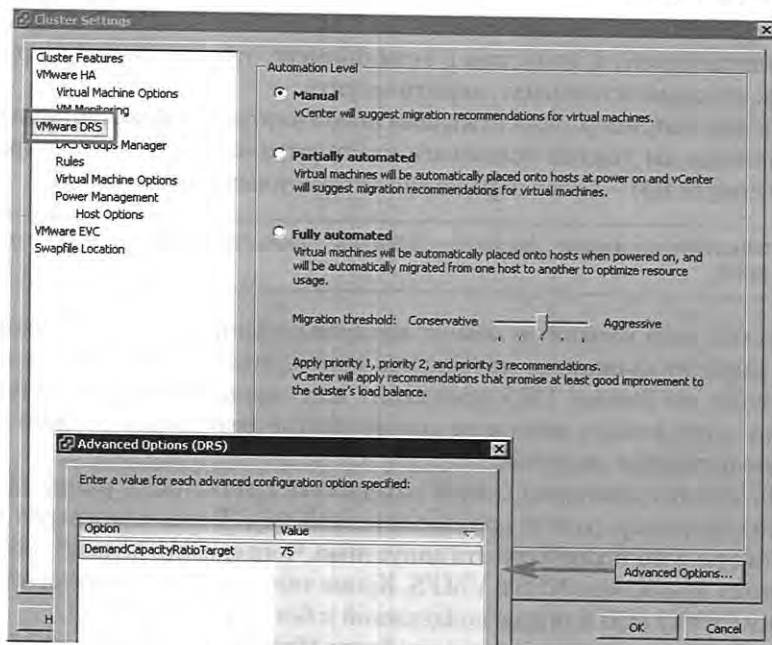


Рис. 6.71. DRS Advanced Settings

6.9. Кластер Storage DRS

В пятой версии vSphere появилась возможность создавать группы хранилищ – кластеры Storage DRS.

Объединив в такую группу несколько хранилищ (это имеет смысл для хранилищ с одинаковыми характеристиками), мы получаем оптимизацию для таких операций, как создание новой VM, развертывание из шаблона, миграции VM, то есть операций, для которых нам следует указывать хранилище для размещения на нем VM.

В чем неудобство – иногда у нас довольно много хранилищ, иногда многие из них одинаково годятся под новую VM, но где-то больше свободного места – где-то меньше. Где-то нагрузка выше – где-то ниже. А выбирать надо нам. Хорошо, если «мы» – это системный администратор, который этот выбор сможет сделать осмысленно. А если за такие операции отвечают люди не с такой высокой квалификацией?

Теперь мы можем выбрать сразу группу хранилищ – а конкретное будет выбрано автоматически, в соответствии с критериями. Критериев два:

- ☐ процент свободного места;
- ☐ нагрузка на хранилище.

По логике вещей, если мы планируем использовать SDRS, то нам следует объединить в кластеры все имеющиеся у нас хранилища. Допустим, у нас были хранилища с разными характеристиками по производительности, доступности и ско-

рости для тестовых ВМ, производственных ВМ, критичных ВМ. А теперь будет не россыпь «датасторов», а, допустим в этом примере, три кластера. Как раз – «для тестовых», «производственных», «критических».

Теперь при, например, развертывании новой виртуальной машины на шаге выбора хранилища мы увидим только эти кластеры, а на каком именно хранилище будет развернута ВМ – будет определено автоматически. Это удобно.

Обратите внимание. Диски одной ВМ могут быть расположены на разных кластерах SDRS.

Более того, если в какой-то момент времени ситуация стала несбалансированной – на какое-то хранилище из кластера приходится заметно больше нагрузки, чем на другие, то Storage DRS предложит или выполнит миграцию одной или нескольких виртуальных машин на другое/другие хранилища данного кластера, с целью балансировки нагрузки.

В один кластер хранилищ можно объединять хранилища с разными характеристиками (например, разной производительности). Лично я не уверен в целесообразности этого, но технически это допустимо. Чего сделать нельзя – объединить в одну группу хранилища NFS и VMFS. Кроме того, явно не рекомендуется включать в одну группу хранилища с поддержкой и без поддержки VAAI.

Для создания группы хранилищ пройдите **Home** ⇒ **Datastores and Datastore Clusters** ⇒ в контекстном меню датацентра выберите **New Datastore Cluster**. Запустится мастер:

- ☐ **General** – здесь мы укажем имя создаваемого хранилища и выберем, следует ли сразу активировать Storage DRS для этого кластера флажком **Turn on Storage DRS**;
- ☐ **SDRS Automation** – уровень автоматизации. В ручном режиме система будет лишь предлагать хранилища со своими рекомендациями, в режиме **Fully Automated** этот выбор будет осуществляться автоматически;
- ☐ **SDRS Runtime Rules** – указание критериев, по которым будет осуществляться работа создаваемого кластера:
 - флажок **I/O Metric Inclusion** позволяет учитывать нагрузку на хранилища, не только процент свободного места. Обратите внимание на то, что установка этого флажка включает Storage IO Control на хранилищах данного кластера;
 - **Storage DRS Thresholds** – пороговые значения утилизации и latency, по достижении которых генерируются рекомендации или непосредственно осуществляется балансировка между хранилищами кластера;
 - **Advanced Options** – здесь мы указываем процент разницы между показателями для разных хранилищ, начиная с которого будет реакция SDRS, а также период времени, за которое оценивается несбалансированность;
- ☐ **Select Hosts and Clusters** – здесь мы выбираем сервера ESXi. В группу хранилищ желательно объединять только те хранилища, которые доступны сразу группе хостов. Мы выберем хосты этом этапе, и затем нам подскажут, какие из хранилищ доступны сразу всем из выбранных. Обычно какие-то

хранилища доступны все серверам одного кластера и недоступны другим серверам, так что чаще всего на этом этапе будет выбран кластер хостов;

- ❑ **Select Datastores** – а на этом этапе мы непосредственно и выбираем хранилища, объединяемые в кластер хранилищ.

Однако некоторые настройки недоступны при создании кластера SDRS, а доступны только в настройках существующего. Это настройки:

- ❑ **SDRS Scheduling** – здесь мы можем указать расписание смены режимов работы кластера, автоматический/ручной. Самый характерный пример – мы бы не хотели автоматического запуска Storage vMotion в разгар рабочего дня, когда эта лишняя нагрузка на систему хранения не к месту. А вне рабочих часов будней – пожалуйста;
- ❑ **Rules** – правила «дружбы/не дружбы» виртуальных машин или дисков одной ВМ. Если у нас есть причины держать на разных хранилищах одного кластера SDRS несколько ВМ или несколько дисков одной ВМ – именно здесь это желание можно воплотить в виде настройки для SDRS. По умолчанию SDRS стремится располагать все файлы каждой одной ВМ на одном хранилище. Единственное исключение – файлы vswp, файлы подкачки, их SDRS не мигрирует;
- ❑ **Virtual Machine settings** – здесь можно указать уровень автоматизации индивидуально для каждой ВМ и, при необходимости, отключить существующее по умолчанию правило сохранять все диски одной ВМ на одном хранилище.

Обратите внимание. Если при создании ВМ сложилась ситуация, что в кластере SDRS свободного места достаточно, но его недостаточно ни на одном отдельно взятом хранилище, то SDRS предложит мигрировать несколько уже существующих ВМ – с целью освободить достаточно места на одном из хранилищ.

Если кластер SDRS используется в автоматическом режиме, то рекомендации по миграции ВМ между хранилищами будут выполнены автоматически. В ручном режиме мы сможем обнаружить эти рекомендации на вкладке **Storage DRS** для кластера. Кнопка **Apply Recommendation** позволит нам применить рекомендации.

На этой вкладке нам доступны несколько кнопок.

Кнопка **Faults** отобразит информацию о конфликтах – например, если на одну и ту же ВМ влияют несколько взаимоисключающих правил.

Кнопка **History** отображает историю действий кластера SDRS.

Ссылка **Run Storage DRS** запустит обсчет текущей ситуации в кластере SDRS.

После добавления хранилища в кластер SDRS в контекстном меню каждого хранилища появляется пункт **Enter SDRS Maintenance Mode**. Он пригодится в ситуации, когда вам необходимо эвакуировать все ВМ с какого-то хранилища. Например, чтобы вывести его из эксплуатации, совсем или в этом кластере хранилищ.

Обратите внимание. Крайне желательно не хранить образы iso на хранилищах кластера SDRS. Это может помешать вводу хранилища в режим обслуживания.

В одном кластере SDRS может быть до 32 хранилищ и до 9000 файлов vmdk. Самых кластеров может быть создано до 256.



Глава 7. Защита данных и повышение доступности виртуальных машин

В этой части поднимем такие вопросы, как обеспечение высокой доступности виртуальных машин с помощью решений VMware (HA, FT). Сторонние решения, работающие «изнутри» ВМ (например, кластер Майкрософт с переходом по отказу, Microsoft Failover Cluster), организовать на виртуальной инфраструктуре проблем не составляет, но касаться этой темы здесь мы не будем – см. документацию (ESXi and vCenter Server 5.0 Documentation ⇒ vSphere Resource Management ⇒ Setup for Failover Clustering and Microsoft Cluster Service, <http://pubs.vmware.com>).

Поговорим про подходы к резервному копированию виртуальных машин и решения резервного копирования VMware (Data Recovery и VCB / vStorage API for Data Protection). Коснемся обновления серверов ESXi, гостевых ОС и приложений с помощью VMware Update Manager. И немного поговорим про решение проблем в виртуальной инфраструктуре.

7.1. Высокая доступность виртуальных машин

Обеспечение высокой доступности приложений – одна из важнейших задач ИТ-подразделения любой компании. И требования к доступности становятся все жестче для все большего круга задач.

Обычно под доступностью понимается время, в течение которого приложение было доступно. Если приложение не было доступно 3 с половиной дня за год, то доступность его была порядка 99%. Если приложение было недоступно порядка часа за год, то доступность порядка 99,99% и т. д.

Разные решения высокой доступности обеспечивают разную доступность, имеют разную стоимость, разную сложность реализации (включая разнообразные условия на инфраструктуру). Таким образом, нельзя выделить какое-то решение как идеальное, необходимо выбирать, исходя из условий поставленной задачи и ее требований.

Самую высокую доступность обеспечат решения, встроенные в само приложение, или кластеры уровня приложений между виртуальными машинами (например, Microsoft Failover Cluster). Следом идет VMware Fault Tolerance, обеспечивающий отличную защиту, но лишь от сбоев сервера (не гостевой ОС или приложения). Последним идет VMware High Availability, который дает наибольшие из этих трех вариантов простои, зато чрезвычайно прост, дешев и налагает

минимум условий на инфраструктуру и виртуальные машины. Поговорим про решения высокой доступности от VMware.

7.1.1. VMware High Availability, HA

VMware High Availability, или кластер высокой доступности VMware, появился еще в третьей версии виртуальной инфраструктуры VMware. С момента появления это решение приобрело дополнительные возможности и лишилось некоторых недостатков. Особенно ярко это проявилось именно в пятой версии vSphere, потому что именно для этой версии vSphere программистами VMware функционал кластера HA был переписан с нуля.

Текущая версия VMware HA умеет делать две вещи:

- ❑ проверять доступность серверов ESXi. Для проверки доступности используются специальные сообщения, «сигналы пульса» (heartbeat), отправляемые по сети управления. Используются сообщения пульса собственного формата. В случае недоступности какого-то сервера делается попытка запустить работавшие на нем ВМ на других серверах кластера;
- ❑ проверять доступность виртуальных машин, отслеживая сигналы пульса (heartbeat) от VMware tools. В случае их пропажи ВМ перезагружается. Отвечающий за эту функцию компонент называется Virtual Machine Monitoring. Этот компонент включается отдельно и является необязательным. С недавних пор Virtual Machine Monitoring получил возможность интегрироваться со сторонними решениями, такими как Symantec ApplicationHA, и отслеживать статус не только VMware tools, но и приложений в гостевой ОС.

Для создания HA-кластера необходимо создать объект **Cluster** в иерархии vCenter. В контекстном меню объекта **Datacenter** выберите **Add new Cluster**, укажите имя создаваемого кластера и поставьте флажок **HA** (не имеет значения, стоит ли флажок DRS). Нажмите **OK**.

Итак, кластер как объект vCenter вы создали. Осталось два шага:

1. Настроить HA-кластер.
2. Добавить в него сервера ESXi.

Впрочем, включить HA можно и для уже существующего кластера.

Обратите внимание, что при включении HA для кластера или при добавлении сервера в кластер с включенным HA в панели **Recent Tasks** появляется задача «Configuring HA». Проследите, чтобы эта задача для всех серверов окончилась успешно. Если это не так, ознакомьтесь с сообщением об ошибке и решите проблему.

Условия для HA

На сервера и ВМ в кластере HA налагается следующее условие – ВМ должны иметь возможность запуститься на любом из серверов кластера. То есть:

- ❑ ВМ должны быть расположены на общих хранилищах, доступных всем серверам;

- ☐ ВМ должны быть подключены к группам портов с такими именами, которые есть на всех серверах;
- ☐ к ВМ не должны быть подключены COM/LPT-порты, FDD и CD-ROM серверов, образы iso и flp с приватных ресурсов – то есть все то, что препятствует включению ВМ на другом сервере. Также воспрепятствует перезапуску на другом сервере использование VMDirectPath.

Обратите внимание, что условия HA весьма напоминают условия для VMotion (что логично), но среди них нет условия на совместимость процессоров. Его нет, потому что в случае HA виртуальные машины *перезапускаются* на других серверах, а не мигрируют на горячую. А перезапуститься ВМ может на любом процессоре, в том числе и на процессоре другого производителя.

Также для кластера HA существуют следующие ограничения:

- ☐ серверов в кластере может быть до 32;
- ☐ без условий на количество серверов в кластере число виртуальных машин на каждом сервере может достигать 512;
- ☐ всего в кластере может быть до 3000 виртуальных машин.

Эти ограничения (если эти цифры можно назвать ограничением) общие для кластеров HA и DRS.

Обратите внимание. 512 виртуальных машин на сервер – это абсолютный максимум. Он не может быть превышен не только в нормально работающей инфраструктуре, но и после отказа сервера или нескольких – иначе HA не сможет включить все виртуальные машины с отказавших серверов.

Какие настройки доступны для кластера HA

Настройки VMware HA представлены на рис. 7.1.

Host Monitoring Status – если флажок **Enable Host Monitoring** не стоит, то HA не работает. Снятие его пригодится, если вы планируете какие-то манипуляции с сетью управления, на которые HA может отреагировать, а вам бы этого не хотелось. Альтернатива снятию данного флажка – снятие флажка **VMware HA** для кластера (Cluster Features), но это повлечет за собой удаление агентов HA с серверов и сброс настроек. Это неудобно, если функционал HA нужно восстановить после завершения манипуляций.

Admission Control – это настройка резервирования ресурсов на случай сбоя. Вариант «Allow VMs to be powered on...» отключает какое-либо резервирование, то есть HA-кластер не ограничивает количества запущенных виртуальных машин. Иначе – HA будет самостоятельно резервировать какое-то количество ресурсов. Настройка способа высчитывания необходимого количества ресурсов делается в пункте **Admission Control Policy**.

Вариант без резервирования может быть удобнее, потому что вы гарантированно не столкнетесь с сообщением вида «невозможно включить эту ВМ, так как HA-кластер не разрешает». Минусом является теоретическая возможность того, что в случае отказа сервера (или нескольких) ресурсов оставшихся серверов будет недостаточно для запуска всех ВМ с упавших серверов.

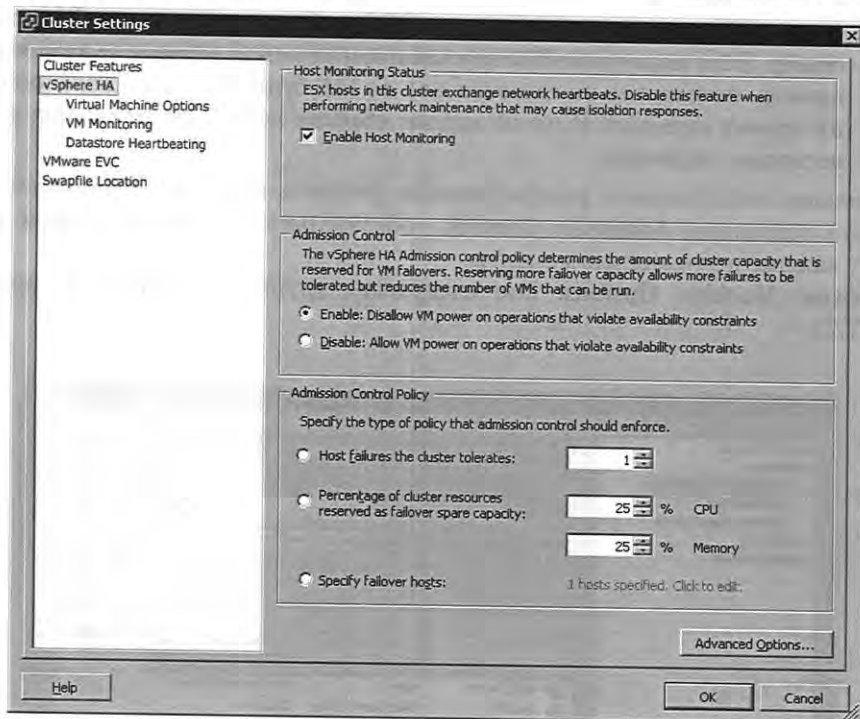


Рис. 7.1. Основные настройки кластера HA

Admission Control Policy – как именно HA должен обеспечивать резервирование ресурсов. Если мы хотим доверить HA-кластеру контроль за этим и выбрали **Admission Control** = «Prevent VMs from being powered on...», то здесь выбираем один из трех вариантов резервирования ресурсов:

- ❑ **Host failures the cluster tolerates** – выход из строя какого числа серверов должен пережить кластер. При выборе этого варианта настройки HA следит за тем, чтобы ресурсов всех серверов минус указанное здесь число хватало бы на работу всех ВМ. В подсчете необходимых для этого ресурсов HA оперирует понятием «слот». О том, что это и как считает ресурсы HA, чуть далее;
- ❑ **Percentage of cluster resources** – в этом случае HA просто резервирует указанную долю ресурсов всех серверов. Для расчетов текущего потребления используются настройки reservation каждой ВМ, или константы 256 Мб для памяти и 256 МГц для процессора, что больше. Кстати, 100% ресурсов в данном случае – это не физические ресурсы сервера, а ресурсы так называемого «root resource pool», то есть все ресурсы сервера минус зарезервированные гипервизором для себя;
- ❑ **Specify a failover hosts** – указанный сервер является «запаской». На этом сервере нельзя будет включать виртуальные машины – только HA сможет включить на нем ВМ с отказавших серверов. На него нельзя будет мигри-

ровать ВМ с помощью VMotion, и DRS не будет мигрировать ВМ на этот сервер. Если включение ВМ на указанном запасном сервере не удалось (например, потому что он сам отказал), то HA будет пытаться включить ВМ на прочих серверах. В пятой версии vSphere можно выделять под резерв несколько серверов.

Больше информации о резервировании ресурсов на случай сбоя будет приведено в подразделе **Admission Control** – обязательно ознакомьтесь с ним, там я опишу важные неочевидные моменты.

Virtual Machine Options. Это следующая группа настроек кластера HA (рис. 7.2).

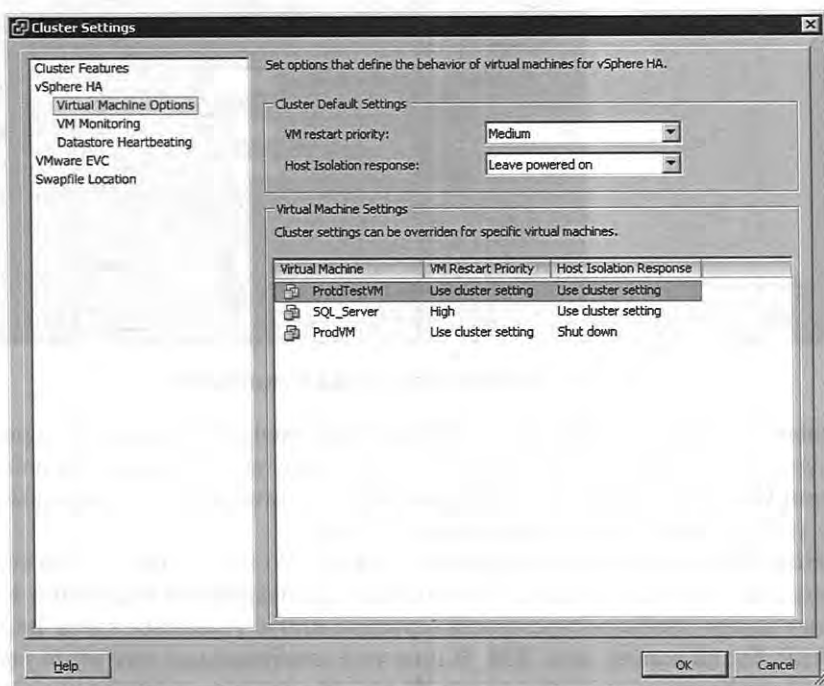


Рис. 7.2. Virtual Machine Options

Здесь нам доступно следующее:

- **VM restart priority** – приоритет рестарта для ВМ-кластера, здесь указываем значение по умолчанию. Очевидно, что первыми должны стартовать ВМ, от которых зависят другие. Например, сервер БД должен стартовать до сервера приложений, использующего данные из БД.

Обратите внимание, что если одновременно отказали хотя бы два сервера, то перезапуск ВМ с них производится последовательно. То есть сначала запустятся все ВМ с одного из отказавших, в порядке приоритета. Затем все ВМ со второго, опять же в порядке приоритета;

- **Host Isolation response** – должен ли сервер выключить свои виртуальные машины, если он оказался в изоляции (isolation). Что такое изоляция, см. чуть далее;
- **Virtual Machine Settings** – здесь можем обе предыдущие настройки указать индивидуально для каждой ВМ.

Admission Control

Когда мы планируем нашу инфраструктуру, мы обычно хотим обеспечить резервирование. На этапе планирования это заключается в том, что мы учитываем запас ресурсов на случай сбоя при сайзинге инфраструктуры. Итак, для простоты примем, что в вашей инфраструктуре достаточно N серверов ESXi для работы ВМ, а куплены были N+1.

Внимание, вопрос: а вы уверены, что N серверов будет достаточно?

Поясню, что я имею в виду: не всегда на этапе планирования у нас есть возможность точно понимать – а сколько ресурсов будут требовать наши ВМ. И потребности отдельных ВМ могут оказаться заниженными, и темпы роста числа пользователей, и, наверное чаще всего, количество ВМ.

Итак, а вы уверены, что N серверов будет достаточно?

Если ответ «да», то в настройках кластера HA пункт **Admission Control** вам не особо интересен. У вас де-факто есть ресурсы на случай отказа сервера/серверов – и отдельно контролировать их нужды нет.

Кстати, частным случаем ситуации, в которой нет нужды контролировать ресурсы принудительно, является ситуация, когда ИТ-отдел имеет возможность своевременно расширять парк серверов. Возросла нагрузка сверх ранее запланированного – мы докупили сервер-другой, и опять ресурсов в достатке, отдельно резервировать под сбой нужды нет, Admission Control можно отключить.

А вот если нет уверенности в том, что ресурсов хватит на все ВМ в случае отказа сервера – вот в этом случае Admission Control нам пригодится. Из-за этой настройки мы просто не сможем включить очередную ВМ, которая первой оказывается «лишней» с точки зрения резервов ресурсов на случай сбоя.

Сделаю акцент – если вы оказались в ситуации, когда все ресурсы серверов, кроме «зачачки на случай сбоя», используются, и вы попытаетесь включить еще одну ВМ, то вы увидите сообщение об ошибке, и она не включится.

Вот в этом суть настройки Admission Control – принудительный контроль «зачачки» на случай сбоя.

Но какого размера «зачачка» должна быть? Как раз за ее расчет и отвечает настройка **Admission Control Policy**.

Начну с конца.

Specify failover host. Этот вариант настройки кажется мне самым хорошим. Его суть – вы просто выбираете один или несколько серверов, которые полностью освобождаются от виртуальных машин и начинают выступать в роли «серверов горячего резерва». Они все время работают, и все время работают свободными от ВМ.

Чем мне нравится этот вариант резервирования ресурсов на случай сбоя: простотой и понятностью. Выбранный сервер/сервера свободен, на остальные ника-

кого влияния не оказывается. Кроме того, хорошая гарантия достаточности «зачачки» – если отказал какой-то из задействованных серверов, то сервер горячего резерва обеспечит то же самое количество ресурсов (предполагая, что все сервера имеют одинаковую конфигурацию).

Минус я вижу только один – в небольшой инфраструктуре выделить целый один сервер под резерв может быть невозможно.

Если вдруг произойдет так, что отказ сервера ESXi произойдет в момент, когда не будет доступных серверов-резервов, то HA будет размещать виртуальные машины с отказавшего сервера по всем остальным. То есть сервер резерва не является тем сервером, на котором ВМ обязаны стартовать. Это всего лишь сервер с гарантированно свободными ресурсами.

Percentage of cluster resources reserved as failover spare capacity. Следующий вариант настройки «зачачки» – указание процента ресурсов процессора и памяти, который будет зарезервирован на каждом сервере.

На первый взгляд, эта настройка выглядит довольно разумно – мы не выделяем сервер или несколько для простаивания в качестве резервных, а запасаем сколько-то ресурсов на каждом сервере.

Однако тут я вижу две проблемы:

1. Сколько процентов указывать?
2. А что означают эти проценты?

Сколько процентов указывать, в принципе, не очень сложно посчитать. Допустим, у вас в кластере 16 серверов, и вы хотите иметь резервирование в 2 сервера из этих 16. Получается, наши X% с каждого из 16 должны давать 200% ресурсов одного сервера. По простой пропорции получаем, что в данном примере нам необходимо зарезервировать 12,5%. Есть, правда, один нюанс – если у вас есть виртуальная машина, которая одна потребляет больше, чем выбранный процент, то в случае перезапуска такой ВМ ей может ресурсов не хватить – ведь «зачачка» на случай сбоя фрагментирована, распределена между серверами. Получается, в некоторых случаях резервировать будет иметь смысл больший процент, чтобы его было достаточно самой «большой» из ваших ВМ.

А вот что эти проценты означают? Первое, что обычно приходит в голову, – что каждый один сервер не будет загружен более чем на 100% минус процент, указанный в настройках кластера. Но это неправильно. А как правильно?

Вот тут нам придется немного углубиться в подробности. Давайте разберем на примере. Скажем, у вас кластер из пяти серверов, в каждом сервере 100 Гб ОЗУ, вы резервируете 20% ресурсов на каждом – получается, у вас зарезервировано 100% ресурсов одного сервера. И, для примера, вы включили только три ВМ с объемом памяти, равным 1, 10 и 50 Гб (см. рис. 7.3). Внимание, вопрос: сколько процентов ресурсов кластера окажется занятым?

Правильный ответ – менее одного процента, что-то вроде одной пятисотой.

Обратите внимание. В данных примерах я не учитываю ресурсов, которые расходуются на сам гипервизор и накладные расходы. Впрочем, часто они пренебрежимо малы.

Name	Host Mem - MB	Guest Mem - %	Memory Size	Reservation - MB
SmallVM	553	25	1024 MB	1024 MB
MediumVM	787	7	10240 MB	0 MB
MonsterVM	1863	5	51200 MB	0 MB

Рис. 7.3. Иллюстрация нагрузки на кластер

Вот смотрите – для каждой ВМ есть несколько показателей ресурсов (я буду говорить на примере оперативной памяти, но для процессора это тоже справедливо). Вы видите их в столбцах с рис. 7.3:

- ❑ **Host Mem** – сколько физической ОЗУ сейчас выдано гипервизором этой ВМ;
- ❑ **Guest Mem** – какой процент от своего максимума сейчас активно использует ВМ;
- ❑ **Memory Size** – сколько памяти доступно гостевой ОС как максимум;
- ❑ **Reservation** – сколько памяти гипервизор гарантированно выдаст ВМ по первому требованию.

Внимание, вопрос: если НА зарезервировал 20% ресурсов (в моем примере), то в оставшиеся 80% должна поместиться сумма каких из этих столбцов?

Внимание, правильный ответ – reservation(!). Получается, если reservation = 0 (то есть вы не трогали значения по умолчанию), то не важно, сколько процентов ресурсов зарезервировал НА, – в оставшиеся проценты «поместятся» сколько угодно ВМ. По факту, если reservation оставлять равным нулю, данная настройка Admission Control лишена смысла.

Вернемся к примерам, следующий пример:

- ❑ пять серверов ESXi, по 100 Гб ОЗУ в каждом;
- ❑ типовая ВМ обладает 10 Гб памяти (столбец Memory Size с рисунка выше), но reservation = 0.

Если на таком кластере запустить 40 типовых ВМ, то:

- ❑ как максимум они займут 400 Гб памяти из 500 имеющихся;
- ❑ де-факто они займут, скорее всего, меньше (взгляните на столбец Host Mem с рисунка выше или в своей инфраструктуре).

Эта ситуация близка к идеальной. Но как раз в таких случаях и смысла что-то резервировать нет (предполагая ситуацию статичной, без увеличения числа ВМ в будущем).

Предположим далее, что каждая типовая ВМ стабильно потребляет 8 Гб из своих 10. В этом случае наши 40 типовых ВМ де-факто потребят 320 Гб из 500 Гб.

Это означает, что мы де-факто сможем запустить не 40, а 50 типовых ВМ – они де-факто потребят 400 Гб из 500 возможных, ресурсов будет достаточно, и даже если один из серверов откажет, ресурсов все равно будет достаточно. Эта ситуация

допустима. Она выгоднее, чем предыдущая, – ведь на тех же ресурсах запущено больше ВМ. Минусом является то, что если ВМ массово потребят максимум ресурсов, этих ресурсов может де-факто не хватить на всех.

А теперь разберем следующий случай: мы запустили 50 типовых ВМ, а затем случилось страшное:

- ❑ все ВМ начали потреблять не 8 Гб, а максимальные 10 Гб. Потребление де-факто стало равно 500 Гб;
- ❑ то есть на каждом одном сервере потребление де-факто стало равно 100%, 100 Гб в моем примере;
- ❑ отказал один из серверов. В кластере осталось всего 400 Гб памяти. Но ВМ совокупным объемом памяти в 100 Гб с отказавшего сервера необходимо перезапустить на остальных.

Внимание, вопрос: этот перезапуск произойдет?

Ответ: да. Гипервизор будет вынужден отнять часть памяти у остальных ВМ и эту отнятую распределить между ВМ с отказавшего сервера.

При настройках по умолчанию имеющиеся ресурсы будут поделены между ВМ поровну, так что каждая получит примерно 8 Гб из максимальных 10 Гб.

Однако если ранее настройки распределения ресурсов были изменены, то есть для приоритетных ВМ была увеличена настройка `shares` или/и `reservation` (на уровне каждой ВМ или (удобнее) разноприоритетные ВМ были помещены в разные пулы ресурсов), то в описываемом сценарии ресурсы будут поделены не поровну. Более приоритетные ВМ получат больше среднего за счет вытеснения в подкачку менее приоритетных.

Получается, если мы не можем себе позволить не включать «избыточные» с точки зрения НА виртуальные машины, альтернативой этому может послужить настройка распределения ресурсов между ВМ разной степени важности.

Наконец, рассмотрим последний сценарий. Точно так же:

- ❑ 5 серверов по 100 Гб;
- ❑ типовая ВМ с 10 Гб памяти, из которых 8 Гб потребляется, всего 48 таких ВМ;
- ❑ но у каждой ВМ стоит `reservation`, равный 8 Гб;
- ❑ и появилась одна уникальная ВМ с объемом памяти, равным 20 Гб, и все 20 Гб зарезервированы. Допустим, она была на отказавшем сервере.

В этом случае после отказа одного сервера на каждом из оставшихся четырех будет запущено по 12 ВМ. Они как максимум могут потребить 102 Гб памяти, то есть все 100 Гб, имеющиеся на каждом сервере. Но в среднем потреблять будут порядка 96 Гб – то есть при имеющихся вводных после отказа одного из серверов ресурсов будет хватать.

Хотя постойте, я же забыл про новую большую ВМ! А вот с ней проблема – она даже не включится. Все дело в ее `reservation` – это блокирующая настройка. Гипервизор или обязуется предоставить столько ресурсов, или не включает ВМ. В моем примере 12 типовых ВМ резервируют в совокупности 96 Гб памяти, то есть ни на одном сервере нет достаточного количества **свободных от `reservation`** ресурсов для включения последней ВМ.

Отсюда вывод: резервирование процента ресурсов в настройке Admission Control – это не способ резервировать ресурсы на потребление де-факто, а способ резервирования ресурсов на reservation виртуальных машин.

Host failures the cluster tolerates. Этот вариант настройки контроля «зачапки» часто называют «по слотам». В самом окне настроек мы указываем цифру от 1 до 31 – это число серверов, отказ которого кластер должен пережить. А теперь HA сам рассчитывает, сколько ресурсов надо зарезервировать на случай отказа указанного количества серверов кластера при **текущем** количестве виртуальных машин. И если количество ВМ или их настройки ресурсов изменятся, будет произведен перерасчет. В этом основное отличие данного способа расчета «зачапки» от резервирования процентов – проценты мы должны посчитать и указать самостоятельно, а «слоты» HA рассчитает сам.

Если для кластера HA мы указали настройки Admission Control = «Prevent VMs from being powered on», то HA начинает резервировать ресурсы кластера. Количество резервируемых ресурсов зависит от производительности серверов, количества и настроек виртуальных машин, от значения **Host failures cluster tolerates** (мы сейчас говорим про настройку Admission Control Policy в значении «Host failures cluster tolerates»).

Смотрите рис. 7.4.

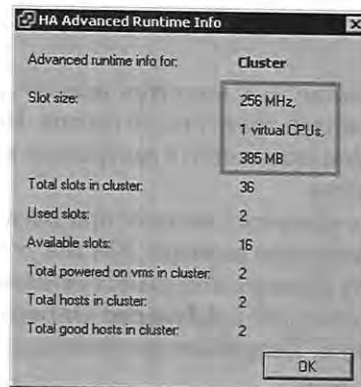


Рис. 7.4. Информация про слоты в HA-кластере

Здесь мы видим:

- ☐ **Slot size** – размер слота для этого кластера;
- ☐ **Total slot in cluster** – сумма слотов на хороших серверах (good hosts, см. последний пункт списка);
- ☐ **Used slots** – сколько слотов сейчас занято;
- ☐ **Available slots** – сколько слотов еще можно занять, чтобы кластер продолжил быть отказоустойчивым;
- ☐ **Total powered on VMs in cluster** – всего включенных ВМ;
- ☐ **Total hosts in cluster** – всего серверов в кластере;

- ❑ **Total good hosts in cluster** – всего «хороших» серверов в кластере. Хорошими считаются сервера: не в режиме обслуживания, невыключенные и без ошибок НА.

Итак, в данном примере мы видим, что всего серверов в кластере 2, из них «хороших» тоже 2. Всего включенных ВМ 2. Всего слотов 36, занято 2, доступны 16. Размер одного слота – 1 vCPU, 256 МГц и 385 Мб ОЗУ. Откуда берутся эти цифры? Почему из 34 ($= 36 - 2$) слотов доступны лишь 16?

«Слот» – это количество ресурсов под виртуальную машину. Очень приблизительное количество ресурсов, которое считается по следующей схеме:

1. Смотрится reservation всех ВМ в кластере.
2. Выбирается наибольшее значение, отдельно по памяти и отдельно по процессору.
3. Оно и становится размером слота. Но для памяти к значению reservation добавляются накладные расходы (memory overhead, его можно увидеть на вкладке **Summary** для виртуальной машины).
4. Если reservation у всех нуль, то берется значение слота по умолчанию – 256 МГц для процессора и максимальное среди значений overhead для памяти.
5. Ресурсы процессора и памяти сервера делятся на посчитанные в пп. 3 и 4 величины с округлением вниз. Выбирается наименьшее значение – столько слотов есть на этом сервере. Эти расчеты выполняются для каждого сервера.

Вернитесь к рисунку выше. НА посчитал размер слота и сделал вывод, что ресурсов двух серверов кластера хватит на 36 слотов. Но НА гарантирует, что запустятся все ВМ при падении указанного в настройках количества серверов, лишь если занято не более 16 слотов.

Эти расчеты лично мне кажутся слишком приблизительными – предположу, что в большинстве инфраструктур размеры ВМ могут значительно различаться. Так что будет лучше указать размер слота самостоятельно, с помощью параметров `das.slotCpuInMHz` и `das.slotMemInMB` в **Advanced Options** для НА-кластера можно указать размер слота вручную. Подробнее про расширенные настройки (Advanced Settings) для кластера НА далее.

Сколько слотов в кластере, считается простым делением ресурсов каждого сервера на размер слота. Если сервера в кластере разной конфигурации, то расчет количества слотов, которое резервируется «на черный день», делается из наихудшего сценария, то есть исходя из выхода из строя серверов с наибольшим количеством ресурсов.

Так что если у вас в кластере сервера со 100 Гб ОЗУ каждый и размер слота вы указали равным 1 Гб, то на каждом сервере 100 слотов.

Внимание, вопрос: если у вас три ВМ, как на рис. 7.3, то сколько слотов занимает каждая из них?

Правильный ответ: каждая по одному. Дело в том, что расчеты идут по reservation. Так что если размер слота 1 Гб, у ВМ 50 Гб памяти, но reservation = 0, то она все равно занимает один слот в расчетах НА.

Так что вариант резервирования по слотам, как и вариант резервирования по процентам, не работает, если не указан адекватный `reservation` для каждой важной ВМ.

А теперь поговорим про Admission Control с точки зрения интерфейса – обратите внимание на рис. 7.5.

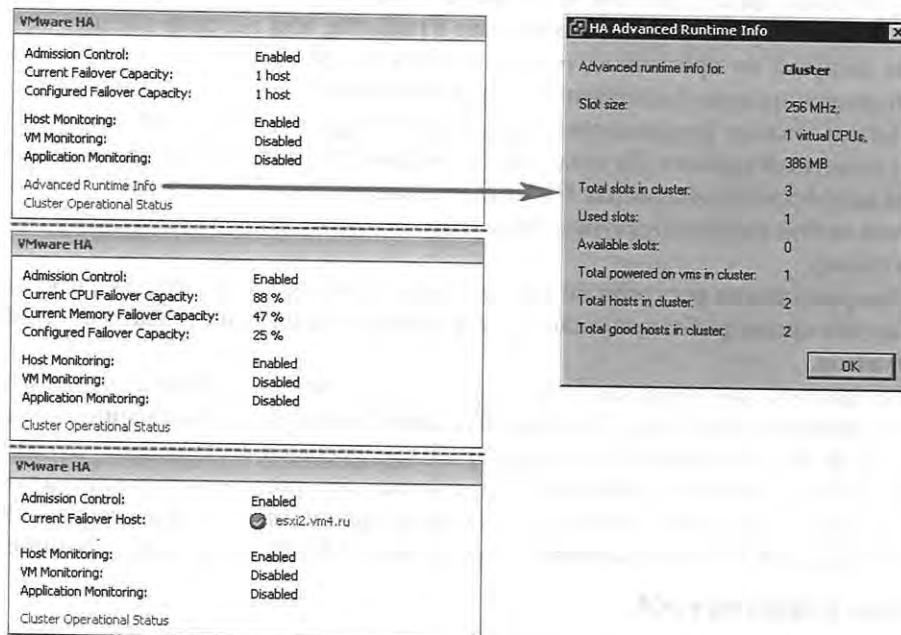


Рис. 7.5. Информация на вкладке **Summary** для HA-кластера

Здесь вы видите три варианта информации со вкладки **Summary** для кластера с включенным HA. Эти три варианта соответствуют трем вариантам настройки **Admission Control**. Строка **Configures failover capacity** напоминает о том, какое значение вы дали настройке резервирования ресурсов. В моем примере, сверху вниз:

- ☐ отказоустойчивость в один сервер настроена и отказоустойчивость в один сервер есть – на последнее указывает **Current failover capacity**. По нажатии ссылки **Advanced Runtime Info** открывается окно с дополнительной информацией по текущему состоянию кластера;
- ☐ резервирование 25% ресурсов указано, а свободно в данный момент 88% для процессора и 47% для памяти;
- ☐ резервным сервером является сервер `esxi2.vm4.ru`. Зеленый значок напротив его имени означает, что сервер доступен и на нем нет работающих ВМ. Желтый значок означал бы, что сервер доступен, но на нем есть работающие ВМ. Красный – что сервер недоступен, или агент HA на нем не работает правильно.

Подведу резюме. Какой из вариантов резервирования ресурсов использовать?

Я бы рекомендовал или вообще отключить Admission Control, или использовать третий вариант резервирования – резервирование конкретного сервера или нескольких.

Отключение Admission Control оправдано в том случае, когда сервера вашей виртуальной инфраструктуры всегда не загружены больше, чем процентов на 80. Если ресурсов процессора и памяти у вас в избытке, или вы сами следите за тем, чтобы загрузка не превышала тех самых 80% (плюс-минус), то дополнительно что-то резервировать средствами НА просто незачем.

Использование резервирования целого сервера нравится мне тем, что это очень понятный вариант. На указанных серверах нет ни одной виртуальной машины, на всех остальных серверах НА не вмешивается. Из минусов я вижу только то, что не в любой инфраструктуре можно себе позволить держать свободным целый один сервер.

Резервирование ресурсов по процентам или по слотам не всегда применимо, потому что **они не работают, если для виртуальных машин не указана настройка reservation**.

Резервирование ресурсов по процентам и по слотам не нравится мне тем, что резервирование делается не для реальных appetites виртуальных машин, а для их reservation. То есть если НА резервирует 25% ресурсов, то это означает, что сервер может быть загружен и на 100%, а 75% относится к сумме reservation виртуальных машин. Это означает, что если для всех или большинства ВМ вы настройку reservation не делали, то такой вариант резервирования ресурсов вообще бесполезен.

Как работает НА

В тот момент, когда вы включаете НА для кластера, vCenter устанавливает на каждом сервере агент НА. В пятой версии vSphere был полностью переписан код этого агента, и теперь его называют агент FDM, Fault Domain Manager. То есть FDM – это такое внутреннее название НА.

Один из серверов кластера назначается старшим, так называемым Failover Coordinator, координатором. Именно он принимает решение о рестарте ВМ в случае отказа. А если откажет он сам – любой из остальных серверов готов поднять упавшее зная, вся необходимая информация о состоянии дел есть на каждом сервере кластера.

Агенты обмениваются между собой сообщениями пульса (heartbeat) вида «я еще жив». По умолчанию обмен происходит каждую секунду. Для этих сообщений используется сеть управления.

Если за 10-секундный интервал от сервера не пришло ни одного сообщения пульса (heartbeat) и сервер не ответил на пинги еще 10 секунд, то он признается отказавшим. Сервер-координатор начинает перезапуск ВМ с этого сервера на других. Причем НА выбирает для включения ВМ сервер с наибольшим количеством незарезервированных ресурсов процессора и памяти, и выбор этот происходит перед включением каждой ВМ.

НА делает 5 попыток включить ВМ с отказавшего сервера.

Первая попытка происходит после признания сервера отказавшим, это 20 секунд по умолчанию (немного больше, если отказал сам координатор, потребуется дополнительное время на перевыборы).

Вторая – через 2 минуты.

Третья – через 6 минут.

Четвертая – через 14 минут.

Пятая – через 30 минут.

На вкладке **Summary** для кластера HA есть ссылка **Cluster Operational Status**. В открывающемся при клике по ней окне отображаются проблемы (если они есть) с узлами кластера (рис. 7.6).

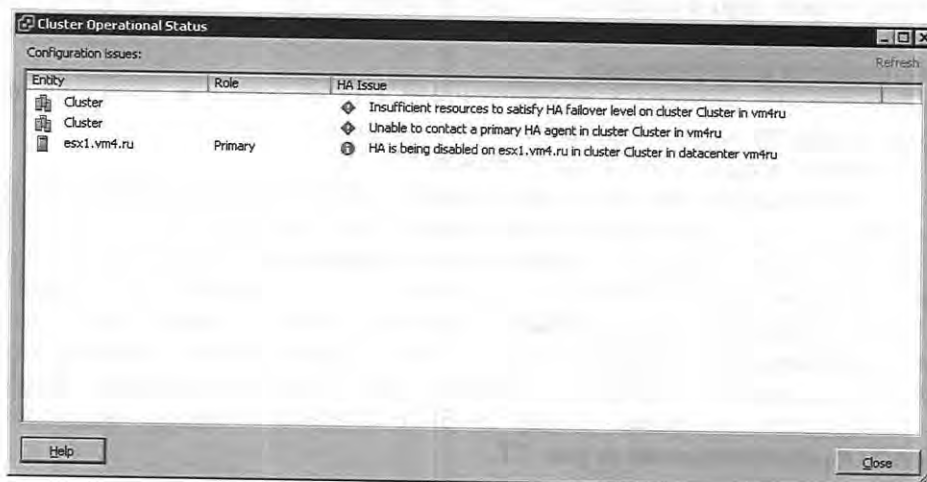


Рис. 7.6. Информация о проблемных узлах кластера HA

Очевидно, в ваших интересах решать отображаемые здесь проблемы в случае их появления.

Если вы работали с предыдущими версиями vSphere, то что имеет смысл сказать про изменения в плане HA:

- ☐ с точки зрения интерфейса, изменений практически нет;
- ☐ с точки зрения работы – стало меньше условий и ограничений.

Каких ограничений теперь нет:

- ☐ агент HA не зависит от DNS;
- ☐ файлы журналов агента HA теперь сохраняются на общих основаниях, через syslog. Все события HA попадают в файл журналов /var/log/fdm.log;
- ☐ если ранее сервера кластера делились между тремя категориями (один Failover Coordinator, четыре его заместителя и все остальные в роли Slave), то сейчас категорий всего две – координатор и Slave. Любой из Slave может стать координатором в случае отказа ранее выбранного координатора. Если ранее кластер HA гарантированно мог пережить отказ 4 серверов, то сейчас – сколько угодно;

- ❑ если из-за сетевого сбоя кластер оказался разбит на две несвязанные части, то обе будут полноценно функционировать, в каждой окажется выбранным по координатору.

Более детальную информацию вы можете обнаружить по ссылке <http://link.vm4.ru/ha-detail>.

Изоляция. Datastore Heartbeating

Сервера в кластере НА проверяют доступность друг друга сообщениями пульса (heartbeat) по сети управления. Однако эти сигналы могут перестать ходить не потому, что упал сервер, а потому, что упала сеть. Физический контроллер может выйти из строя, порт в коммутаторе, коммутатор целиком. Ситуация, когда ESXi работает, но недоступен по сети, называется *изоляция*. Недоступный по сети сервер становится изолированным.

Виртуальные машины с изолированного сервера могут, в принципе, или остаться работать на этом сервере, или выключиться.

Внимание, вопрос: что бы мы предпочли?

Изоляция случается тогда, когда возникают проблемы в сети управления. Это означает, что наш ESXi недоступен для управления, однако ВМ на нем продолжают работать. Другие сервера кластера, увидев пропажу сигналов пульса с этого сервера, попытаются включить его ВМ у себя, но не смогут из-за блокировки файлов на VMFS. Чтобы ВМ смогли перезапуститься на других серверах НА-кластера, их надо выключить. Это соображение номер раз, почему мы можем захотеть, чтобы ВМ на изолированном сервере выключились. Кстати, попыток включить будет 5.

Соображение номер два – а будут ли ВМ с отказавшего сервера иметь доступ в сеть? Обратите внимание на рис. 7.7.

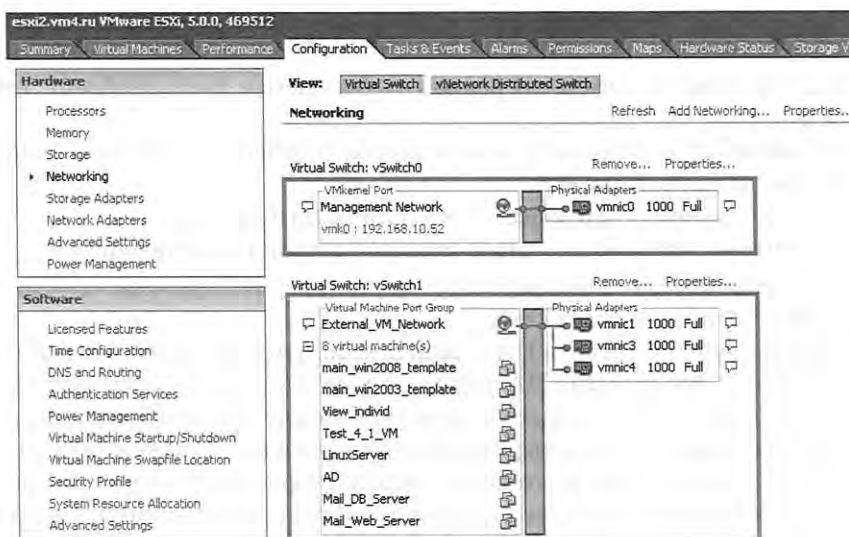


Рис. 7.7. Пример организации сети

Как вы видите, в данном примере сеть управляющая и сеть виртуальных машин разнесены физически. И выход из строя сетевого контроллера `vmpic0` приведет к изоляции сервера, но не к прекращению работы ВМ по сети. Пытливый читатель, правда, может привести пример глобального сбоя – если `vmpic0` и `vmpic1,3,4` подключены к одному коммутатору и из строя вышел коммутатор. Однако, во-первых, коммутатор не должен быть незадублирован; и во-вторых – тогда сеть пропадет на всех ESXi, и HA здесь не поможет.

Вывод из этих двух соображений следующий:

- ❑ если изоляция сервера означает пропажу сети для ВМ (потому что для их трафика и для управления используются одни и те же каналы во внешнюю сеть), то однозначно нужно настроить выключение ВМ в случае изоляции. Тогда HA перезапустит их на других серверах кластера;
- ❑ если изоляция не означает пропажу сети для ВМ (как в моем примере), то нужно уже подумать – выключать или нет. Если вам не нравится, что ВМ продолжают работать на сервере, управление которым вы потеряли, то выключайте, их перезапустят на других серверах. Но многие оставляют «Leave powered on», дабы уменьшить вероятность накладок при выключении и переключении виртуальных машин;
- ❑ если часть ВМ расположена на локальных или частных хранилищах и HA не сможет их переключить – однозначно их имеет смысл оставить включенными.

В настройках кластера мы можем указать, что должен делать сервер, оказавшись изолированным (рис. 7.8).

Как вы видите, вариантов три:

- ❑ **Leave powered on** – не делать ничего, оставить ВМ работающими;
- ❑ **Power off** – отключить питание ВМ, некорректное завершение работы;
- ❑ **Shutdown** – корректное завершение работы. Требуется установленных в ВМ VMware tools. Если за 300 (по умолчанию) секунд ВМ не выключилась, ее выключают уже некорректно.

Настройки делаются для всех ВМ кластера (верхнее выпадающее меню) и могут быть индивидуально изменены для каждой виртуальной машины.

Ну и чтобы эти настройки отработали, необходимо, чтобы сервер понял, что он изолирован. ESXi умеет обнаруживать изоляцию. Если сервер перестал получать сообщения пульса (heartbeat) от всех прочих серверов, то через 12 секунд он пробует пропинговать проверочный IP-адрес. Если тот на пинг не отозвался, сервер признает себя изолированным. По умолчанию проверочным является адрес шлюза по умолчанию, но вы можете указать произвольный адрес и даже несколько через расширенные настройки (Advanced Settings).

Таким образом, если сервер в кластере перестал принимать сообщения от других серверов и плюс к тому перестал пинговать проверочные IP-адреса, то он считает себя изолированным. (Если же другие сервера пропали, но проверочный адрес пингуется, то отказавшими признаются прочие сервера.) Признав себя изолированным, сервер применяет вышеупомянутую настройку к своим ВМ, то есть выключает их или не делает ничего.

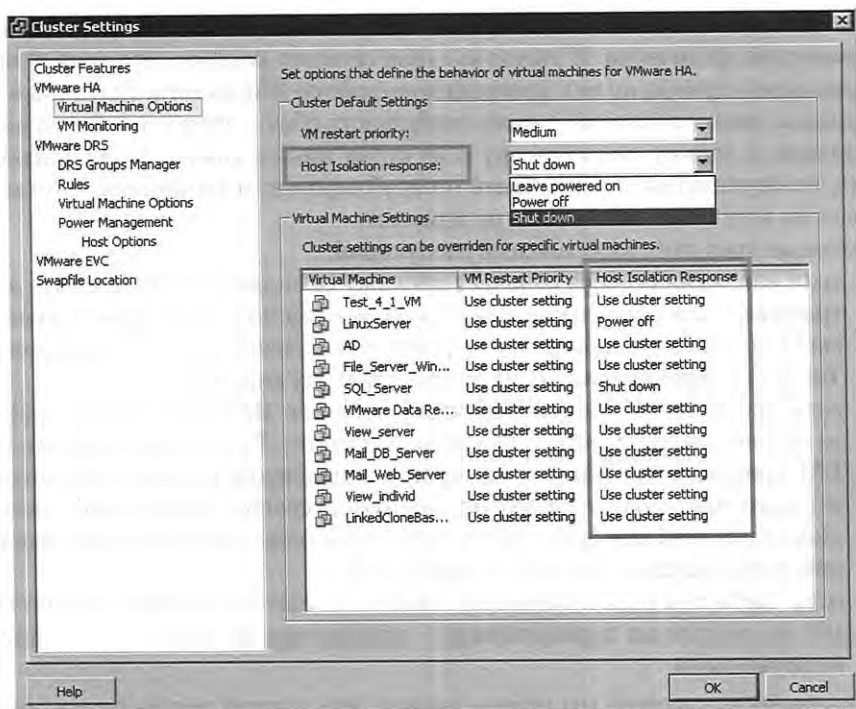


Рис. 7.8. Настройки реакции на изоляцию

Однако если сервер оказался изолированным, но не стал выключать ВМ, то как это видят прочие сервера? Они видят просто отказавший сервер и в любом случае пытаются перезапустить его ВМ. Но у них это не удастся, потому что файлы ВМ заблокированы. Но они все равно предпринимают несколько попыток, что увеличивает нагрузку. Это не очень хорошо. К счастью, в пятой версии vSphere этот момент оптимизировали.

В пятой версии VMware добавила еще один механизм, имеющий отношение к изоляции, — это обмен информацией через хранилища. В настройках кластера мы можем указать несколько хранилищ, которые будут использоваться, чтобы хранить на них информацию HA.

На этих хранилищах создаются файлы каждым сервером, и эти файлы блокируются соответствующим сервером ESXi стандартными механизмами блокировок VMFS. В каждом таком файле хранится список запущенных на конкретном сервере виртуальных машин, а в начале файла хранится статус сервера в контексте изоляции. Так что если сервер обнаружил, что оказался изолирован, то он сообщает об этом через хранилища.

Если изолированный сервер выключает все (или некоторые ВМ) — он сообщает об этом, и прочие сервера кластера будут перезапускать только выключенные ВМ с изолированного сервера.

А если сервер в действительности отказал, то он не обновит в очередной раз блокировку своего файла на хранилищах, и координатор сделает отсюда вывод, что сервер отказал.

Проверки файлов на хранилище происходят после проверок на сетевом уровне.

Можно доверять выбрать эти хранилища vCenter и сделать это самостоятельно, в свойствах кластера НА. Выбирать имеет смысл те хранилища, что видны сразу всем серверам одного кластера. Кроме того, правильно будет выбрать хранилища с разных систем хранения – чтобы СХД не была точкой отказа для работы этого механизма.

Обращу ваше внимание на то, что использование хранилищ для задач проверки на изоляцию носит вспомогательный характер, позволяя более точно определить статус сервера – отказал он или изолирован, и стал ли выключать ВМ и какие, в случае если изолирован. Данные о ВМ на хранилищах тоже не являются критично необходимыми – список запущенных на каждом сервере ВМ составляется и редактируется по ходу штатной работы кластера и хранится на каждом сервере ESXi.

Как видно, для НА особенно важна конфигурация сети. На что здесь следует обратить внимание:

- ☐ на дублирование управляющей сети;
- ☐ на дублирование физических коммутаторов;
- ☐ проверочный IP-адрес должен отвечать на пинги 24/7. Вариант по умолчанию – шлюз для управляющего интерфейса хорош далеко не всегда, и он один. В разделе, посвященном расширенным настройкам (Advanced Settings), для кластера НА сказано, как задавать произвольный и дополнительные проверочные IP.

К дублированию управляющих интерфейсов у вас два подхода (рис. 7.9).

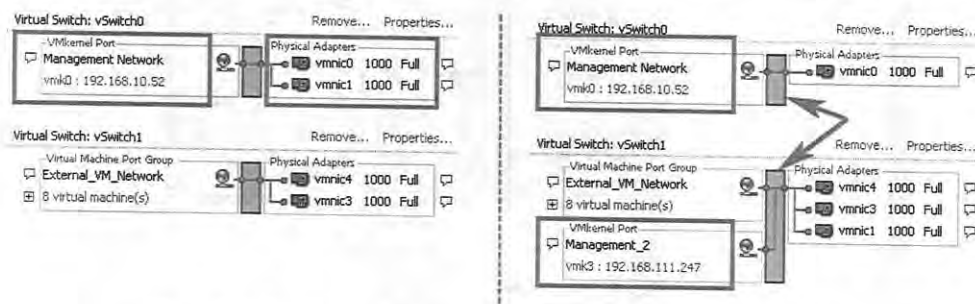


Рис. 7.9. Схемы дублирования управляющих интерфейсов

Слева вы видите один управляющий интерфейс, подключенный к двум физическим контроллерам. А справа – два управляющих интерфейса на разных физических контроллерах. В любом случае, мы достигаем желаемого – отсутствия единой точки отказа.

Если управляющих интерфейсов несколько, то сообщения пульса (heartbeat) будут посылаются через все.

Кстати, если у вас не будет дублирования по одной из этих схем, то статус сервера будет Warning, и на вкладке **Summary** для сервера вы будете видеть его ругань по тому поводу (рис. 7.10).

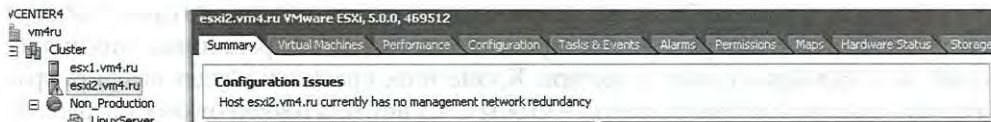


Рис. 7.10. Сообщение об отсутствии дублирования управляющего интерфейса сервера

Это информационное сообщение, работе кластера не препятствующее. Есть возможность отключить это сообщение, см. **Advanced Settings**.

VM Monitoring

Настройки VM Monitoring, рис. 7.11.

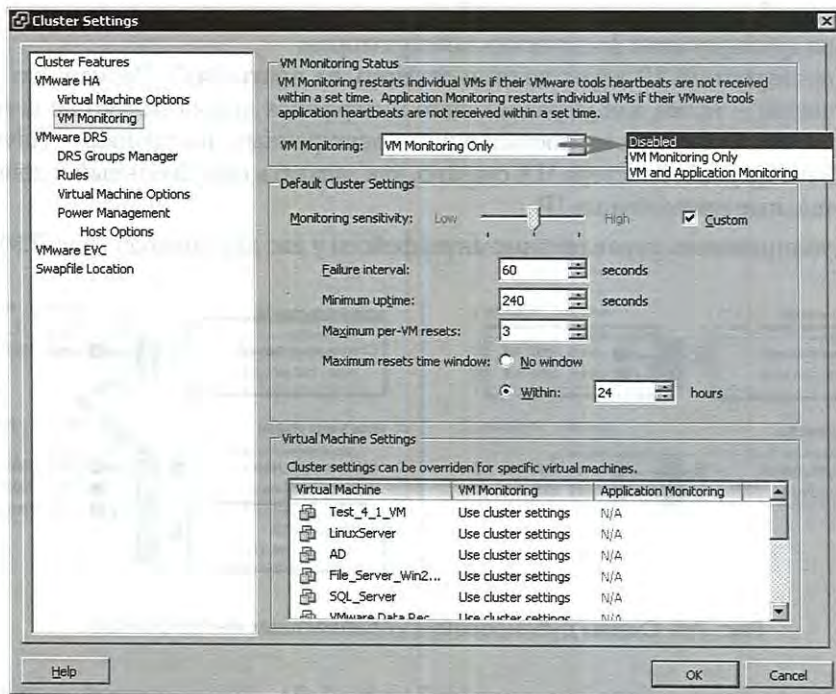


Рис. 7.11. VM Monitoring

Суть этого механизма HA – в том, что он отслеживает наличие сигналов пульса (heartbeat) от VMware tools. Предполагается, что отсутствие этих сигналов означает зависание VMware tools вследствие зависания гостевой ОС.

Если сигналы пульса (heartbeat) отсутствуют в течение «Failure interval» секунд, а с момента старта ВМ прошло не менее «Minimum uptime» секунд, то ВМ перезагружается. Однако ее перезагрузят не больше «Maximum per-VM resets» раз за время «Maximum resets time window».

Этот механизм, по сути, является аналогом так называемого watchdog-таймера, реализованного в BIOS многих серверов.

Описание настроек:

- ☐ **VM monitoring status** – если этот флажок не стоит, то данный механизм не работает;
- ☐ **Default Cluster settings** – настройки работы механизма. На рисунке приведен вариант «Custom». Если одноименный флажок не стоит, то мы можем выбрать один из трех вариантов по умолчанию;
- ☐ **Virtual Machine Settings** – указание предыдущих настроек индивидуально на уровне каждой ВМ.

Иногда возможны ситуации, что сигналы пульса (heartbeat) от VMware tools пропали, но ВМ работает. Чтобы не перезагружать работающую ВМ в такой ситуации, компонент VM Monitoring отслеживает еще и активность ввода-вывода ВМ. Если сигналы пульса (heartbeat) пропали и не возобновились вновь в течение «Failure interval» секунд, то смотрят на активность работы этой ВМ с диском и сетью за последние 120 секунд. Если активности не было, ВМ перезагружается.

Мне этот механизм представляется слишком грубым, чтобы с ходу начать его использовать. Однако если у вас есть регулярно зависающие виртуальные машины и перезапуск как решение проблемы вас устраивает, функция VM Monitoring подойдет отлично (включить ее следует только для этих ВМ).

Обратите внимание, что в версии 4.1 VMware реализовала API для сторонних поставщиков кластерных решений под названием Application Monitoring. Суть этого механизма – в том, что сторонние агенты, установленные в гостевые ОС, могут отслеживать статус приложений и взаимодействовать с сервером vCenter и агентами НА для инициации перезагрузки ВМ при проблемах в работе приложения. На момент написания единственным известным мне приложением, реализующим Application Monitoring для VMware НА, является Symantec ApplicationNA.

Так как настройки подобного рода дополнений в полной мере зависят от используемого стороннего средства, здесь они не рассматриваются.

Advanced Options

Для кластера НА могут быть выполнены некоторые расширенные настройки. Список таких настроек см. в табл. 7.1.

Список неполон, в базе знаний и в советах поддержки VMware вам могут встретиться и другие настройки.

Обратите внимание. «das» в названии этих настроек присутствует потому, что эта аббревиатура – первый вариант названия того, что сейчас называется «кластер НА».

Чтобы указать эти настройки, зайдите в свойства кластера и в настройках НА нажмите кнопку **Advanced Settings** (рис. 7.12).

Таблица 7.1. Список расширенных настроек (Advanced Settings) для кластера HA

Название настройки	Описание
Сетевые настройки	
das.isolationaddress[...]	Значение – IP-адрес. Указание произвольного IP-адреса как проверочного на изоляцию IP. das.isolationaddress1, das.isolationaddress2..das.isolationaddress10 являются дополнительными проверочными адресами. Слишком много лучше не указывать, иначе процесс проверки на изоляцию затянется. Обычно указывают по одному на каждый управляющий интерфейс или два для единственного управляющего интерфейса. Сервер будет считать себя изолированным, если перестанут отвечать все из указанных проверочных адресов. Изменение требует переключения HA-кластера
das.usedefaultisolationaddress	Значение – true/false. Использовать или нет шлюз по умолчанию как проверочный адрес. Изменение требует переключения HA-кластера
das.ignoreRedundantNetWarning	Значение – true/false. Если true, то пропадает предупреждение об отсутствии резервирования сети управления
das.config.fdm.deadIcmpPingInterval	Значение – число. По умолчанию 10. Через столько секунд координатор начнет пинг slave-сервера, связь с которым пропала
das.config.fdm.icmpPingTimeout	Значение – число. По умолчанию 5. Столько секунд координатор ждет ответа на пинг
das.config.fdm.hostTimeout	Значение – число. По умолчанию 10. Столько секунд координатор ожидает возобновления пропавших сигналов пульса, перед тем как признать сервер отказавшим и переходить к следующему шагу – определению, отказал ли сервер или изолирован
Разное	
das.config.fdm.stateLogInterval	Значение – число. По умолчанию 600. Через столько секунд сохраняется состояние кластера в журнал
das.perHostConcurrentFailoversLimit	Значение – число. По умолчанию 32. Столько VM может быть параллельно включено на одном сервере при обработке отказа
das.isolationShutdownTimeout	Значение – число. Время, в течение которого ожидается выключение VM при изоляции, когда выбрано корректное завершение работы. После окончания этого времени VM выключается жестко. По умолчанию 300 секунд. Изменение требует переключения HA-кластера
Slot Size	
das.slotMemInMB	Значение - число. Количество памяти для слота, в мегабайтах. При выборе размера слота это значение сравнивается с наибольшим значением reservation + overhead между всеми включенными VM в данном кластере. Меньшее из двух этих значений и будет принято за размер слота

Таблица 7.1. Список расширенных настроек (Advanced Settings) для кластера HA (окончание)

Название настройки	Описание
das.slotCpuInMHz	Значение – число. Количество ресурсов процессора для слота, в мегагерцах. При выборе размера слота это значение сравнивается с наибольшим значением reservation между всеми включенными ВМ в данном кластере. Меньшее из двух этих значений и будет принято за размер слота
das.vmMemoryMinMB	Значение – число. Количество памяти, используемое для расчетов слота под ВМ, когда ее reservation для памяти равен нулю. По умолчанию 0
das.vmCpuMinMHz	Значение – число. Количество ресурсов процессора, используемое для расчетов слота под ВМ, когда reservation для процессора равен нулю. По умолчанию 256 МГц
VM Monitoring	
das.iostatsInterval	Значение – число. Количество секунд, за которое анализируется активность ввода-вывода ВМ, когда пропали сигналы пульса (heartbeat) от VMware tools. По умолчанию 120 секунд
das.config.fdm.policy.unknownStateMonitorPeriod	Значение – число. Столько секунд координатор ожидает после определения сбоя ВМ, перед тем как попытаться ее перезапустить
Datastore Heartbeat	
das.ignoreInsufficientHbDatastore	Значение – true/false. Если true, то подавляется сообщение о недостаточном количестве хранилищ, настроенных как datastore heartbeat
das.heartbeatDsPerHost	Значение – число между 2 и 5. Количество хранилищ под цели HA для каждого сервера. По умолчанию 2
Fault Tolerance	
das.maxFtVmsPerHost	Значение – число. По умолчанию 4. Столько ВМ, защищенных при помощи FT, может работать на одном сервере ESXi
das.includeFTcomplianceChecks	Значение – true/false. Если false, то при настройке кластера не производится проверка на соответствие условиям FT
das.config.fdm.ft.cleanupTimeout	Значение – число. По умолчанию 900. Столько времени координатор HA ожидает включения secondary-ВМ. По истечении этого времени HA повторяет попытку включения

Использование HA и DRS вместе

HA и DRS без каких-либо проблем можно использовать для одного кластера вместе. Более того, так даже лучше. Функционал этих решений не пересекается, а дополняет друг друга.

Обратите внимание, при оценке используемых в данный момент ресурсов HA смотрите не на *текущую* нагрузку, а на значения *reservation*. Это означает, что он

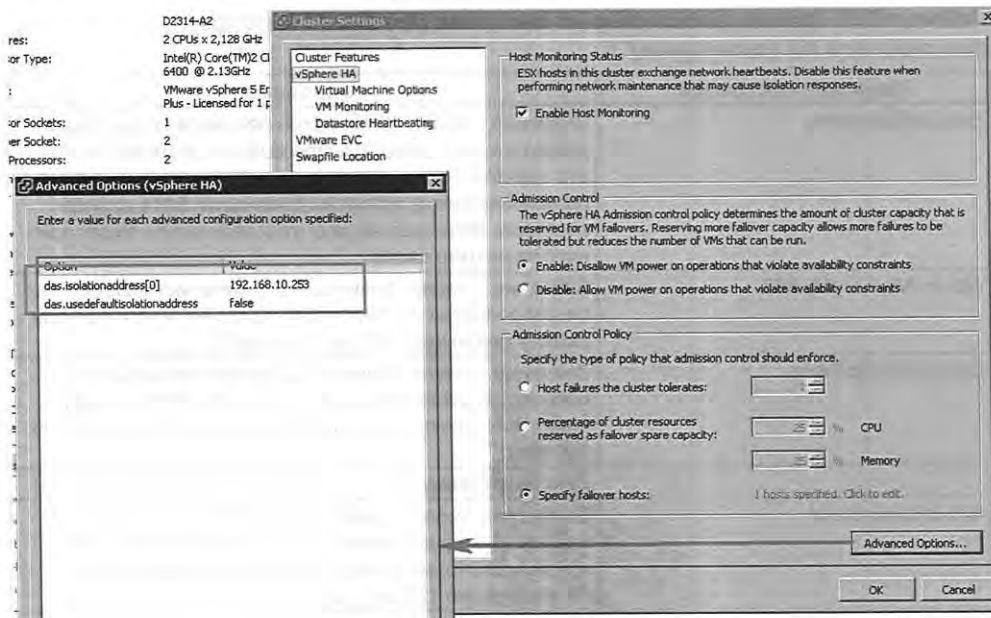


Рис. 7.12. Расширенные настройки для кластера HA

не самым эффективным образом выбирает сервера для рестарта ВМ. Поэтому для вас намного удобнее использовать HA вместе с DRS, который сбалансирует нагрузку более эффективно.

Начиная с версии 4.1, при включении HA и DRS вместе последний будет стараться консолидировать свободные ресурсы на каком-то одном сервере, чтобы снизить вероятность «фрагментирования» свободных ресурсов между серверами кластера.

Если для HA включен Admission Control, то DRS может не удастся мигрировать ВМ с серверов в режиме обслуживания, если они будут претендовать на зарезервированные HA ресурсы на других серверах. В таком случае вам придется временно отключить Admission Control.

Если Admission Control выключен, то HA не резервирует ресурсов. Поэтому DRS сможет мигрировать ВМ с сервера в режиме обслуживания и stand-by, даже если это сделает конфигурацию уязвимой к сбою.

7.1.2. VMware Fault Tolerance, FT

Задачей VMware HA-кластера является *минимизация времени простоя всех* или большинства ВМ из-за отказа сервера (а считая компонент VM Monitoring – и из-за отказа на уровне гостевой ОС). А VMware Fault Tolerance позволяет *отдельные ВМ избавить от простоев* из-за отказа сервера (подразумевается аппаратный сбой или проблема с самим ESXi). Предполагается, что таким образом защищать мы будем наиболее критичные для нас ВМ.

Обратите внимание. FT не защитит VM от сбоя системы хранения или от программного сбоя приложения и гостевой ОС. Зато от сбоя сервера эта функция защищает очень качественно. Кроме того, она работает прозрачно для гостевой ОС и приложений, таким образом не налагая на них условий и не требуя настроек на этом уровне.

Суть FT – в том, что для защищаемой VM создается ее копия на другом сервере. И выполняемые исходной VM процессорные инструкции непрерывно реплицируются на копию. Если падает сервер, на котором работает исходная VM, то достаточно выпустить в сеть копию, чтобы работа продолжилась без перерыва.

Еще один вариант – отказоустойчивость по требованию. Например, есть VM с формирующим отчетность сервером. Обычно эта VM защищена только HA. Но в отчетный период, когда прерывание формирования отчета грозит потерей времени, для этой VM легко включить FT и получить большую доступность, чем дает HA.

Настройка VMware FT

Для работы VMware FT должны быть выполнены некоторые условия.

Условия для инфраструктуры:

- ☐ должен существовать кластер HA. FT является его подфункцией. Притом если HA включается для кластера и защищает все VM в нем, то FT включается индивидуально для отдельных VM в нем;
- ☐ для всех серверов, использующихся для FT, должна быть включена проверка сертификатов серверов (она включена по умолчанию);
- ☐ на каждом сервере должен быть интерфейс VMkernel, настроенный для VMotion, и интерфейс VMkernel, настроенный для FT Logging (и то, и другое – флажки в свойствах интерфейса VMkernel). VMware рекомендует, чтобы это были два разных интерфейса, работающие через разные физические сетевые контроллеры;
- ☐ между серверами должна быть совместимость по процессорам;
- ☐ начиная с версии 4.1 сервера не обязаны иметь одинаковую версию ESXi и одинаковый набор обновлений. В новых версиях vSphere проверяется только совместимость версий компонента, отвечающего за Fault Tolerance. Таким образом, вполне возможна ситуация, когда FT работает между хостами разных версий ESXi, и даже версии FT-компонента могут отличаться – но они должны быть совместимы;
- ☐ защищаемые FT VM должны использовать дисковые ресурсы, доступные со всех серверов. Еще раз обращу внимание – VM и ее копия используют одну копию дисков.

Условия для серверов:

- ☐ процессоры серверов должны быть из списка совместимости VMware Fault Tolerance. Подробности – в статье базы знаний № 1008027. Желательно, чтобы тактовая частота процессоров серверов отличалась не более чем на 300 МГц;
- ☐ в BIOS серверов должна быть включена аппаратная поддержка виртуализации.

Условия для виртуальных машин. К сожалению, FT налагает достаточно много ограничений на виртуальную машину под своей защитой:

- ❑ у виртуальной машины не должно быть снимков состояния (snapshot) на момент включения FT, и их нельзя создавать для VM под защитой FT. Это может быть важно для резервного копирования этой VM – многие решения резервного копирования используют снимки состояния в своей работе;
- ❑ VMware протестировала FT не для любых ОС и не любых комбинаций ОС и процессоров. Подробности – в статье базы знаний № 1008027;
- ❑ нельзя осуществить Storage VMotion для VM под защитой FT;
- ❑ DRS получил полную интеграцию с FT начиная с версии 4.1. Теперь Primary и Secondary виртуальные машины могут быть перенесены между серверами для балансировки нагрузки, в том числе автоматически;
- ❑ у VM должен быть только один vCPU. Это очень сильно ограничивает применение данной функции для критичных и требовательных к процессору задач, ведь один vCPU – это одно ядро физического процессора. Уточню – у VM должен быть один одноядерный виртуальный процессор, и никак иначе;
- ❑ к VM не должны быть подключены диски в формате physical RDM;
- ❑ CD-ROM и FDD этой VM могут ссылаться только на файлы-образы с общих хранилищ. Если подмонтирован образ с приватного хранилища и произошел сбой сервера с Primary VM, то переезд состоится, но новая Primary к этому образу доступа уже не получит;
- ❑ не поддерживаются VM с паравиртуализованным SCSI-контроллером, так что в конфигурации VM не должно быть PVSCSI;
- ❑ не должно быть USB- и аудиоустройств;
- ❑ NPIV должен не использоваться для этой VM;
- ❑ VMDirectPath I/O должен не использоваться для этой VM;
- ❑ для защищенной FT VM невозможно горячее добавление устройств;
- ❑ не поддерживается Extended Page Tables/Rapid Virtualization Indexing (EPT/RVI);
- ❑ файлы VM должны быть расположены на общем хранилище. Тип хранилища не важен;
- ❑ диском VM может быть virtual RDM или файл vmdk типа eagerzeroedthick. Для создания такого vmdk укажите тип **Thick Provision Eager Zeroed** при его создании (рис. 7.13).

Впрочем, диски VM можно преобразовать и после создания. Для этого поможет любое действие из следующего списка:

- ❑ запуск Storage VMotion и выбор **Thick Provision Eager Zeroed** как тип дисков;
- ❑ или пункт **Inflate** в контекстном меню файла vmdk, если найти его через встроенный файловый менеджер;
- ❑ или команда `vmkfstools --diskformat eagerzeroedthick`;
- ❑ наконец, самое простое – при включении FT сам мастер предложит вам изменить тип дисков на необходимый. Но обратите внимание: ESXi преоб-

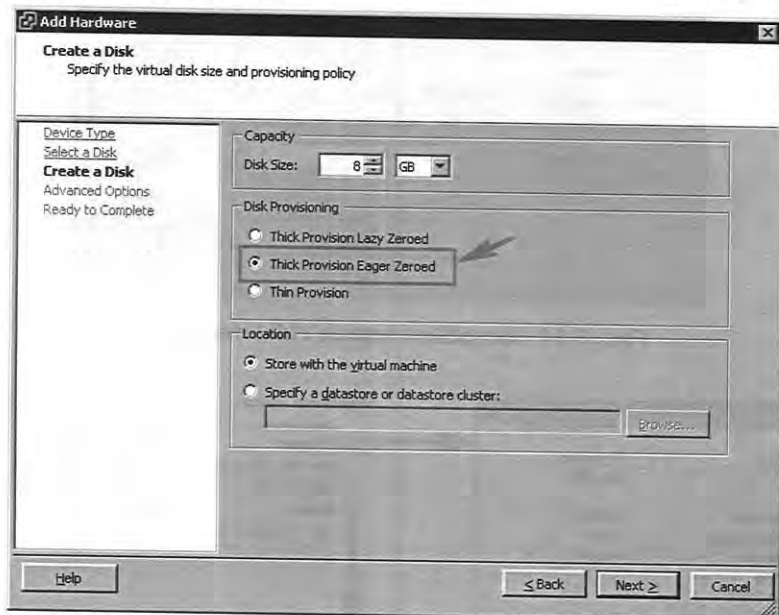


Рис. 7.13. Создание файла vmdk с предварительным обнулением

разует формат диска в необходимый для FT, лишь если вы включили FT для выключенной ВМ.

Настройка инфраструктуры и включение FT

Итак, для включения Fault Tolerance вам необходимо произвести следующие действия:

1. Включить проверку сертификатов серверов.
2. Настроить сеть на каждом сервере.
3. Создать кластер НА, добавить в него сервера и проверить соответствие настроек.

Для включения проверки сертификатов серверов зайдите в меню **Administration** ⇒ **vCenter Settings** ⇒ **SSL Settings** ⇒ отметьте **Check host certificates**.

Под настройкой сети подразумевается следующее: вам нужны два интерфейса VMkernel, один из которых будет использоваться под VMotion, а второй – под трафик Fault Tolerance. Чтобы конфигурация была поддерживаемой, они должны иметь по собственному и выделенному гигабитному сетевому контроллеру, хотя бы по одному.

Таким образом, вам необходимо создать два порта VMkernel, выделить каждому по физическому сетевому контроллеру и расставить флажки (рис. 7.14 и 7.15):

Мною приведен лишь пример конфигурации сети. Разумеется, нет нужды помещать VMotion и FT-интерфейсы VMkernel на один виртуальный коммутатор. Конечно, эти порты могут быть созданы и на распределенном виртуальном коммутаторе.

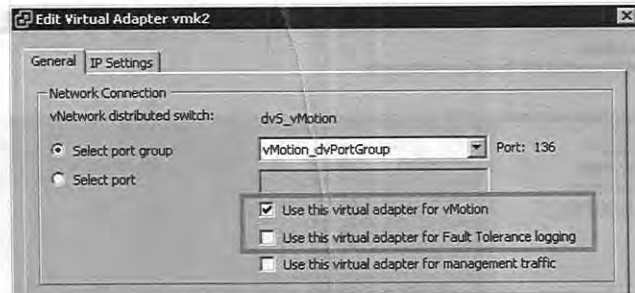


Рис. 7.14. Настройки портов VMkernel для FT

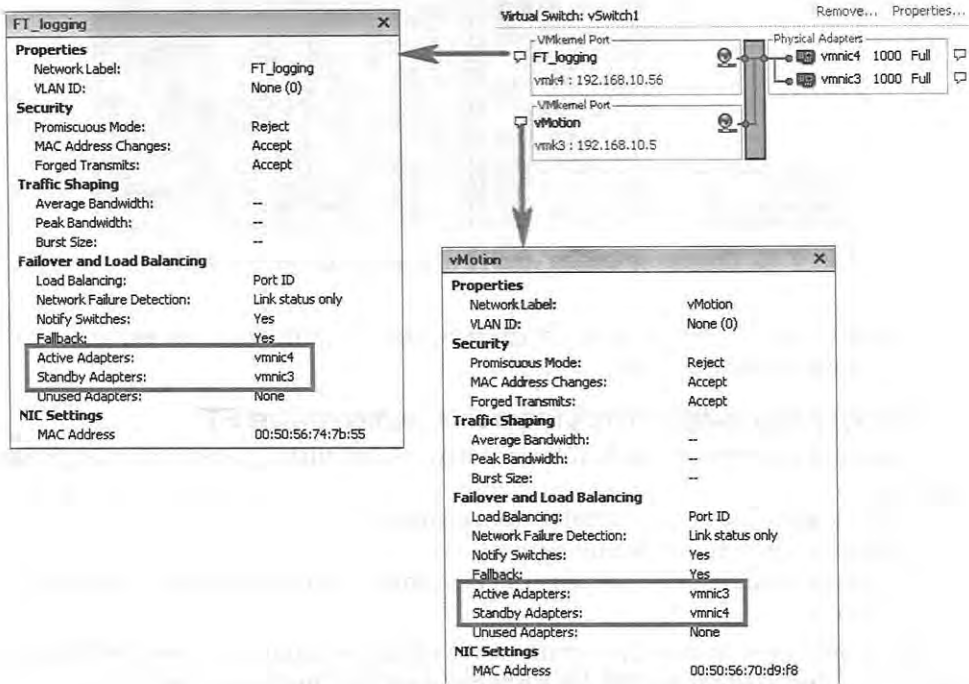


Рис. 7.15. Пример сети для FT

Следующий шаг – сервера должны быть в кластере HA. Если это еще не так – сделайте это сейчас.

Обратите внимание, что на вкладке **Summary** для каждого сервера у вас должно быть указано, что VMotion и Fault Tolerance для него включены (рис. 7.16).

Для проверки соответствия настроек перейдите на вкладку **Profile Compliance** для кластера и нажмите ссылку **Check Compliance Now**. Вам покажут статус серверов относительно кластера (рис. 7.17).

Также у VMware есть специальная утилита под названием **VMware SiteSurvey** (<http://www.vmware.com/support/sitesurvey>). Она способна заблаговременно по-

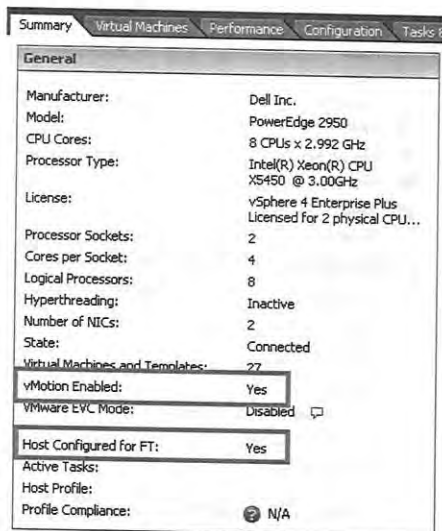


Рис. 7.16. Summary для сервера

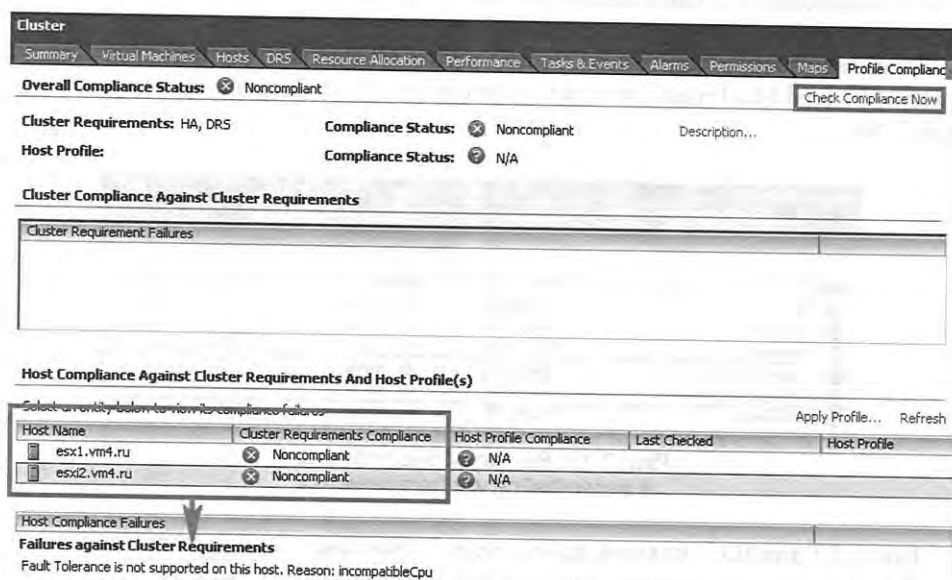


Рис. 7.17. Встроенная проверка кластера на соответствие требованиям FT

казать, годится ли наша инфраструктура для включения FT. Дается расклад как по каждому серверу, так и по каждой VM.

Наконец, мы готовы начать пользоваться благами FT. Для этого выберите в контекстном меню VM пункт **Fault Tolerance** ⇒ **Turn On** (рис. 7.18).

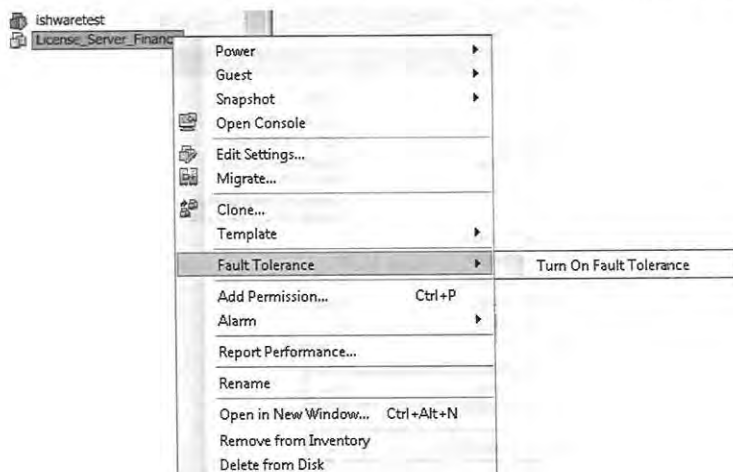


Рис. 7.18. Включение FT

Если в настройках сервера, конфигурации сервера или конфигурации ВМ есть что-то, мешающее FT, вам покажут соответствующее сообщение о невозможности активировать Fault Tolerance для данной ВМ.

Если все прошло нормально, то вы увидите изменение иконки этой ВМ, а на вкладке **Virtual Machines** для кластера увидите и Secondary виртуальную машину (рис. 7.19).

PRI-HA-DRS-Cluster				
Summary Virtual Machines Hosts DRS Resource Allocation Performance Tasks & Events Alarms Permissions				
Name, State, Host or Guest OS contains:				
Name	State	Status	Host	
hot add	Powered On	Normal	vie-esx-pri-02.vmware.local	
ishwaretest	Powered On	Normal	vie-esx-pri-03.vmware.local	
ishwaretest	Powered On	Normal	vie-esx-pri-02.vmware.local	
License_Server_Finance	Powered On	Normal	vie-esx-pri-04.vmware.local	
License_Server_Finance	Powered On	Normal	vie-esx-pri-05.vmware.local	
MS Exchange	Powered Off	Normal	vie-esx-pri-02.vmware.local	

Рис. 7.19. Primary и Secondary ВМ в интерфейсе vSphere клиента

Когда FT для ВМ включен, на ее вкладке **Summary** показывается информация о статусе его работы (рис. 7.20).

Что мы здесь видим:

- **Fault Tolerance Status** – показывает текущий статус защиты ВМ. Варианты – Protected и Not Protected. В последнем случае нам покажут причину:

Fault Tolerance	
Fault Tolerance Status:	Protected
Secondary Location:	vie-esx-pri-05.vmware.local
Total/Secondary CPU:	208 MHz
Total/Secondary Memory:	31.00 MB
vLockstep Interval:	0.008 seconds
Log Bandwidth:	3349 KBps

Рис. 7.20. Информация о статусе FT для ВМ

- **Starting** – FT в процессе запуска Secondary BM;
- **Need Secondary VM** – Primary работает без Secondary. Обычно такое бывает, когда FT выбирает, где включить Secondary. Или в кластере нет сервера, совместимого с сервером, где работает Primary. В последнем случае, когда проблема решена, выключите (disable) и включите FT заново;
- **Disabled** – администратор отключил FT;
- **VM not Running** – Primary BM выключена;
- ☐ **Secondary location** – на каком сервере запущена Secondary BM;
- ☐ **Total Secondary CPU** – сколько мегагерц потребляет Secondary BM;
- ☐ **Total Secondary Memory** – сколько мегабайт памяти потребляет Secondary BM;
- ☐ **vLockstep Interval** – на сколько секунд Secondary BM отстает от Primary. В большинстве случаев это время не превышает половины секунды, обычно меньше 1 мс – но величина этой задержки зависит от сети. В данном примере Secondary BM отстает на 8 тысячных секунды;
- ☐ **Log Bandwidth** – полоса пропускания, сейчас задействованная под трафик FT этой BM.

Для защищенной FT BM доступны следующие операции (рис. 7.21):

- ☐ **Turn Off Fault Tolerance** – выключает FT для BM;
- ☐ **Disable Fault Tolerance** – отключает FT для BM, но сохраняет все настройки, историю и Secondary BM. Используется в случае временного отключения FT для этой BM;
- ☐ **Migrate Secondary** – мигрировать Secondary BM на другой сервер;
- ☐ **Test Failover** – Secondary BM становится Primary и создает себе новую Secondary. Бывшая Primary пропадает;
- ☐ **Test Restart Secondary** – пересоздает Secondary BM.

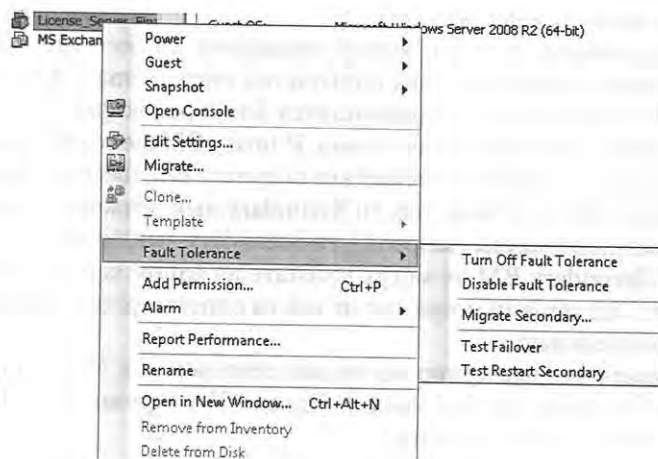


Рис. 7.21. Действия с BM, защищенной FT

Как работает VMware FT

Когда вы включаете FT для VM, vCenter инициирует ее VMotion (то есть копирование содержимого ее оперативной памяти) на другой сервер. Однако VMotion используется не для миграции, а для создания такой же VM на другом сервере, то есть исходная VM не удаляется.

FT можно включить для выключенной VM, тогда при включении виртуальная машина начинает работать сразу на двух серверах.

Выбор сервера для включения Secondary осуществляет DRS (если он включен). И Primary, и Secondary VM можно мигрировать с помощью VMotion, но DRS для этих VM выключен (то есть выдавать рекомендации по их миграции DRS не будет).

Мы получаем две идентичные VM, два идентичных процесса на разных серверах. Исходная VM называется Primary, ее копия – Secondary. События и данные с Primary VM непрерывно реплицируются на Secondary по сети. Эта репликация называется virtual lockstep, или vLockstep. Таким образом, состояние Secondary VM всегда такое же, как у Primary, с отставанием на незначительное время. Файлы vmx и некоторые другие у Secondary свои собственные, но создаются они в том же каталоге, где расположены файлы Primary.

И Primary, и Secondary VM работают с одними и теми же дисками. Вернее, Secondary не работает – ESXi подавляет обращения Secondary VM к дискам и к сети для предотвращения конфликтов. Но когда она станет Primary – она начнет работать с ними. Важно понимать, что это не сказывается на целостности данных, так как единственную копию обновляет Primary VM. Все действия, включая работу с этими данными, выполняются и на Secondary VM тоже. То есть при переходе по отказу Secondary VM получит диск именно в том состоянии, в котором ожидает.

Primary и Secondary VM обмениваются сообщениями пульса (heartbeat). Если сообщения пропали – Secondary немедленно подменяет Primary. Сразу после этого инициируется создание очередной Secondary VM, и спустя короткое время после сбоя данная VM опять отказоустойчива с помощью VMware FT. Выбор сервера для новой Secondary осуществляет HA.

Если сбой произошел в сети между серверами, то Secondary VM перестала получать heartbeat-сообщения. Она попытается стать Primary и получить доступ к файлам VM. Однако из-за сохранившихся блокировок файлов у нее не получится это сделать, и она будет уничтожена. Primary VM же создаст для себя новую Secondary, так как от старой она перестала получать heartbeat-сообщения.

Если Primary VM выключается, то Secondary выключается. Если Primary переводится в состояние паузы (suspend), то Secondary выключается.

Primary и Secondary VM не могут работать на одном сервере, за этим следит Fault Tolerance. Однако они могут числиться на одном сервере в выключенном состоянии – это нормально.

Если отказали одновременно несколько серверов, и в числе них – сервера и с Primary, и с Secondary, то VM упадет. Однако HA перезапустит Primary, после этого FT сделает для нее Secondary.

Если внутри VM случится программный сбой, например BSOD, то он отреплицируется в Secondary VM. FT не защищает от такого рода проблем. Однако в

составе HA есть компонент VM Monitoring, который способен перезапустить зависшие VM.

Secondary VM тратит ресурсы того сервера, на котором работает. То есть после включения FT для какой-то VM она начинает использовать в два раза больше ресурсов. Это – плата за отказоустойчивость. HA учитывает потребляемые Secondary VM ресурсы в своих расчетах резервирования ресурсов. Обратите внимание, что ресурсов для Secondary VM должно быть в достатке. Если их будет не хватать на работу с той же скоростью, что и Primary, то работа Primary VM замедлится! Хорошей аналогией являются две шестеренки на одной цепи – если вторая крутится медленнее, то и первая будет вынуждена замедлиться.

Если частоты процессоров серверов, на которых работают Primary и Secondary VM, отличаются (не рекомендуется разница более чем на 300 МГц), то FT может периодически перезапускать Secondary VM из-за накопившегося отставания. Если перезапуск происходит регулярно, имеет смысл попробовать отключить технологии понижения энергопотребления в BIOS серверов, так как эти технологии могут понижать тактовые частоты. Некоторые источники рекомендуют отключать HyperThreading при траблшутинге FT.

FT-трафик между Primary- и Secondary-узлами содержит в себе процессорные инструкции, выполняемые Primary, и необходимые для этого данные. Этот трафик может быть значителен, особенно когда на сервере работают несколько FT-защищенных VM. Используйте Jumbo frames, задумайтесь об использовании 10 Гбит Ethernet.

Все данные, поступающие на вход Primary VM, пересылаются на Secondary по FT-сети. Это данные, приходящие по сети, и прочитанное с дисков. Если данных много, то нагрузка на FT-сеть большая. В случае интенсивного чтения с дисков и нехватки пропускной способности FT-сети хорошо бы отключить эту пересылку – позволив Secondary VM читать данные непосредственно с дисков VM. Это не рекомендуется VMware, но возможно. Настраивается это следующим образом: в файл vmx выключенной VM добавляется строка

```
replay.logReadData = checksum
```

Для отмены этой настройки выключите VM и удалите эту строку. Подробно-сти см. в базе знаний VMware, статья 1011965.

Старайтесь разносить защищенные FT виртуальные машины по серверам кластера равномерно, для равномерной загрузки сети FT.

Если под трафик FT выделены несколько физических сетевых контроллеров, то с настройкой по умолчанию виртуальные коммутаторы не смогут сбалансировать по ним нагрузку. Трафик FT (то есть репликация состояния всех защищаемых FT виртуальных машин с одного сервера) идет от одного интерфейса VMkernel, от одного MAC-адреса. Чтобы виртуальные коммутаторы смогли балансировать трафик FT по разным каналам во внешнюю сеть, необходимо включить метод балансировки нагрузки по IP-хешу (Route based on IP hash). Этот метод балансировки использует пару IP-источника – IP-получателя для распределения трафика. FT-пары между разными серверами имеют разные пары IP-источник – IP-

получатель, и трафик к разным серверам сможет выходить наружу через разные физические контроллеры. Напомню, что для включения балансировки нагрузки по IP-хешу необходимо настроить etherchannel (802.3ad Link aggregation) на портах физического коммутатора.

Обратите внимание, что включение и отключение FT занимают секунды. Таким образом, если с ВМ необходимо выполнить операцию, для FT-защищенной ВМ невозможную, то можно отключить FT, выполнить операцию, включить FT. Примером такой операции является горячее добавление любого устройства, например диска ВМ.

vCenter обладает несколькими alarm, которые отслеживают события FT (рис. 7.22).

На вкладке **Performance** для Primary ВМ доступна информация по совокупной нагрузке ВМ и ее копии (рис. 7.23).

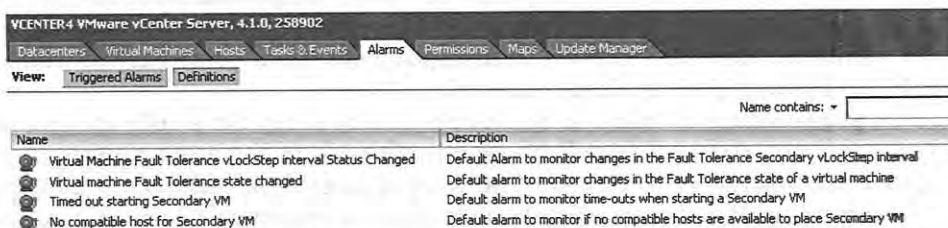


Рис. 7.22. Alarm для мониторинга событий FT

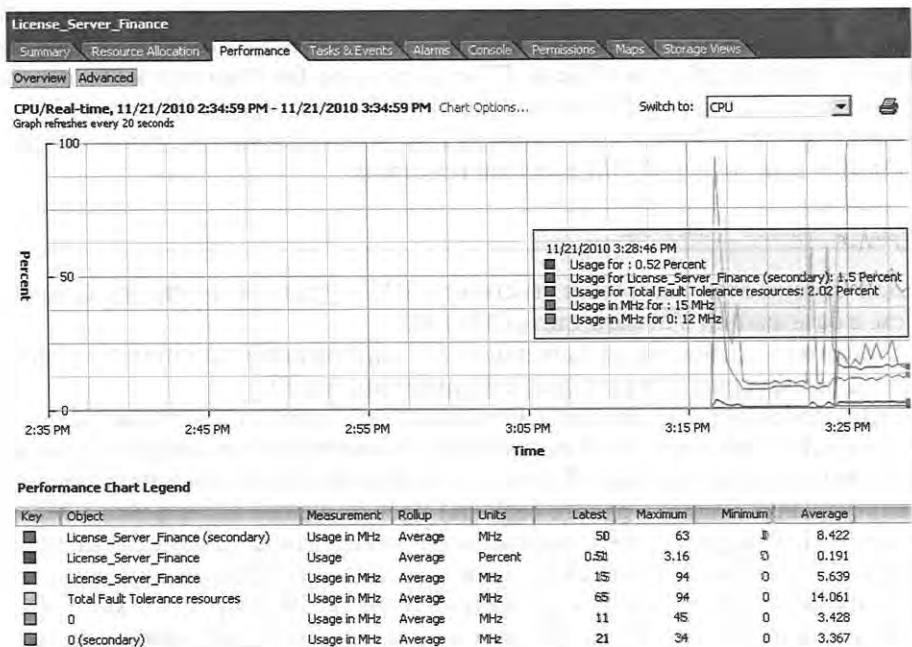


Рис. 7.23. Вкладка **Performance** для FT-защищенной ВМ

7.2. Управление обновлениями виртуальной инфраструктуры, VMware Update Manager

У нас с вами есть несколько путей для обновления серверов ESXi 5. Я упомяну обо всех, но упор сделаю на входящем в состав дистрибутива vCenter средстве VMware Update Manager.

Сначала упомяну о возможности обновлять сервера ESXi из командной строки, однако если вы используете коммерческую версию vSphere и вам доступен VMware Update Manager – очень может быть, что вам не потребуются методы командной строки.

7.2.1. Установка обновлений в командной строке локальной, удаленной и PowerCLI

Для работы с обновлениями без автоматизации при помощи VMware Update Manager нам необходимо вручную найти и загрузить недостающие обновления, а затем установить их.

Для того чтобы обнаружить недостающие обновления, очень полезным инструментом окажется специальный раздел сайта VMware – <http://www.vmware.com/patchmgr/download.portal>. На этом ресурсе вы без труда обнаружите вышедшие обновления и их описания. Вас интересует обновление с максимальным номером сборки (build number). Этот номер для вашего текущего ESXi вы можете обнаружить в клиенте vSphere, выделив сервер и обратив внимание на информацию чуть выше вкладок в правой части окна.

Затем следует скопировать загруженный файл обновления на какое-то из хранилищ, доступное обновляемому серверу или серверам. Часто для копирования проще всего воспользоваться встроенным файлом-менеджером (Browse Datastore) или сторонними файловыми менеджерами (WinSCP\FastSCP). Итак, допустим, вы скопировали загруженный файл обновления (у меня это будет ESXi500-201112001.zip) на хранилище (у меня оно называется SharedLUN1).

Теперь следует определиться с тем, какой интерфейс командной строки мы будем использовать для установки обновления. Доступные варианты:

- ☐ локальная командная строка – всегда доступна. Единственный доступный вариант, если вы используете бесплатную версию ESXi;
- ☐ PowerCLI – по моему мнению, самый удобный интерфейс командной строки для vSphere, в принципе, удобен и для решения обсуждаемой задачи, особенно когда необходимо обновить несколько серверов;
- ☐ vSphere CLI – по сути, альтернатива PowerCLI для Linux-инфраструктур, где PowerCLI установить некуда.

Локальная командная строка

Для установки обновления из командной строки потребуется единственная команда:

```
esxcli software vib install --depot=/vmfs/volumes/SharedLUN1/ESXi500-201112001.zip
```

После большинства обновлений потребуются перезагрузка. После перезагрузки вы должны увидеть, что поменялся номер сборки вашего ESXi.

PowerCLI

Для установки обновления при помощи PowerCLI архив с обновлением необходимо распаковать и распакованным скопировать на хранилище ESXi. Допустим, вы распаковали архив с обновлением в каталог D:\temp\ESXi500-201112001\.

Для копирования этого каталога на хранилище (у меня это SharedLUN1) можно воспользоваться встроенным файловым менеджером (Browse Datastore), сторонними файловыми менеджерами (WinSCP\FastSCP) или средствами PowerCLI:

```
## Сначала подключимся к обновляемому серверу или к vCenter
Connect-VIServer esxi01.vm4ru.local -User root -Password Password

## Скопируем каталог с обновлением на хранилище хоста
$DS = Get-VMHost esxi01.vm4ru.local | Get-Datastore "SharedLUN1"
Copy-DatastoreItem D:\temp\ESXi500-201112001\ $DS.DatastoreBrowserPath -Recurse

## Переведем сервер в режим обслуживания
Get-VMHost esxi01.vm4ru.local | Set-VMHost -State Maintenance
## Установим обновление
Get-VMHost esxi01.vm4ru.local | Install-VMHostPatch -Hostpath "/vmfs/volumes/Shared-
LUN1/ESXi500-201112001/metadata.zip"
## Перезагрузим сервер
restart-VMHost esxi01.vm4ru.local -Confirm:$false
```

Для командлета Install-VMHostPatch доступен параметр -LocalPath, позволяющий указать файлы обновления не на хранилище сервера ESXi, а на диске машины, где работает PowerCLI. Однако такой вариант не рекомендуется к использованию, потому что в подобном случае обновление будет закешировано во временных каталогах ESXi, а большие обновления там не поместятся.

Однако на момент написания командлет Install-VMHostPatch находился в статусе экспериментального. Альтернативой ему может послужить использование того же самого esxcli из-под PowerCLU, но я не смог победить соответствующую командную строку.

vSphere CLI

Для пятой версии vSphere работа в удаленной командной строке осуществляется практически идентично локальной командной строке. Так что мы можем воспользоваться той же самой командой esxcli, просто указав параметром, к какому серверу необходимо подключиться для выполнения команды:

```
esxcli -server esxi01.vm4ru.local -username root software vib install --depot=/vmfs/
volumes/SharedLUN1/ESXi500-201112001.zip
```

7.2.2. VMware Update Manager

В состав дистрибутива vCenter 5 входит VMware Update Manager 5 (VUM). Это средство, с помощью которого вы весьма эффективно сможете управлять обновлениями своей виртуальной инфраструктуры. VUM способен обновлять:

- ☐ сервера ESXi 5 и предыдущих версий;
- ☐ некоторые сторонние компоненты для ESXi, например модуль multipathing EMC PowerPath или распределенный виртуальный коммутатор Cisco Nexus 1000V;
- ☐ гостевые ОС Windows;
- ☐ версию VMware tools;
- ☐ версию виртуального оборудования;
- ☐ Virtual Appliance как Virtual Appliance, а не как связку VM+ОС+приложения.

Обратите внимание: в версии 5 VUM потерял возможность обновлять гостевые ОС и приложения.

Установка VUM

Установка его описана в первой главе книги, да и в любом случае труда не составляет. Установлен он может быть как на тот же сервер, что и vCenter, так и на выделенный сервер (BM). VUM требует БД под хранение метаданных обновлений. В качестве БД может использоваться Microsoft SQL Server или Oracle, а также SQL 2005 Express, идущий в комплекте с vCenter. Если размер инфраструктуры превышает 5 серверов и 50 BM, то VMware рекомендует не использовать в качестве БД SQL Express. Информацию о настройке Oracle и MS SQL Server под нужды VUM см. в документации **vSphere Update Manager 5.0 Documentation** ⇒ **Installing and Administering VMware vSphere Update Manager**.

VMware рекомендует выдать серверу VUM от 2 Гб оперативной памяти и расположить его максимально близко (с точки зрения сети) к серверам ESXi.

Есть три варианта размещения VUM относительно vCenter:

- ☐ VUM и vCenter установлены на один сервер и используют общую базу;
- ☐ VUM и vCenter установлены на один сервер, но используют разные БД. VMware рекомендует этот вариант в инфраструктурах от 300 BM и 30 серверов;
- ☐ VUM и vCenter установлены на разные сервера и разные БД. VMware рекомендует этот вариант в инфраструктурах от 1000 BM и 100 серверов.

Обратите внимание, что если vCenter и VUM установлены в виртуальных машинах, то применение обновлений VMware Tools на эти виртуальные машины следует делать отдельной задачей от обновления прочих. Это связано с тем, что если в процессе выполнения задачи обновления нескольких BM будут обновлены и перезагружены виртуальные машины с vCenter и VUM, то вся задача обновления окажется прерванной. Имеет смысл выделить виртуальные машины vCenter и Update Manager в отдельный каталог в иерархии **Virtual Machines and Templates**, обновления на который применять отдельной задачей.

Для установки понадобится указать FQDN сервера vCenter, учетную запись с административными привилегиями на сервере vCenter (лучше, чтобы это была выделенная учетная запись) и параметры доступа к БД (в случае использования не SQL 2005 Express).

В зависимости от количества версий ESXi в вашей инфраструктуре и типов Virtual Appliance для VUM может потребовать больше или меньше места на хранение обновлений. На странице документации VUM доступен инструмент Sizing Estimator for vSphere Update Manager 5.0 (http://www.vmware.com/support/pubs/vum_pubs.html), он поможет понять, сколько места для обновлений потребуется в ваших условиях.

После установки серверной части Update Manager он регистрирует себя на vCenter, и в клиенте vSphere появляется возможность установить соответствующий плагин. Для этого обратитесь к меню **Plug-ins** (рис. 7.24).

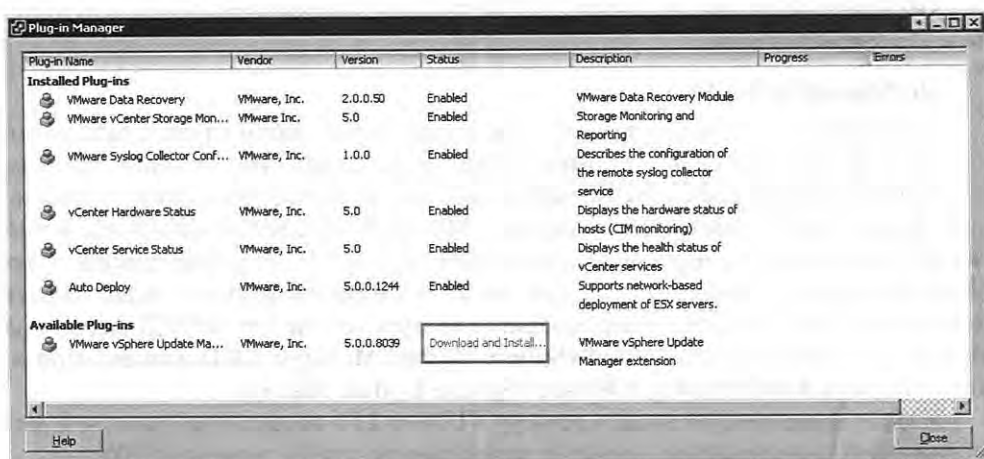


Рис. 7.24. Установка плагина VUM

После успешной установки плагина у вас появятся новые объекты интерфейса в клиенте vSphere:

- ☐ пункт **Update Manager** на странице **Home**;
- ☐ вкладка **Update Manager** для серверов или групп серверов;
- ☐ вкладка **Update Manager** для ВМ или групп ВМ.

Как работает VUM

VUM, как и любые другие средства установки обновлений, работает по следующей схеме:

1. Загружается список всех возможных обновлений.
2. Все или только часть серверов или ВМ сканируются в поисках отсутствующих на них обновлений из этого списка.
3. Недостающие обновления устанавливаются.

Пройдемся по шагам последовательно.

Шаг первый, список всех обновлений

Если вы пройдете **Home** ⇒ **Update Manager**, то попадете в настройки VUM. На вкладке **Configuration** в пункте **Download Settings** настраиваются параметры того, для каких ОС и откуда загружать описание обновлений (рис. 7.25).

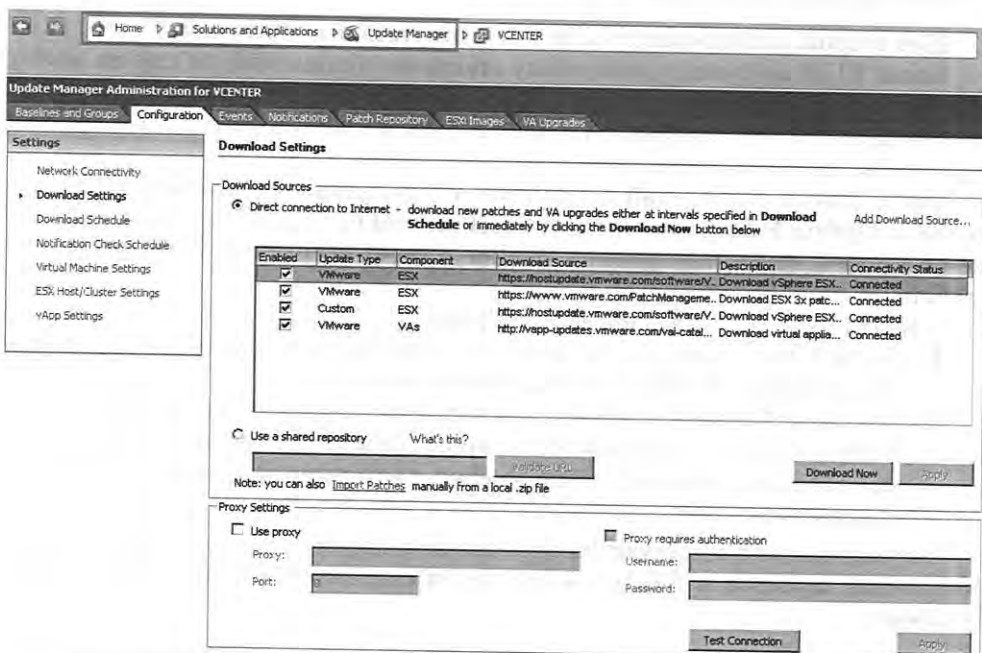


Рис. 7.25. Настройки VUM

Как видите, обновления для ESXi забираются с vmware.com.

VUM загружает описания всех существующих обновлений для ESXi. Сами обновления будут загружаться только перед установкой, и только те, что необходимы.

По ссылке **Add Patch Source** вы можете добавить дополнительный источник. Эта возможность вам понадобится, если какие-то из используемых у вас Virtual Appliance или установленные на ESXi третьесторонние модули будут поддерживать обновления через VUM.

Если с того сервера, куда установлен VUM, есть доступ в Интернет, то он сможет загружать описания и сами обновления самостоятельно. Как видите на рисунке выше, можно указать настройки прокси-сервера.

Расписание поиска новых обновлений в Интернете настраивается в собственном планировщике VUM **Home** ⇒ **Update Manager** ⇒ **Configuration** ⇒ **Download Schedule**.

В соседнем пункте настроек (**Notification Check Schedule**) настраивается оповещение по электронной почте о том, что найдены новые обновления.

Если доступа в Интернет с сервера VUM нет, то можно воспользоваться утилитой командной строки из его состава для загрузки обновлений и последующего их подкладывания VUM. За это отвечает пункт **Use a shared repository**. Утилита называется Update Manager Download Service (umds), подробности про ее использование см. в документации VUM.

Шаг второй, сканирование на наличие отсутствия обновлений

Когда VUM загрузил к себе в базу описания обновлений, он еще не может сканировать на их наличие сервера и ВМ. Необходимо сначала создать список обновлений, на наличие которых хотим проверить, а затем привязать этот список к группе объектов (ВМ/серверов).

Такой список обновлений в терминах VUM называется **baseline**. Пройдите **Home** ⇒ **Update Manager** ⇒ вкладка **Baselines and Groups**. Именно здесь мы можем создавать, изменять и удалять baseline, а также объединять их в группы для удобства.

Обратите внимание на элементы интерфейса (рис. 7.26):

1. **Hosts/VMs** – здесь мы выбираем, на baseline для каких объектов, то есть для серверов или ВМ, мы хотим посмотреть.
2. По ссылке **Create** в левой верхней части мы создаем новый baseline.
3. По ссылке **Create** в правой части мы создаем новую группу baseline. Группа – это всего лишь несколько baseline, с которыми мы затем работаем как с одним объектом.

По умолчанию существуют baseline для всех критичных и всех некритичных обновлений серверов и ВМ и baseline для обновления VMware tools и Virtual Hardware.

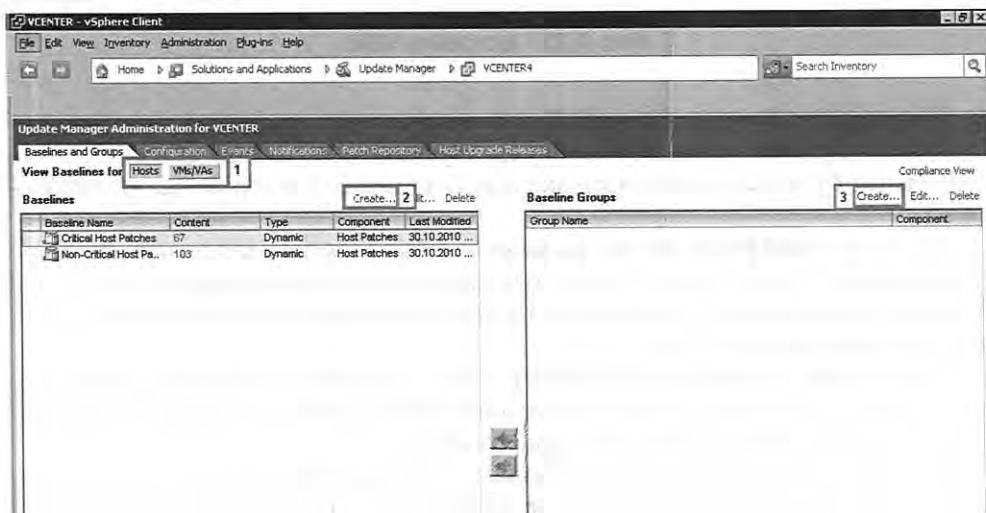


Рис. 7.26. Интерфейс работы с baseline

При создании нового baseline нас спросят:

- ❑ **Baseline Name and Type** – для обновления или апгрейда он нужен, а также для объектов какого типа. Например, создадим baseline для обновления серверов (рис. 7.27);

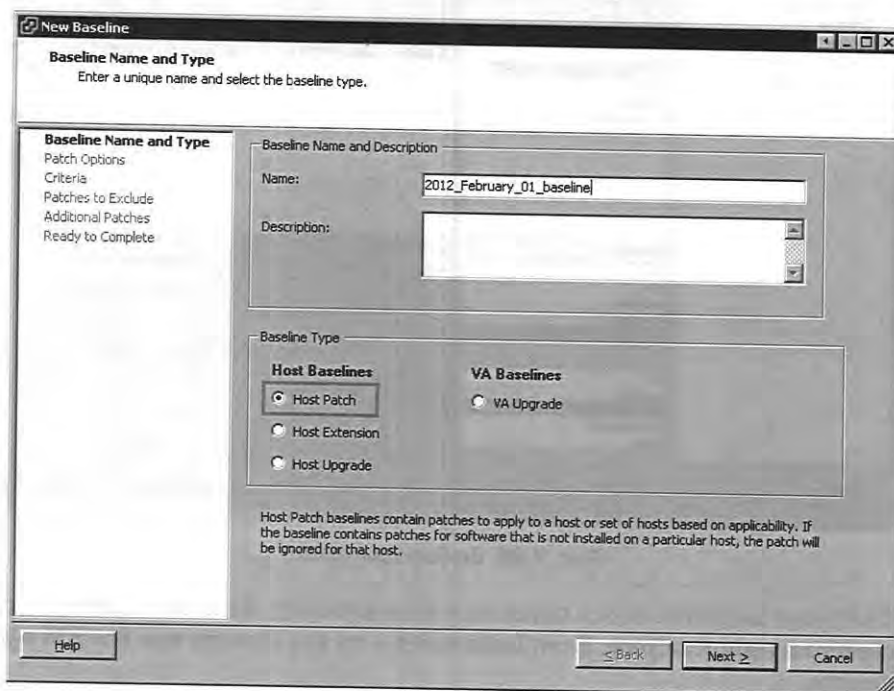


Рис. 7.27. Мастер создания baseline, шаг 1

- ❑ **Patch Options** – будет ли это фиксированный или динамически изменяющийся набор обновлений с указанным критерием. Динамические baseline хороши тем, что вновь вышедшие обновления, удовлетворяющие критериям, попадают в них автоматически;
- ❑ **Criteria** – критерий вхождения обновления в baseline. Им могут быть версия продукта, дата релиза, статус, поставщик обновления (рис. 7.28).

В этом примере я создаю baseline со всеми Critical обновлениями от VMware для серверов ESXi. Флажок **Add or Remove...** позволит добавить или удалить из baseline с динамическим содержанием какие-то конкретные обновления.

Когда baseline создан, его необходимо назначить на объект. В нашем случае – на сервер (хотя практичнее будет на кластер или Datacenter). Для манипуляции с baseline для серверов пройдите **Home** ⇒ **Hosts and Clusters**. Для манипуляции с baseline для ВМ необходимо пройти **Home** ⇒ **Virtual Machines and Templates**.

Пройдя туда, выберите объект, на который хотите назначить baseline. Если это сервер – baseline назначится только на него. Если это каталог с серверами –

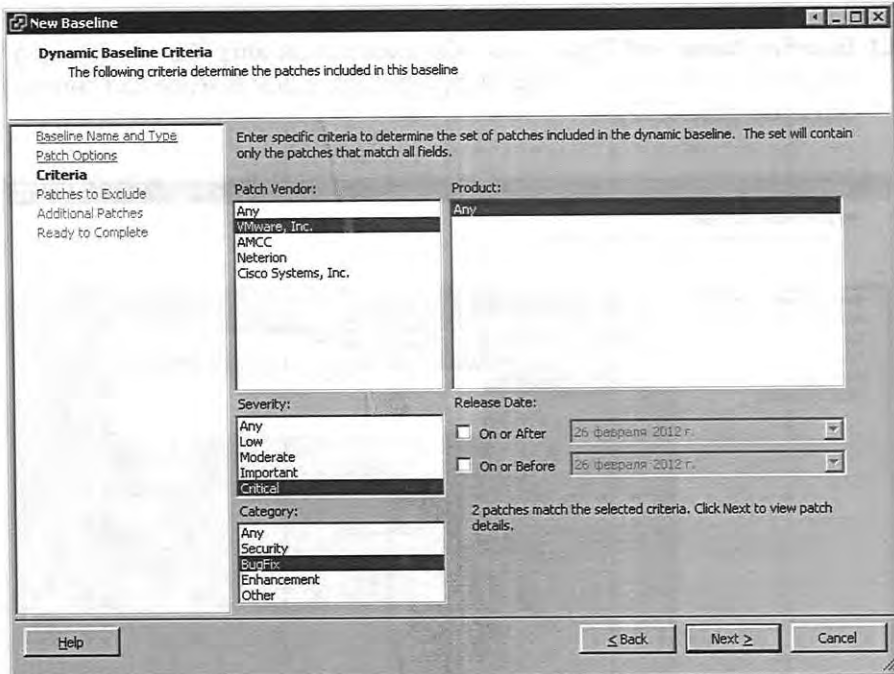


Рис. 7.28. Выбор критерия

baseline будет назначен на все объекты в этом каталоге. Если это кластер – то на все сервера в этом кластере. Если Datacenter – на все сервера или все ВМ в этом датацентре.

Для выбранного объекта перейдите на вкладку **Update Manager**. Вас интересует ссылка **Attach**. Выберите желаемые baseline (рис. 7.29).

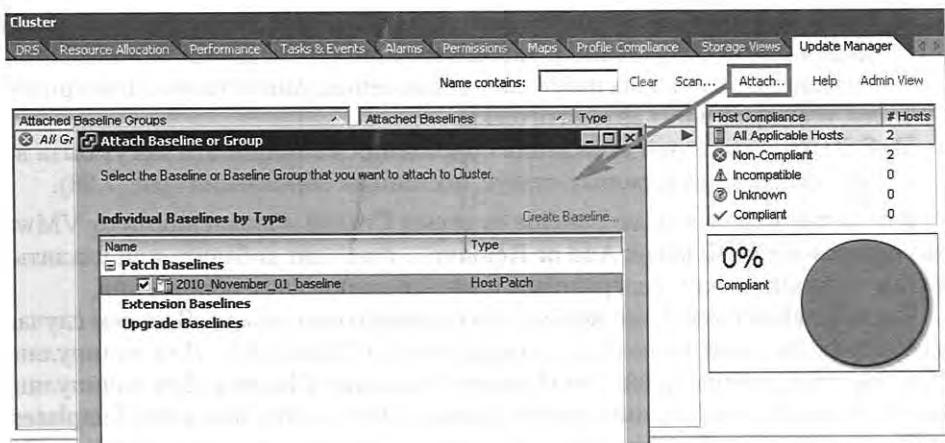


Рис. 7.29. Привязка baseline к кластеру

Разумеется, на каждый объект может быть назначено большое количество baseline и их групп.

Последний шаг – нажать ссылку **Scan** и подождать результатов (рис. 7.30).

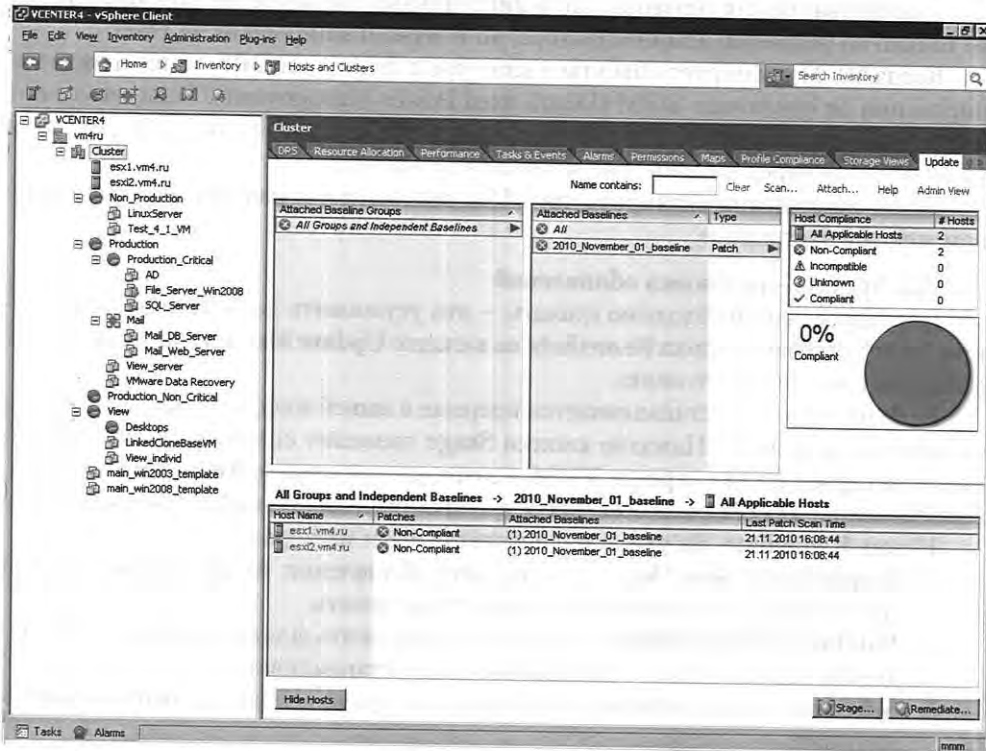


Рис. 7.30. Результаты сканирования серверов кластера

Как видите, оба сервера не обновлены. Справа наверху приводится статистика:

- ❑ **All Applicable** – количество объектов (серверов или VM) всего минус количество объектов, к которым не применимо ни одного обновления из назначенных baseline;
- ❑ **Non-compliant** – сколько из них не удовлетворяют назначенным baseline'ам. «Не удовлетворяют» значит «содержат не все из указанных в baseline обновлений»;
- ❑ **Incompatible** – к скольким объектам не применимы обновления из назначенных baseline;
- ❑ **Unknown** – статус объектов неизвестен. Обычно это значит, что столько объектов еще не сканировалось хотя бы на один baseline;
- ❑ **Compliant** – сколько объектов имеет все указанные в назначенных baseline обновления.

Выбирая эти строки, в нижней части окна будем видеть список серверов или ВМ с данным статусом.

Разумеется, для каждого сервера или ВМ учитываются только применимые обновления.

Сканирование как серверов, так и виртуальных машин может производиться не только по указанию администратора, но и через планировщик vCenter.

Когда VUM сканирует объекты в кластере с функцией DRS, то на время сканирования он отключает DPM (Distributed Power Management). Если на момент запуска сканирования в кластер есть хосты в режиме stand-by, то VUM дожидается их включения перед началом операции сканирования.

После завершения сканирования VUM возвращает в изначальное состояние измененные настройки.

Шаг третий – установка обновлений

Последнее, что необходимо сделать, – это установить недостающие обновления. За это отвечает кнопка **Remediate** на вкладке **Update Manager** или одноименный пункт контекстного меню.

Если обновление устанавливается впервые в вашей инфраструктуре, оно сначала будет загружено. Нажатие кнопки **Stage** позволит сперва загрузить обновление на сервер ESXi с сервера VUM, а затем уже нажимать **Remediate**. Это особенно полезно для обновления серверов, подключенных по медленным каналам.

Нажав **Remediate**, вы увидите следующие шаги мастера:

1. **Remediation Selection** – выбор того, обновления из каких baseline или групп baseline и на какие объекты устанавливать.
2. **Patches and Extensions** – здесь вы можете снять флажок с каких-то обновлений, если не хотите в данный заход их устанавливать.
3. **Schedule** – запустить задачу обновления прямо сейчас или запланировать на позже.
4. **Host Remediation Options** – параметры создаваемой задачи обновления. Здесь вы можете указать:
 - должен ли VUM выключать или помещать в состояние паузы ВМ на обновляемом сервере (если у вас есть DRS и все ВМ подпадают под его действие – это не нужно);
 - количество попыток установки, интервалы между ними;
 - должен ли VUM отключить iso и flr от ВМ на обновляемом сервере. Если ВМ выключатся перед обновлением и включатся после, то подключенный образ iso может помешать нормальному старту ОС.

Если задача обновления запускается для серверов в кластере, то она начнет помещать сервера в режим обслуживания и обновлять по одному. Если при обновлении какого-то сервера произошел сбой – вся задача останавливается. Обновленные к этому моменту сервера остаются обновленными, необновленные – необновленными.

Однако начиная с пятой версии в настройках VUM можно разрешить параллельное обновление серверов одного кластера. Пройдите **Home** ⇒ **Update Mana-**

ger ⇒ ESX Host/Cluster Settings ⇒ флажок **Enable parallel remediation for hosts in cluster**.

Если задача обновления запущена для Datacenter, в котором несколько кластеров, то задача будет работать параллельно, для каждого кластера независимо.

Если в вашей инфраструктуре есть DRS, он автоматически мигрирует все ВМ с входящего в режим обслуживания сервера, и он войдет в режим Maintenance. Если DRS нет или какие-то ВМ мигрировать нельзя, сделайте это сами, чтобы сервер смог войти в режим обслуживания. Также вы можете настроить, чтобы VUM выключал или помещал в состояние паузы ВМ на серверах при переводе в режим обслуживания. Кроме того, на время установки обновлений можно автоматически выключать DPM, HA и FT (рис. 7.31).

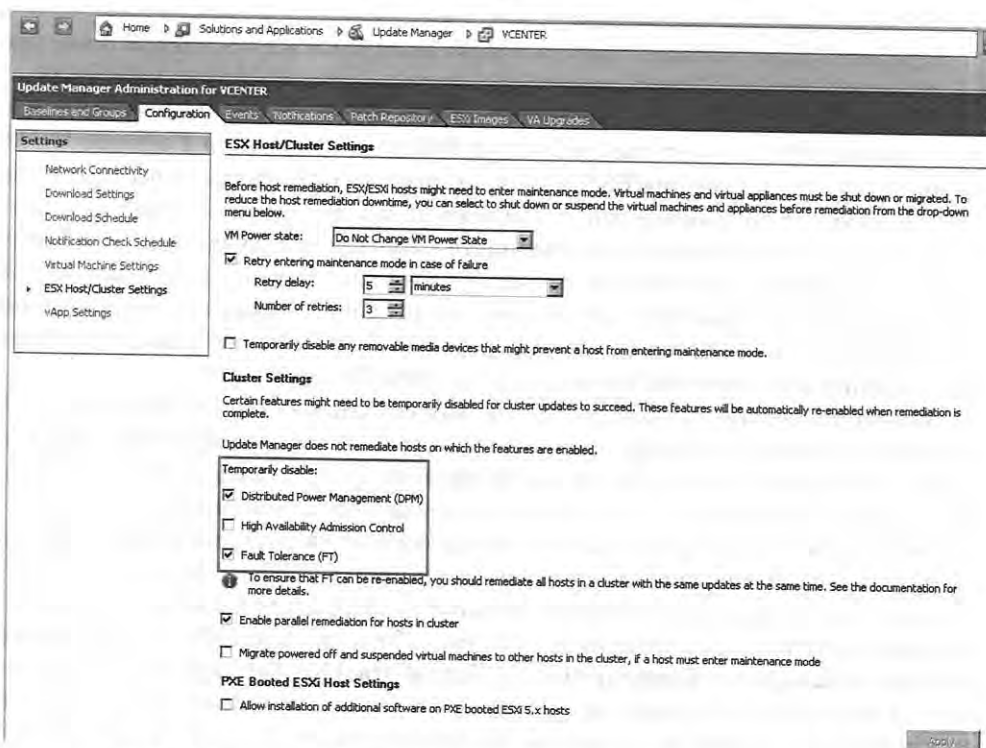


Рис. 7.31. Настройки автоматического временного отключения служб

Эти функции могут помешать обновлению. Если так, то вы можете указать VUM отключать их на период обновления. После завершения процесса Remediate VUM вернет настройки этих функций в изначальное состояние.

Remediation может выполняться как при запуске этой задачи администратором, так и через планировщик vCenter.

vCenter учитывает зависимости обновлений друг от друга, если таковые присутствуют. В частности, это может вылиться в то, что какие-то обновления будут

помечены как «конфликтующие». Происходит это в основном в тех случаях, когда у назначенного обновления есть зависимость от другого обновления, которое еще не установлено и не присутствует в текущем списке устанавливаемых обновлений.

Обратите внимание, что обновления ESXi имеют специфику. Последнее обновление включает в себя все предыдущие. Также напоминаю, что на диске или USB-накопителе с ESXi есть два раздела для хранения его образа. При обновлении VUM формирует образ из активного раздела с ESXi и устанавливаемых обновлений и записывает его в резервный раздел. После этого он помечает активный раздел как резервный, и наоборот. Таким образом, ESXi перезагружается в свою обновленную версию, но в случае неудачного обновления вы можете откатиться на образ с резервного раздела, нажав **Shift+R** при старте сервера.

VUM для виртуальных машин

В работе Update Manager с виртуальными машинами все в основном точно так же, как и с серверами. Но есть и некоторые нюансы, которые я здесь перечислю.

VUM может сканировать выключенные ВМ, ВМ в состоянии Suspend (пауза) и шаблоны. В таком случае диски этих ВМ подмонтируются к серверу VUM по сети (по сети управления ESXi), и он забирает на сканирование необходимые данные. Процесс оптимизирован, VUM обращается только к необходимым блокам, поэтому сканирование в таком варианте возможно и в случае медленной сети с большими задержками. Однако при использовании антивируса на сервер VUM эта оптимизация не работает, так как антивирус будет запрашивать копирование всего файла, а не некоторых блоков. Для решения подобной проблемы рекомендуется добавить в исключения для антивируса путь `\Device\vmstor*`.

Мастер Remediate для ВМ предложит вам создать снимок состояния каждой виртуальной машины. Также, что самое интересное, он предложит вам указать время, через которое этот снимок автоматически будет удален. Это удобно тем, что, с одной стороны, вы защищаетесь от неудачного обновления, а с другой – нет необходимости вручную удалять снимки состояния у многих ВМ или писать соответствующий сценарий. А неудаленные снимки, во-первых, занимают дополнительное место на диске, во-вторых, мешают некоторым операциям. А также некоторым средствам резервного копирования, работе кластера DRS. Пройдя **Home** ⇒ **Update Manager** ⇒ **Configuration** ⇒ **Virtual Machine Settings**, вы можете настроить использование снимков состояния по умолчанию. Когда вы запускаете мастер Remediate вручную, эти настройки можно указать для этой конкретной задачи обновления.

VUM позволяет устанавливать обновления на включенные ВМ, выключенные ВМ, ВМ в состоянии паузы и шаблоны. Имейте в виду: выключенные ВМ будут включены, обновлены, перезагружены и выключены. На время обновления они будут потреблять дополнительные ресурсы.

Шаблоны будут преобразованы в виртуальные машины, обновлены, перезагружены, выключены и преобразованы обратно в шаблоны. Если шаблоны у вас хранятся в «запечатанном» состоянии (например, с помощью `sysprep`) из-за требований обезличивания, такой подход к их обновлению неприменим.

В пятой версии VUM появилась опция обновления VMware tools при следующей плановой перезагрузке.

Напомню, что VMware Update Manager 5 способен помочь вам с обновлением версии виртуального оборудования (пригодится при апгрейде виртуальной инфраструктуры на пятую версию с предыдущих) и с обновлением VMware tools (новые версии VMware tools содержатся в минорных обновлениях для ESXi).

Подведение итогов

С очень большой вероятностью вам будет удобно пользоваться VUM. И обновления ESXi, и обновления VMware tools, и прочие обновления – это важные действия в рамках эксплуатации vSphere.

Скорее всего, хорошей идеей будет назначить baseline вида «Все критичные обновления» на корень иерархии. Не забудьте, что это делается отдельно для серверов (**Home** ⇒ **Inventory** ⇒ **Hosts and Clusters**) и отдельно для виртуальных машин (**Home** ⇒ **Inventory** ⇒ **Virtual Machine and Templates**). Сканирование на соответствие этим baseline поставить на ежедневное или еженедельное расписание. Это позволит вам своевременно обнаруживать отсутствие критических обновлений для вашей инфраструктуры.

К сожалению, VMware не предоставляет средств для автоматического обновления vCenter Server и вспомогательного ПО (VUM, VMware Converter и прочего).

VUM генерирует достаточно большой список событий (events), которые можно отслеживать с помощью механизма alarms для своевременного оповещения об успешных и неуспешных операциях, выполняемых Update Manager'ом. В настройках **Home** ⇒ **Update Manager** ⇒ **Configuration** ⇒ **Patch Download Schedule** штатно предусмотрено оповещение по e-mail о доступности новых обновлений.

7.3. Резервное копирование и восстановление

Когда мы говорим про резервное копирование и восстановление в контексте vSphere, то первое, с чем стоит определиться, – резервное копирование того, что именно нам необходимо.

Варианты:

- ☐ настройки ESXi;
- ☐ настройки и данные vCenter;
- ☐ данные виртуальных машин.

7.3.1. Резервное копирование ESXi и vCenter

Поговорим сначала про инфраструктуру.

Резервное копирование vCenter

Практически все важное для vCenter хранится в базе данных. В случае потери БД vCenter вы потеряете настройки кластеров и пулов ресурсов в DRS-кластерах, а также распределенных виртуальных коммутаторов.

Для осуществления резервного копирования БД vCenter используются обычные средства резервного копирования базы данных используемого у вас типа.

Кроме резервного копирования БД, следует делать копии сертификатов ssl и файла настроек vpxd.conf.

Резервное копирование настроек ESXi

Если поставлена задача осуществлять резервное копирование ESXi, то сделать это можно следующей командой Power CLI:

```
Connect-VIServer vcenter -User administrator -Password password  
Get-VMHost | get-VMHostFirmware -BackupConfiguration -DestinationPath d:\esxibackup
```

Теперь в каталоге d:\esxibackup (он должен быть создан заранее) у вас появятся резервные копии конфигурации ESXi. Если выполнить сценарий так, как он приведен, – резервные копии конфигурации всех серверов ESXi в том vCenter, куда мы подключились в этой posh-сессии.

Восстановить конфигурацию можно вот так:

```
$esxi = get-vmhost esxi01  
Set-VMHost $ESXi.Name -State 'maintenance'  
Set-VMHostFirmware -vmhost $ESXi -Restore -SourcePath "D:\temp\esxibackup\" -HostUser  
root -HostPassword password
```

Если в указанном каталоге хранятся резервные копии настроек нескольких серверов, то будет выбран нужный по имени.

Если изначально сделать выборку не одного сервера ESXi, а нескольких (или всех), будет произведено восстановление конфигурации на всех. Но обратите внимание: восстановление конфигурации требует предварительного перевода сервера в режим обслуживания (maintenance mode), а это значит, что на нем не должно быть включенных ВМ.

Кроме того, для этого резервного копирования существует неофициальный графический интерфейс. См. <http://link.vm4.ru/backup-gui>.

7.3.2. Резервное копирование виртуальных машин

У нас есть несколько подходов к резервному копированию виртуальных машин. Здесь я кратко опишу каждый из них.

Кроме подходов, у нас есть варианты того, резервную копию чего мы хотим сделать.

Типы данных для резервного копирования

Объектами резервного копирования могут быть файлы гостевой ОС и файлы виртуальной машины.

Файлы виртуальной машины. Если мы создадим резервную копию файла vmdk, то создадим резервную копию всех данных ВМ. Такой подход называется

image level backup (но здесь не имеется в виду снятие образа с помощью ПО, запущенного внутри виртуальной машины, такого как Ghost или Acronis).

Условные плюсы:

- ☐ легко настроить резервное копирование, легко восстановить ВМ.

Условные минусы:

- ☐ требуется много места для хранения резервных копий;
- ☐ много данных надо передавать по сети.

Файлы гостевой ОС. Еще мы можем обращаться к данным виртуальной машины на уровне файлов гостевой ОС. Такой подход называют file level backup.

Условные плюсы:

- ☐ можем забирать только часть данных, измененных с последнего сеанса резервного копирования;
- ☐ восстанавливаем только необходимые данные.

Условные минусы:

- ☐ возможно не для всех ОС, обычно только Windows.

Плюсы и минусы здесь «условны» потому, что среди средств резервного копирования наблюдается большое разнообразие по функциям и возможностям. Например, VMware Data Recovery, несмотря на то что резервное копирование осуществляется именно на уровне vmdk-файлов, может осуществлять инкрементальное копирование на уровне измененных блоков этого vmdk.

Обратите внимание, что практически для любых решений резервного копирования актуальна проблема целостности данных (кроме агента в гостевой ОС, который ее решает своими способами). Поясню, в чем дело.

1. Для осуществления резервного копирования работающей ВМ ПО резервного копирования создает снимок состояния (snapshot).
2. Виртуальная машина продолжает работать, но ее файл vmdk теперь не изменяется, все изменения пишутся в delta.vmdk.
3. Основной файл vmdk мы можем копировать или подмонтировать к внешней системе для копирования данных.
4. Однако в момент создания снимка часть файлов гостевой ОС будет открыта и не дописана на диск, в памяти могут висеть незавершенные транзакции и несохраненные данные приложения – и в файле vmdk эти данные в отрыве от данных в ОЗУ ВМ недостаточны, нецелостны, некорректны. А приложения резервного копирования обращаются только к файлу vmdk, не обращаясь к содержимому памяти. Таким образом, резервное копирование без механизмов обеспечения целостности данных, скорее всего, не подойдет для баз данных и вообще любых файлов со сложной структурой, которые читаются и записываются на диск не целиком, а частями. Очень велика вероятность того, что после восстановления такая копия базы данных окажется полностью или частично неработоспособной.

Для решения этой проблемы VMware tools позволяют обратиться к службе Volume Shadow Copy Services, VSS. VSS запрашивает у гостевой ОС и приложений в ней (поддерживающих VSS) создание «контрольной точки» – то есть кор-

ректное сохранение на диск всех данных, которые находятся в работе на текущий момент. В момент «контрольной точки» делается снимок состояния (snapshot), и данные в этом снимке находятся в завершенном и целостном состоянии. Сам процесс создания снимка проходит прозрачно для приложений, так как нормальная работа с диском после создания «контрольной точки» не прерывается. Но все изменения, произведенные после создания «контрольной точки», в снимок состояния уже не попадают. Таким образом, для приложений с поддержкой VSS обеспечивается консистентность на уровне данных приложения.

Для приложений, VSS не поддерживающих, VMware предлагает компонент Sync driver, который обеспечивает целостность данных на уровне гостевой ОС за счет сброса на диск всех буферов операционной системы. Это менее эффективное решение, так как оно не способно обеспечить целостность данных на уровне приложения. Sync driver доступен также не для всех ОС.

Однако VMware tools могут запускать произвольные сценарии до и после создания снимка состояния. Если настроить в этих сценариях остановку и последующий старт приложения, то, особенно вместе с sync driver, можно получить снимок состояния ВМ с целостными данными на диске.

Для Windows ВМ сценарий должен быть расположен в каталоге C:\Program Files\VMware\VMware Tools\backupScripts.d. Запускаться будет первый по алфавиту файл. При запуске данного сценария перед созданием снимка состояния он будет запущен с аргументом «freeze».

Сценарий, отрабатывающий после создания снимка, должен располагаться в том же каталоге и быть последним по алфавиту. При запуске данного сценария после создания снимка состояния он будет запущен с аргументом «thaw» или «freezeFail» в зависимости от того, было ли создание снимка успешным.

Пример такого сценария для резервного копирования ВМ с MySQL:

```
IF "%1%" == "freeze" goto doFreeze
goto postThaw
```

```
:postThaw
IF "%1%" == "thaw" goto doThaw
goto fail
```

```
:doFreeze
net stop mysql
goto EOF
```

```
:doThaw
net start mysql
goto EOF
```

```
:fail
echo "$ERRORLEVEL% >> vcbError.txt
```

```
:EOF
```

В этом сценарии объединены действия перед снимком и после снимка. В указанном каталоге он должен быть единственным, чтобы быть и первым, и последним по алфавиту.

Подробности ищите в документации **VMware Data Recovery 2.0 Documentation** ⇒ **VMware Data Recovery Administration Guide** ⇒ **Understanding VMware Data Recovery**.

Подходы к организации резервного копирования

Теперь поговорим про подходы к организации и способы резервного копирования. Первое, о чем упомянем, – резервное копирование на каком уровне возможно для того или иного подхода.

Обратите внимание на табл. 7.2.

Таблица 7.2. Связь подходов и типов резервного копирования

	Копирование файлов ВМ	Копирование файлов гостевой ОС
Агент в гостевой ОС	–	+
Средства без поддержки vStorage API for Data Protection	+	–
Средства с поддержкой vStorage API for Data Protection	+	+

В заголовке таблицы вы видите подходы, описанные выше. В левом столбце – способы организации резервного копирования, о которых подобно поговорим ниже. На пересечении обозначены возможности того или иного подхода к резервному копированию. Например, при резервном копировании агентом в гостевой ОС в качестве данных для резервного копирования могут выступать лишь файлы гостевой ОС, но не файлы виртуальной машины.

Что такое vStorage API for Data Protection, я опишу далее.

Средства с поддержкой vStorage API отличаются большим разнообразием, поэтому их характеристики в данной таблице немного условны, какие-то конкретные решения могут не поддерживать резервного копирования и file-level, и image-level. Например, VMware Data Recovery осуществляет резервное копирование только на уровне image, а вот восстановление может происходить и на уровне отдельных файлов гостевой ОС.

Агент резервного копирования в гостевой ОС

Мы можем осуществлять резервное копирование виртуальных машин так же, как мы делаем это для наших физических серверов, – установив агент резервного копирования в гостевую ОС.

Плюсы этого подхода:

- ☐ после переноса инфраструктуры в ВМ можно использовать старую схему резервного копирования без изменений;
- ☐ единая схема резервного копирования для всей инфраструктуры – и физической, и виртуальной;

- ☐ осуществлять резервное копирование можно для всего, что поддерживает-ся агентами, при возможности задействовав специфические возможности или особенности приложений.

Минусы:

- ☐ подобная схема не задействует возможности по оптимизации резервного копирования, которые предоставляют виртуализация и vSphere;
- ☐ агенты стоят денег.

Такой тип резервного копирования позволяет осуществлять резервное копирование только на уровне файлов гостевой ОС.

Бесплатные средства или сценарии

Как правило, эту категорию можно еще назвать «Средства, не поддерживающие vStorage API for Data Protection». Из заслуживающего внимания бесплатного ПО обращаю ваше внимание на Veeam FastSCP (графический файловый менеджер) и сценарий GhettoVCB (<http://communities.vmware.com/docs/DOC-8760>).

Плюсы:

- ☐ дешево или бесплатно.

Минусы:

- ☐ могут потребоваться большие усилия для администрирования подобного решения;
- ☐ сценарии придется или написать, или искать. Найденные готовыми сценарии под ваши условия могут не подойти.

С помощью этого варианта вам будет доступно, скорее всего, только резервное копирование на уровне файлов vmdk виртуальной машины.

Средства, поддерживающие vStorage API for Data Protection

Это самая широко представленная категория. Здесь мы поговорим про сторонние средства резервного копирования специально для vSphere, про решение VMware для резервного копирования – VMware Data Recovery и про то, как VMware позволяет виртуальные машины сторонним средствам резервного копирования, не специализирующимся на vSphere.

Решения резервного копирования, специализированные под vSphere. Выбор среди средств резервного копирования именно под инфраструктуру от VMware достаточно обширен. Чтобы было с чего начать, упомяну продукт от самой VMware, разрабатываемый специально для резервного копирования vSphere: VMware Data Recovery.

Плюсы средств этой категории:

- ☐ законченное решение резервного копирования;
- ☐ оптимизировано именно на резервное копирование виртуальных машин;
- ☐ не требует установки агентов в ВМ.

Минусы:

- ☐ отдельное решение, именно для резервного копирования виртуальной среды;
- ☐ стоит денег;
- ☐ не все решения работают с ленточными накопителями.

Впрочем, разнообразие поддерживаемых функций здесь достаточно велико, так что из безусловных плюсов и минусов здесь разве что ненулевая цена.

В зависимости от решения вам будет доступно или резервное копирование только на уровне файлов виртуальной машины, или еще и на уровне файлов гостевой ОС.

Продукт под названием VMware Data Recovery я опишу более подробно далее.

vStorage API for Data Protection – механизмы доступа к ВМ для сторонних средств резервного копирования. vStorage API for Data Protection – это набор программных интерфейсов, который обеспечивает сторонним продуктам резервного копирования доступ к данным виртуальных машин. То есть продукт с поддержкой vStorage API умеет напрямую взаимодействовать с ESXi для осуществления операций резервного копирования. Большинство или все продукты для резервного копирования поддерживают эти интерфейсы «из коробки».

Если вы планируете использовать средства резервного копирования с поддержкой vStorage API for Data Protection, то могут быть какие-то уникальные нюансы в использовании. Например, для сервера резервного копирования могут поддерживаться разные операционные системы, резервное копирование может быть возможно как на уровне файлов ВМ, так и на уровне файлов гостевых ОС – а может быть, и только каким-то одним способом. Так что вынужден посоветовать читать документацию от этих систем. Но в целом суть решений похожа, поэтому я рекомендую ознакомиться с тем, как работает резервное копирование на примере VMware Data Recovery (VDR). У прочих продуктов основа подхода будет та же.

Все средства работают примерно так:

Мы создаем выделенный сервер резервного копирования (Backup Proxy) – виртуальный или физический. Устанавливаем и настраиваем ПО резервного копирования.

При запуске задачи резервного копирования, до начала самого копирования, ПО резервного копирования при помощи vStorage API подмонтирует данные виртуальной машины к этой ВМ. ПО резервного копирования получает доступ к ним так, как будто это локальные данные той ОС, где ПО установлено.

Важным фактом является то, что резервное копирование осуществляется без остановки ВМ. Это достигается за счет того, что в начале процедуры с ВМ производится снимок состояния (snapshot), его суть – зафиксировать состояние ВМ и, в частности, состояние диска. Теперь зафиксированный диск и копируется в резервную копию. После окончания копирования снимок состояния удаляется.

Обратите внимание. К сожалению, есть вероятность некорректного завершения процесса резервного копирования по разным причинам. Следствием этого является тот факт, что снапшоты создадутся, но не удалятся. Это одна из главных причин того, почему обязательно следует автоматизировать мониторинг наличия снапшотов.

Суть vStorage API – в том, что это средство резервного копирования способно удобным образом осуществлять резервное копирование и виртуальных машин. Этим решением резервного копирования могут быть такие продукты, как CA BrightStor ARCserve, Commvault Galaxy, EMC Avamar, EMC NetWorker, HP Data Protector, Symantec Backup Exec, Tivoli Storage Manager, Symantec NetBackup и др.

Плюсы:

- ☐ в некоторых вариантах инфраструктуры и настроек данные резервного копирования могут передаваться по SAN-сети. Это ускорит процесс и снизит нагрузку на локальную сеть;
- ☐ один агент (который может стоить денег) осуществляет резервное копирование всех наших ВМ;
- ☐ возможно применять одно средство резервного копирования для физической и виртуальной инфраструктур;
- ☐ резервное копирование и image level, и на уровне файлов гостевой ОС (в некоторых продуктах это может быть недоступно).

Минусы:

- ☐ не все приложения могут быть защищены таким образом из-за проблем целостности данных при снимке состояния (это актуально для любых подходов к резервному копированию, кроме разве что случая установки агентов внутрь гостевых ОС).

7.3.3. VMware Data Recovery

Этот продукт поставляется в виде Virtual Appliance, виртуальной машины с предустановленными ОС и ПО.

Плюсы:

- ☐ просто внедрить – всего лишь импортировать виртуальную машину;
- ☐ настроек минимум;
- ☐ интерфейс и резервного копирования, и восстановления интегрируется в клиент vSphere;
- ☐ резервное копирование осуществляется на уровне файлов vmdk, но поддерживается инкрементальное резервное копирование на уровне измененных блоков данных;
- ☐ для гостевых ОС Windows и Linux поддерживается восстановление на уровне отдельных файлов гостевой ОС;
- ☐ VDR осуществляет дедупликацию резервных копий;
- ☐ лицензия на DR входит во многие лицензии vSphere.

Минусы:

- ☐ настроек минимум;
- ☐ не умеет работать с ленточными накопителями.

Это решение позволяет осуществлять резервное копирование только на уровне файлов виртуальной машины. Про работу с данным средством будет рассказано чуть ниже.

VDR поставляется в виде Virtual Appliance. Архив с VMware Data Recovery содержит виртуальную машину с Linux и предустановленным программным обеспечением и инсталлятор дополнения (plugin) к клиенту vSphere. Таким образом, для начала использования этого средства резервного копирования вам необходимо сделать три шага:

1. Импортировать в виртуальную инфраструктуру эту виртуальную машину, загруженную с сайта VMware. При необходимости указать настройки IP, воспользовавшись локальной консолью к этой ВМ.
2. Установить дополнение (plugin) Data Recovery в свой клиент vSphere. Именно это дополнение является интерфейсом и к резервному копированию, и к восстановлению.
3. Выполнить минимальную настройку:
 - куда осуществлять резервное копирование;
 - для каких виртуальных машин и в какое время.

Пройдемся по шагам настройки VDR и задачи резервного копирования.

Первоначальная настройка

После установки дополнения должна появиться соответствующая иконка в клиенте vSphere (рис. 7.32).



Рис. 7.32. Иконка Data Recovery в клиенте vSphere

Выбрав эту иконку, вы попадете в интерфейс Data Recovery. Первое, о чем вас попросят, – это указать IP-адрес или имя виртуальной машины Data Recovery. Именно эта ВМ является сервером резервного копирования. IP-адрес этой ВМ можно посмотреть на вкладке **Summary** для нее. Или достаточно выбрать виртуальную машину VMware Data Recovery в списке слева.

В самый первый раз запустится мастер первоначальной настройки. Его шаги:

- ☐ Set vCenter Server Credentials – вы должны указать для VDR пароль для доступа к vCenter;
- ☐ Backup Destinations – хранилища для резервных копий. Здесь вы сможете подключить CIFS-сетевой ресурс или отформатировать диск (если он был ранее подключен к ВМ VDR). Впрочем, эти настройки можно будет выполнить и позже.

Обратите внимание, что виртуальных машин с VDR у вас может быть несколько, а управлять резервным копированием через них вы можете из одного клиента, указывая здесь адреса разных ВМ Data recovery. Например, это вам пригодится для снижения нагрузки на ВМ Data Recovery или когда число виртуальных машин для резервного копирования превышает 100 – один VDR не может обслуживать большее число ВМ, это встроенное архитектурное ограничение. Если серверов VDR у вас несколько, управлять ими вам придется независимо.

Первый из шагов первоначальной настройки – это указание места для хранения резервных копий. Таким местом может быть подключенный к VDR диск (файл vmdk или RDM), а также сетевой ресурс (SMB).

Затем пройдите **Home** ⇒ **Solutions and Applications** ⇒ **VMware Data Recovery** ⇒ вкладка **Configuration** ⇒ **Destinations**.

По ссылке **Add Network Share** можем добавить сетевой ресурс (поддерживается только протокол SMB). Пароль для доступа на этот сетевой ресурс не должен быть пустым.

Если вы хотите размещать резервные копии на диске, сначала добавьте этот диск к VDR, как к обычной виртуальной машине, и он отобразится в настройках VDR (рис. 7.33).

В списке вы видите «Physical disk». На самом деле это подключенный мною к виртуальной машине VMware Data Recovery RDM LUN. Обратите внимание на

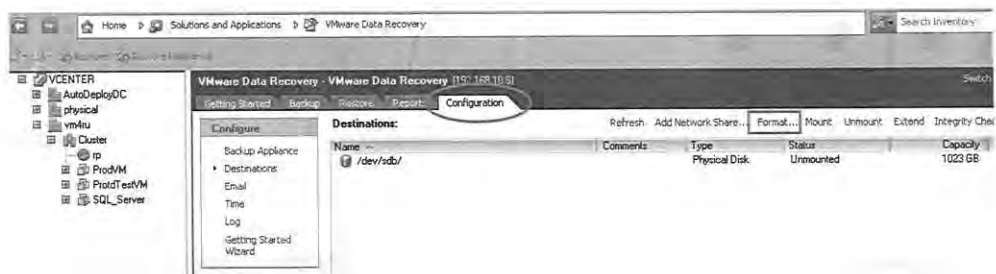


Рис. 7.33. Добавление хранилищ для резервных копий

его статус – Unmounted. Перед использованием этот диск следует отформатировать (ссылка **Format**).

После форматирования статус изменится (рис. 7.34).

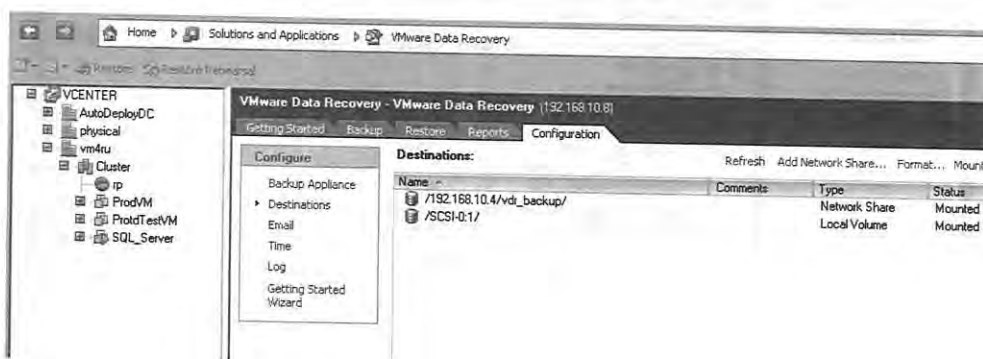


Рис. 7.34. Отформатированный диск и подмонтированный сетевой ресурс

Настройка задания резервного копирования

Создавать и редактировать задания для Data Recovery очень удобно, выбор машин осуществляется простым выделением в дереве. Заодно отображается информация о дате последнего успешного резервного копирования. Пройдите **Home** ⇒ **Solutions and Applications** ⇒ **VMware Data Recovery** ⇒ вкладка **Backup**. Нажав ссылку **New**, вы запускаете мастер задания резервного копирования.

Первый шаг – выбор виртуальных машин с точностью до диска (рис. 7.35).

На шаге **Destination** выбирается хранилище, где будут располагаться резервные копии.

Затем шаг **Backup Windows** – здесь мы задаем время суток для каждого дня недели, в которое VDR может осуществлять создаваемую задачу резервного копирования.

Последняя настройка – **Retention Policy**. Здесь мы указываем срок хранения резервных копий (рис. 7.36).

Например, при настройках с рисунка хранятся 7 последних резервных копий, а также 4 последних недельных бекапа, 2 месячных и один кварталный. Годовой бекап не хранится.

Под недельным понимается резервная копия, сделанная до 23.59 пятницы. Под месячным – последняя, сделанная до 23.59 последнего дня месяца, и т. д.

Обратите внимание. Настройки задач резервного копирования хранятся в том числе на хранилищах резервных копий. Это означает, что если вы удалите старую версию VDR и замените ее новой, а затем подключите к новой VDR старые хранилища резервных копий – то настройки задач резервного копирования будут импортированы. Перед удалением старой версии VDR следует отключить от этой ВМ хранилища резервных копий.

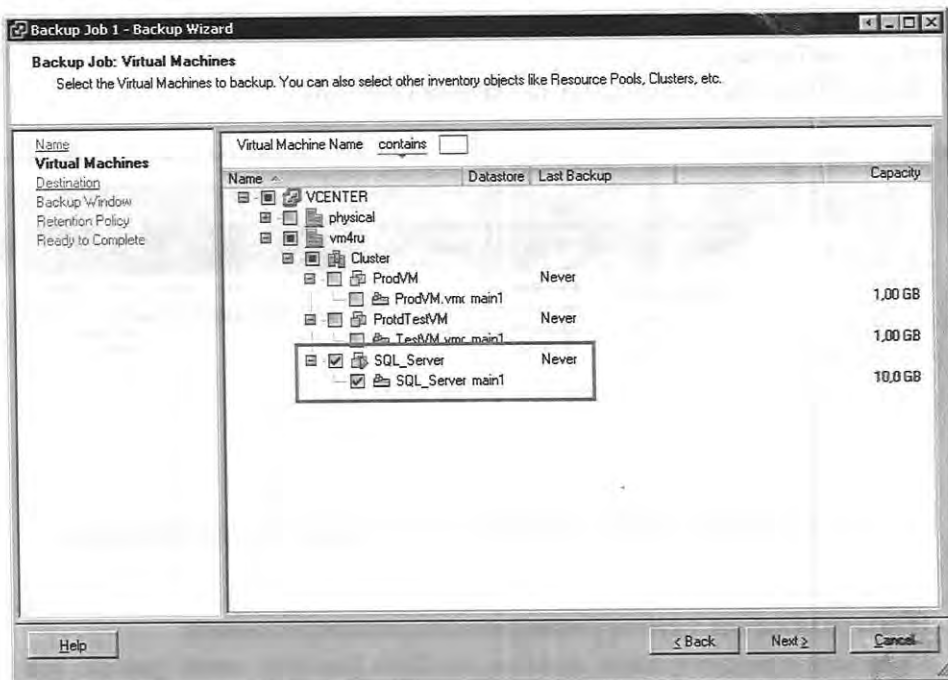


Рис. 7.35. Мастер задания резервного копирования VMware Data Recovery

Восстановление виртуальной машины из резервной копии VMware Data Recovery

Для восстановления виртуальной машины из резервной копии пройдите Home ⇒ Solutions and Applications ⇒ VMware Data Recovery ⇒ вкладка **Restore**. Здесь доступна информация о том, резервные копии на какую дату хранятся в базе (рис. 7.37). Отсюда же запускается процедура восстановления (ссылка **Restore**).

Восстановление файлов гостевой ОС из резервной копии VMware Data Recovery

Если нам удобнее восстановить несколько файлов гостевой ОС, нежели всю виртуальную машину, то VDR даст нам такую возможность в случае гостевых ОС Windows и некоторых Linux. Этот механизм называется VMware File Level Restore (FLR).

Последовательность действий для восстановления отдельных файлов из резервной копии VMware Data Recovery (для гостевых ОС Windows):

1. Копируем клиент VMware File Level Restore (VMwareRestoreClient.exe) в Windows той машины, куда нам надо восстановить файлы из резервной копии.

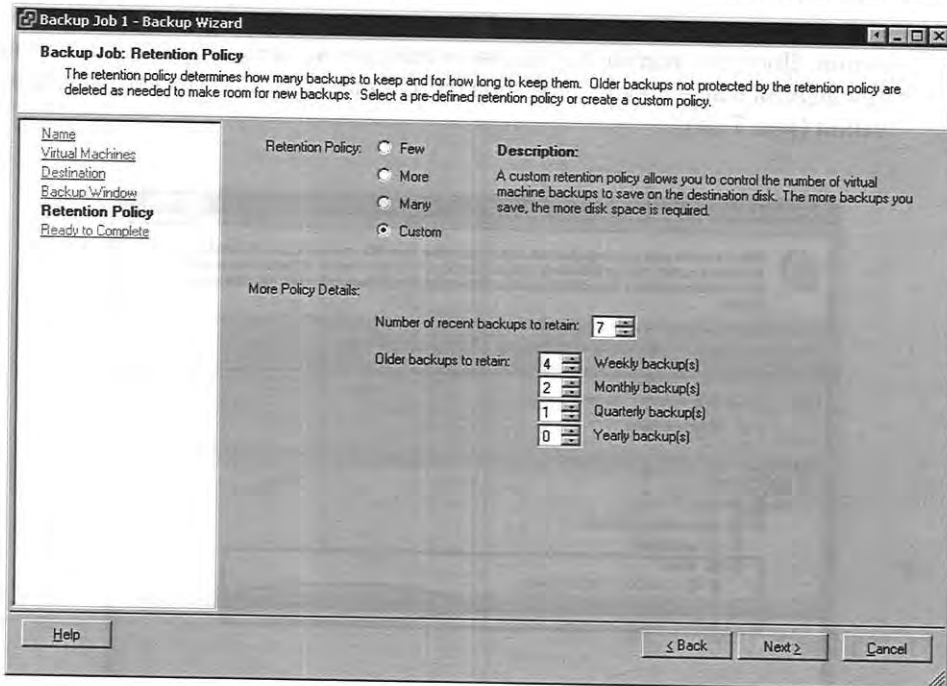


Рис. 7.36. Задание сроков хранения резервных копий

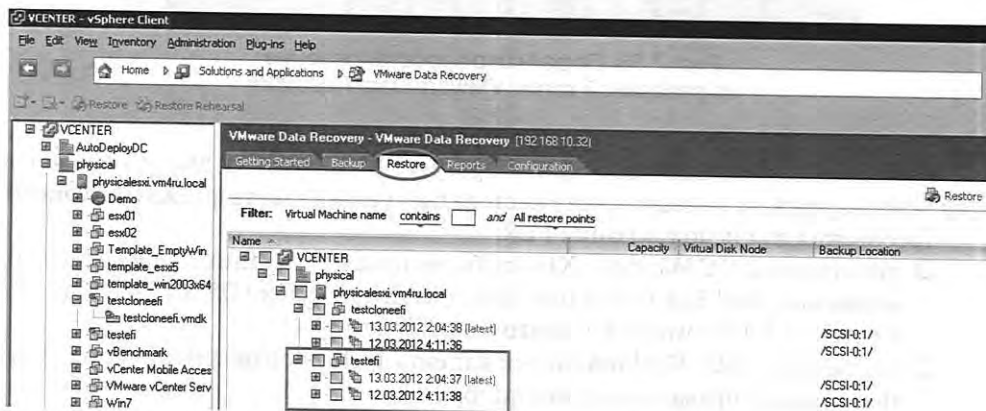


Рис. 7.37. Выбор резервной копии для восстановления

2. Запускаем этот VMwareRestoreClient.exe и указываем IP-адрес виртуальной машины VMware Data Recovery.
3. В появившемся списке точек восстановления виртуальной машины (restore points) необходимо выбрать нужную и нажать **Mount**. Диски виртуальной машины из данной точки восстановления подмонтируются в каталог на

локальном диске. Имя каталога будет совпадать с именем точки восстановления. Доступ к подмонтированным данным возможен только на чтение. Последний шаг – просто скопировать нужные файлы в директорию назначения (рис. 7.38).

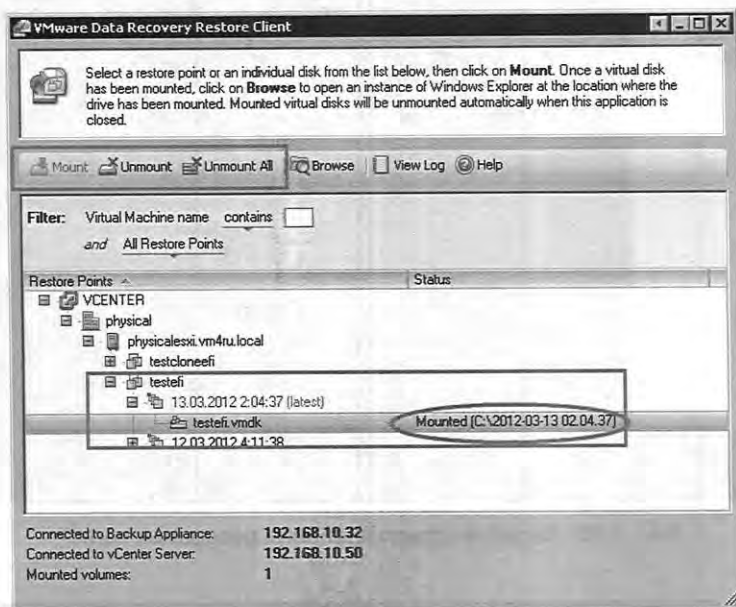


Рис. 7.38. Пофайловое восстановление из резервной копии VMware Data Recovery

Если следует восстановить файлы в другую ВМ, нежели та, чья резервная копия используется, то на первом шаге мастера следует поставить флажок **Advanced**.

Требования для использования FLR:

- ❑ тип гостевой ОС Windows XP или более новая. Для Linux в списке поддерживаемых Red Hat Enterprise Linux (RHEL) 5.4/CentOS 5.4, Red Hat 4.8/CentOS 4.8, Ubuntu 8.04, Ubuntu 8.10, Ubuntu 9.04;
- ❑ на гостевых ОС Windows запуск клиента FLR производится из-под учетной записи с правами администратора;
- ❑ FLR не работает с точками восстановления виртуальных машин, использующих таблицу разделов GUID (GPT).

Факты про VDR

- ❑ Для резервного копирования ВМ создается снимок состояния. ВМ, для которых создание снимка невозможно, не могут быть защищены с помощью VDR. Например, сегодня невозможно создание снимка состояния ВМ, защищенных с помощью FT.

- ❑ Data Recovery использует свой собственный формат хранения резервных копий, что позволяет управлять ими автоматически. Восстановление возможно только из базы Data Recovery и только с помощью Data Recovery.
- ❑ Для хранения резервных копий VDR использует дедупликацию на уровне блоков данных. Отключить ее нельзя.
- ❑ Резервное копирование может осуществляться на SMB сетевые ресурсы, а также на подключенные к виртуальной машине VDR-диски (vmdk или RDM).
- ❑ Резервное копирование осуществляется раз в сутки (ограниченно на это можно повлиять настройкой окна резервного копирования, если хотим делать резервное копирование реже).
- ❑ Одна VDR VM может содержать задания резервного копирования для до 100 VM.
- ❑ Задания резервного копирования выполняются по одному, в пределах одного задания может осуществляться одновременное резервное копирование до 8 VM параллельно.
- ❑ К VDR могут быть подключены несколько хранилищ, каждое до 1 Тб размером. Однако параллельно запись будет вестись не более чем на два. Задания резервного копирования, использующие третье хранилище, будут вставать в очередь. VMware рекомендует, чтобы на хранилище для резервных копий было хотя бы 50 Гб, так как часть места потребуется под технические данные и операции.
- ❑ Резервное копирование выполняется только на уровне целой VM или отдельных дисков, то есть image level backup.
- ❑ Поддерживаются восстановление на уровне VM и восстановление отдельных файлов с помощью отдельной утилиты File Level Restore (FLR). Она доступна и для Windows, и для Linux.

Алфавитный указатель

A

Alarm, 404
AMD-V
 См. Аппаратная поддержка виртуализации процессора
Auto Deploy, 28

B

Ballon, 376
Baseline
 См. Update Manager
Beacon Probing, 150
Browse Datastore
 См. Файловый менеджер

C

CDP
 См. Cisco Discovery Protocol
CHAP, 190
CIM Provider, 21
Cisco Discovery Protocol, 154, 155
Cluster
 См. Кластер
Compatibility Guides
 См. Списки совместимости
CPU
 LCPU, Logical CPU, 362
 pCPU, Physical CPU, 86, 362
 vCPU, Virtual CPU, 278, 362
CPU affinity, 365
CUID Mask, 316, 413
Customization
 См. Обезличивание гостевой ОС
Системы хранения данных, 162

D

DAS
 См. Direct Attached Storage
Datacenter, 65
Data Recovery.
 См. Резервное копирование
Datastore
 См. Хранилище
Datastore Heartbeating
 См. Кластер HA
DCUI
 См. Direct Console User Interface
Direct Attached Storage, 164, 166
Direct Console User Interface, 68, 236
Distributed Power Management
 См. Кластер DPM
Distributed Resource Scheduler
 См. Кластер DRS
Distributed Virtual Switch
 См. Виртуальный коммутатор, распределенный
Distributed vSwitch
 См. Виртуальный коммутатор, распределенный
DPM
 См. Кластер DPM
DRS
 См. Кластер DRS
dvPortGroup
 См. Группа портов распределенного коммутатора
dvSwitch
 См. Виртуальный коммутатор, распределенный

dvUplink, 124
Dynamic binding, 122

E

Enhanced vMotion Compatibility, 425
Ephemeral binding, 122
ESXi, 16, 21
 ESXi Embedded, 21
 ESXi Installable, 21
ESXi Shell, 72
esxtop, 391
EtherChannel
 См. Группировка сетевых контроллеров
EVC
 См. Enhanced vMotion Compatibility
Events, 256
Expandable reservation, 350

F

Failover Order, 148
Fault Tolerance, 458
Firewall
 См. Брандмауэр
Forged Transmits, 146
FT
 См. Fault Tolerance

G

Guest OS
 См. Гостевая ОС

H

HA
 См. Кластер HA
Hardware Compatibility Guides
 См. Списки совместимости
HBA
 См. Host Bus Adapter
HCL
 См. Списки совместимости
High Availability
 См. Кластер HA

Host, 18
Host Bus Adapter, 163
Host Cache, 380
Host Profiles, 244

I

Image Builder, 36
Intel-VT
 См. Аппаратная поддержка виртуализации процессора
IQN, 189
iSCSI, 184
 Инициатор iSCSI, 185
Isolation response
 См. Кластер HA

J

Jumbo Frames, 156, 283

L

LACP
 См. Группировка сетевых контроллеров
Large Ring Sizes, 284
Limit
 Limit для дисков, SIOC, 356
 Limit для памяти, 343
 Limit для процессора, 341
 Limit для сети, NIOC, 361
Link aggregation
 См. Группировка сетевых контроллеров
Linked Mode
 См. vCenter Server Linked Mode
Link Layer Discovery Protocol, 155
LLDP
 См. Link Layer Discovery Protocol
Load Balancing
 См. Группировка сетевых контроллеров
Lockdown mode, 68
Logs
 См. Журналы

LUN, 163
 LUN Masking
 См. Маскировка
 LUN Presentation
 См. Маскировка

M

MAC Address Changes, 146
 Memory Compression, 378
 Memory Overcommitment, 367
 MSI-X, 285
 MTU
 См. Jumbo Frames
 Multipathing, 175, 177
 iSCSI multipathing, 191
 MPP, 179
 NMP, 179
 PSA, 178
 PSP, 179
 SATP, 179

N

NAS
 См. Network Attached Storage
 Netflow, 125
 Network Attached Storage, 164, 167
 Network IO Control, 360
 Network vMotion, 114
 NFS
 См. Network Attached Storage
 NIC
 См. Физический сетевой контроллер
 NIC Teaming
 См. Группировка сетевых контроллеров
 NIOC
 См. Network IO Control
 NPV, 209
 NTP, 67
 NUMA, 366

P

p2v
 См. VMware Converter

Ports group
 См. Группа портов виртуального коммутатора
 PowerCLI, 76
 PowerShell
 См. PowerCLI
 Private VLAN, 143
 Profile Driven Storage, 216
 Promiscuous Mode, 146
 PuTTY, 81
 PVLAN
 См. Private VLAN

R

Raw Device Mapping, 206, 311
 Physical RDP, 312
 Virtual RDM, 312
 RDM
 См. Raw Device Mapping
 Reservation
 Reservation для памяти, 342
 Reservation для процессора, 340
 Resource pool
 См. Пул ресурсов
 resxtop
 См. esxtop
 RSS, 285
 RVtools, 84

S

SAN
 См. Storage Area Network
 SC
 См. Servis Console
 Scheduling Affinity
 См. CPU Affinity
 SCSI Bus Sharing, 290
 SDRS
 См. Storage DRS
 Shares
 Shares для дисков, SIOC, 357
 Shares для памяти, 345
 Shares для процессора, 342
 Shares для сети, NIOC, 361

SIOC*См.* Storage IO Control**Snapshot***См.* Снимок состояния**SNMP, 252**

Software Depot, 37

SSH, 72, 81

SSL, 243

Standalone port, 124, 159

Static binding, 122

Storage Area Network, 164, 172

Storage DRS*См.* Кластер Storage DRS

Storage IO Control, 356

Storage vMotion, 411

SVMotion*См.* Storage vMotion

Syslog Collector, 258

T

TCP Segmentation Offloading, 283

Teaming*См.* Группировка сетевых контроллеров**Template***См.* Шаблон ВМ

Thin disk, 301

TPS*См.* Transparent Page Sharing

Traffic shaping, 147, 360

Transparent Page Sharing, 370

TSO*См.* TCP Segmentation Offloading**U**

Update Manager, 471

Baseline, 474

USB, 293

V**v2v***См.* VMware Converter**VAAI***См.* vSphere API for Array Integration**VADP***См.* vStorage API for Data Protection

vApp, 337

VASA*См.* vSphere API for Storage

Awareness

VCB*См.* Consolidated Backup

vCenter Server, 46

vCenter Server Appliance, 54

vCenter Server Linked Mode, 51

vCMA*См.* vCenter Mobile Access**vCSA***См.* vCenter Server Appliance**VDR***См.* Резервное копирование

VIB, или vSphere Installation

Bundle, 36

Virtual Storage Appliance, 221

VLAN, 136

vMA*См.* vSphere Management Assistant

VMCI, 280

VMDirectPath, 294

vmrk*См.* Файлы виртуальных машин**VMFS***См.* Хранилище**vmk***См.* Виртуальный сетевой

контроллер VMkernel

VMkernel, 18

См. Гипервизор

VMkernel Swap, 378

VM Monitoring

См. Кластер НА**vmnic***См.* Физический сетевой контроллер

vMotion, 412

VMSafe, 229

VMware Converter, 17

VMware Data Recovery

См. Резервное копирование

VMware Dump Collector, 41
VMware tools, 333
vmx
 См. Файлы виртуальных машин
vNIC, 281
vNUMA, 366
vRAM, 66
VSA
 См. Virtual Storage Appliance
vShield Zones, 229
vSphere, 16
vSphere API for Array Integration, 214
vSphere API for Storage
 Awareness, 220
vSphere CLI, 78
vSphere Client, 44
vSphere Management Assistant, 78
vSphere Web Client, 69
vStorage API for Data
 Protection, 485, 486
vSwitch
 См. Виртуальный коммутатор
vswp
 См. Файлы виртуальных машин
VUM
 См. Update Manager

W

Web Client
 См. vSphere Web Client
World Wide Name, 183
WWN
 См. World Wide Name

Z

Zipped memory
 См. Memory Compression
Zoning
 См. Зонирование

A

Аппаратная поддержка виртуализации
 процессора, 88

Б

Брандмауэр ESXi, 231

В

Веб-интерфейс
 См. vSphere Web Client
Виртуальный коммутатор
 Распределенный виртуальный
 коммутатор, 113
 Стандартный виртуальный
 коммутатор, 110
Виртуальный сетевой контроллер
 VMkernel, 106
Время ESXi, 261

Г

Гипервизор, 16
Гостевая ОС, 265
Группа портов виртуального
 коммутатора, 110
Группа портов распределенного
 коммутатора, 122
Группировка сетевых
 контроллеров, 148

Д

Датацентр
 См. Datacenter
Документация, 17

Ж

Живая миграция
 См. vMotion;
 См. также Storage vMotion
Журналы, 257

З

Зонирование, 183

К

Канал во внешнюю сеть
 См. Физический сетевой контроллер

Кластер

Кластер DPM, 429

Кластер DRS, 418

Кластер HA, 437

Admission Control XE

См. Кластер HA

VM Monitoring, 454

Изоляция, 450

Кластер Storage DRS, 433

Консоль VM, 61

Л

Логи

См. Журналы

Локальная командная строка

См. ESXi Shell

Локальная консоль

См. ESXi Shell

М

Маскировка, 183

О

Обезличивание гостевой ОС, 271

П

Пул ресурсов, 349

Р

Распределенная группа портов

См. Группа портов распределенного коммутатора

Распределенный вКоммутатор

См. Виртуальный коммутатор, распределенный

Резервное копирование, 481

VMware Data Recovery, 488

Резервное копирование ESXi, 482

Резервное копирование vCenter, 481

Резервное копирование VM, 482

С

Снимок состояния, 325

Списки совместимости, 18

СХД

См. Системы хранения данных

Т

Тонкий диск

См. Thin disk

Ф

Файловый менеджер

Veeam FastSCP, 83

WinSCP, 83

Встроенный файловый менеджер, 64

Файл подкачки гипервизора

См. VMkernel swap

Файлы виртуальных машин, 317

vmdk, 319

vmx, 318

vswp, 319

Физический сетевой контроллер, 104

Х

Хранилище, 166

NFS-хранилище, 169

VMFS-хранилище, 195

Ш

Шаблон VM, 269

Книги издательства «ДМК Пресс» можно заказать в торговом-издательском холдинге «АЛЪЯНС БУКС» наложенным платежом, выслать открытку или письмо по почтовому адресу: 123242, Москва, а/я 20 или по электронному адресу: orders@alians-kniga.ru.

При оформлении заказа следует указать адрес (полностью), по которому должны быть высланы книги; фамилию, имя и отчество получателя. Желательно также указать свой телефон и электронный адрес.

Эти книги вы можете заказать и в интернет-магазине: www.alians-kniga.ru.

Оптовые закупки: тел. (499) 725-54-09, 725-50-27; электронный адрес books@alians-kniga.ru.

Михеев Михаил Олегович

Администрирование VMware vSphere 5

Главный редактор	Мовчан Д. А.
	dm@dmk-press.ru
Корректор	Синяева Г. И.
Верстка	Чаннова А. А.
Дизайн обложки	Мовчан А. Г.

Подписано в печать 10.10.2012. Формат 70×100 1/16 .

Гарнитура «Петербург». Печать офсетная.

Усл. печ. л. 47,25. Тираж 1000 экз.

Веб-сайт издательства: www.dmk-press.ru

Отпечатано в цифровой типографии «Буки Веди» на оборудовании Konica Minolta
ООО «Ваш полиграфический партнер», ул. Ильменский пр-д, д. 1, корп. 6
Тел.: (495) 926-63-96, www.bukivedi.com, info@bukivedi.com



Softline на страже облаков. Обеспечим. Настроим. Поддержим.



Консультации и помощь
в подборе оптимального решения
виртуализации



Аудит физической и виртуальной
инфраструктуры



Комплексные проекты
по внедрению решений



Аренда инженера-консультанта



Техническая поддержка
(на русском языке 24x7)



Обучение и сертификация



Аренда лицензий VMware
(программа VSPP)



Softline – ведущая международная компания в области лицензирования программного обеспечения и предоставления полного спектра IT-услуг – технической поддержки, IT-аутсорсинга, обучения, юридической поддержки, консалтинга, облачных решений.

Softline является партнером VMware – VMware Premier Partner, обладает статусом авторизованного учебного центра (VATC) и является первой российской компанией-обладателем статуса VSPP Aggregator по программе VMware Service Provider Program, которая дает право партнерам VSPP оказывать услуги на базе ПО VMware своим заказчикам.

softline®

Services Software Cloud

8 495 232-00-23
8 800 100-00-23

vmware@softline.ru
virtual.softline.ru



АДМИНИСТРИРОВАНИЕ СЕРВЕРА



Книга посвящена вопросу работы с семейством продуктов VMware vSphere 5. В книге рассмотрены самые разнообразные моменты, с которыми можно столкнуться при работе с продуктом: здесь вы встретите описание требований и возможностей продуктов VMware, варианты настроек, необходимую для работы с продуктом информацию, в том числе из смежных областей знаний. Материал книги подается в виде пошаговых инструкций с достаточно подробной детализацией.

Издание будет полезно как начинающим, так и опытным системным администраторам. Последние могут использовать книгу как справочное пособие, позволяющее оперативно уточнить нюансы работы тех или иных механизмов, найти необходимые параметры и команды командной строки.

**Автор книги - Михеев Михаил Олегович**

Окончил Казанский Государственный Университет, факультет вычислительной математики и кибернетики. В 2005 году начал чтение ИТ курсов в учебном центре Микроинформ, и практически сразу же начал заниматься направлением VMware. Кроме чтения курсов ведет независимый блог, посвященный виртуализации – <http://vm4.ru>. Является одним из лидеров русскоязычного сообщества VMware (VMUG), организатором регулярных встреч ИТ-специалистов для обмена опытом. Удостоен от VMware звания VMware vExpert.

Интернет-магазин:

www.dmk-press.ru

Книга - почтой:

orders@alians-kniga.ru

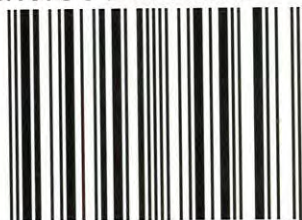
Оптовая продажа:

"Альянс-книга"

Тел.: (499)725-5409

books@alians-kniga.ru

ISBN 978-5-94074-569-3



9 785940 745693 >

DMK
ИЗДАТЕЛЬСТВО

УРОВЕНЬ
ПОЛЬЗОВАТЕЛЯ

- начинающий
- средний
- опытный
- профессиональный

Категория: Виртуализация