

Михаил Михеев

Второе
издание
бестселлера!

Администрирование VMware vSphere 4.1

Виртуализация для профессионалов

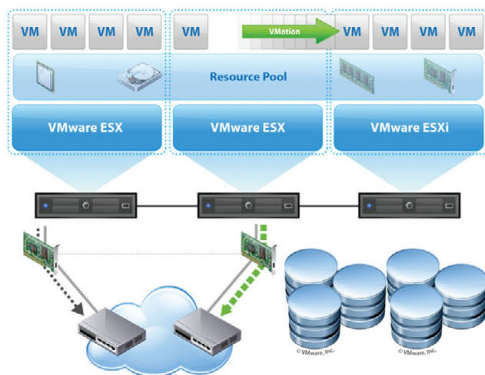
Настройка сети виртуальной инфраструктуры

Системы хранения данных

Управление ресурсами сервера

Мониторинг достаточности ресурсов

Защита данных и доступность виртуальных машин



УДК 32.973.26-018.2
ББК 004.4
М69

Михеев М. О.

М69 Администрирование VMware vSphere 4.1. – М.: ДМК Пресс, 2011. – 448 с.: ил.
ISBN 978-5-94074-658-0

Новое издание книги посвящена работе с семейством продуктов последней версии VMware vSphere 4.1

В ней рассмотрены установка vSphere, настройка сети виртуальной инфраструктуры, системы хранения данных, виртуальные машины, управление ресурсами сервера, защита данных в виртуальных машинах. Кроме того, приводятся сведения о принципах работы, способах мониторинга и диагностики неполадок. Наконец, дается информация по дополняющим сторонним продуктам, которые могут помочь в работе или решении возникающих перед администратором проблем. Материал книги подается в виде пошаговых инструкций с подробной детализацией.

Издание будет полезно как начинающим, так и опытным системным администраторам, которые могут использовать книгу как справочник по параметрам и командам VMware vSphere.

УДК 32.973.26-018.2
ББК 004.4

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 978-5-94074-685-0

© Михеев М. О., 2011
© Оформление, ДМК Пресс, 2011



Содержание

Введение	8
Для кого?	8
О какой версии продукта?	8
Как книга организована?	8
Обратная связь	10
Предисловие	11
Глава 1. Установка vSphere	12
1.1. Обзор	12
1.2. Установка и начало работы с ESX(i)	13
1.2.1. Чем отличаются ESX и ESXi	13
1.2.2. До установки	15
1.2.3. Установка ESXi	18
1.2.4. Установка ESX	21
1.2.5. Автоматическая установка ESX	26
1.2.6. Автоматическая установка ESXi	32
1.3. Начало работы	33
1.3.1. Начало работы без vCenter	33
1.3.2. Установка и начало работы с vCenter Server	35
1.4. Интерфейс клиента vSphere, vCenter, ESX(i). Веб-интерфейс	41
1.4.1. Элементы интерфейса клиента vSphere при подключении к vCenter	41
1.4.2. Первоначальная настройка vCenter и ESX(i)	48
1.4.3. Работа через веб-интерфейс	53
1.5. Обновление ESXi, ESX и vCenter с версий 3.x	55
1.5.1. Обновление до vCenter Server 4 и Update Manager 4	56
1.5.2. Обновление ESX(i) с помощью Update Manager	59
1.5.3. Обновление виртуального оборудования VM и VMware tools	60
1.5.4. Установка обновлений из командной строки	62
1.5.5. Отмена обновления ESX 3.x на ESX 4	63
1.6. Основы работы из командной строки	64
1.6.1. Локальная командная строка ESX, SSH	64
1.6.2. Локальная командная строка ESXi, SSH	66
1.6.3. vSphere CLI, работа с vMA	67
1.6.4. Полезные команды	69

1.6.5. Полезные сторонние утилиты	70
1.7. Сайзинг и планирование	74
1.7.1. Процессор	75
1.7.2. Память	79
1.7.3. Дисковая подсистема	79
1.7.4. Сетевая подсистема	83
1.7.5. Масштабируемость: мало мощных серверов или много небольших?	85

Глава 2. Настройка сети виртуальной

инфраструктуры	88
2.1. Основы сети ESX(i), объекты виртуальной сети	88
2.1.1. Физические сетевые контроллеры, vmnic	91
2.1.2. Виртуальные контроллеры Service Console и VMkernel	93
2.2. Стандартные виртуальные коммутаторы VMware – vNetwork Switch	98
2.3. Распределенные коммутаторы –vNetwork Distributed Switch, dvSwitch. Настройки	101
2.3.1. Основа понятия «распределенный виртуальный коммутатор VMware»	102
2.3.2. Добавление сервера в dvSwitch, настройки подключения vmnic	105
2.3.3. Группы портов на dvSwitch, добавление интерфейсов Service Console и VMkernel	109
2.3.4. Уникальные настройки dvSwitch	111
2.3.5. Уникальные настройки портов dvSwitch: Miscellaneous и Advanced	112
2.3.6. Миграция со стандартных виртуальных коммутаторов на распределенные	113
2.3.7. Технические особенности распределенных виртуальных коммутаторов VMware	117
2.4. Настройки Security, VLAN, Traffic Shaping и NIC Teaming	118
2.4.1. VLAN, виртуальные локальные сети. Настройка VLAN для стандартных виртуальных коммутаторов	118
2.4.2. Настройка VLAN для dvSwitch. Private VLAN	123
2.4.3. Security	126
2.4.4. Ограничение пропускной способности (Traffic Shaping)	128
2.4.5. NIC Teaming. Группировка сетевых контроллеров	128
2.4.6. Cisco Discovery Protocol, CDP	134
2.5. Разное	135
2.5.1. Jumbo Frames	135
2.5.2. TSO – TCP Segmentation Offload, или TOE – TCP offload engine	137
2.5.3. VMDirectPath	138

2.5.4. Отдельные порты.....	139
2.6. Рекомендации для сети.....	140
Глава 3. Системы хранения данных и vSphere	141
3.1. Обзор типов СХД.....	142
3.2. DAS	144
3.3. NAS (NFS).....	145
3.3.1. Настройка и подключение ресурса NFS к ESX(i)	147
3.4. SAN, Fibre Channel.....	149
3.4.1. Адресация и multipathing.....	152
3.4.2. Про модули multipathing. PSA, NMP, MMP, SATP, PSP.....	155
3.4.3. Про зонирование (Zoning) и маскировку (LUN masking, LUN presentation)	160
3.5. SAN, iSCSI	162
3.5.1. Как настроить программный инициатор или аппаратный зависимый iSCSI на ESX(i)	164
3.5.2. iSCSI Multipathing	169
3.6. VMFS, Virtual Machine File System.....	172
3.6.1. Увеличение размера хранилища VMFS. Grow и Extent	178
3.6.2. Доступ к клонированному разделу VMFS, или к разделу VMFS с изменившимся номером LUN	181
3.7. RDM, Raw Device Mapping.....	183
3.8. NPIV	186
3.9. Адресация SCSI.....	188
3.10. vSphere API for Array Integration, VAAI. Интеграция и делегирование некоторых операций системам хранения данных.....	191
Глава 4. Расширенные настройки, безопасность, профили настроек	194
4.1. Расширенные настройки (Advanced settings).....	194
4.2. Безопасность.....	195
4.2.1. Общие соображения безопасности	196
4.2.2. Брандмауэр ESX	198
4.2.3. Аутентификация на серверах ESX(i), в том числе через Active Directory	200
4.2.4. Контроль доступа, раздача прав при работе через vCenter ...	201
4.3. Настройка сертификатов SSL.....	208
4.4. Host Profiles.....	210
4.5. Использование SNMP	217
4.5.1. Настройка SNMP для vCenter	218
4.5.2. Настройка SNMP для серверов ESX(i)	221
Глава 5. Виртуальные машины.....	222
5.1. Создание VM. Начало работы с ней.....	222

5.2. Клонирование и шаблоны ВМ (Clone и Template)	227
5.2.1. Клонирование виртуальных машин	227
5.2.2. Шаблоны виртуальных машин (template).....	229
5.2.3. Обезличивание гостевых ОС, SysPrep.....	231
5.2.4. Рекомендации для эталонных ВМ	234
5.3. Виртуальное оборудование ВМ	236
5.3.1. Memory	237
5.3.2. CPUs.....	237
5.3.3. IDE, PS2 controller, PCI controller, SIO controller, Keyboard, Pointing device	238
5.3.4. Video card	238
5.3.5. VMCI device, VM Communication Interface	239
5.3.6. Floppy drive	239
5.3.7. CD/DVD Drive	239
5.3.8. Network Adapter	240
5.3.9. SCSI controller	247
5.3.10. Hard Disk	249
5.3.11. Parallel port.....	250
5.3.12. Serial port	250
5.3.13. SCSI device.....	251
5.3.14. USB controller и USB device	251
5.3.15. VMDirectPath	252
5.4. Все про диски ВМ.....	254
5.4.1. Виртуальные диски – файлы vmdk	254
5.4.2. Изменение размеров дисков ВМ	260
5.4.3. Выравнивание (alligment).....	266
5.4.4. Raw Device Mapping, RDM	270
5.5. Настройки ВМ	272
5.6. Файлы ВМ, перемещение файлов между хранилищами	276
5.7. Снимки состояния (Snapshot).....	284
5.8. VMware tools.....	292
5.9. vAPP.....	296

Глава 6. Управление ресурсами сервера.

Мониторинг достаточности ресурсов. Живая

миграция ВМ. Кластер DRS..... 299

6.1. Настройки распределения ресурсов для ВМ. Пулы ресурсов	299
6.1.1. Настройки limit, reservation и shares для процессоров и памяти.....	299
6.1.2. Пулы ресурсов	307
6.1.3. Рекомендации по настройкам Limit, Reservation и Shares	311
6.1.4. Storage IO Control, SIOC, для дисковой подсистемы.....	314
6.1.5. Network IO Control, NIOC и traffic shaping для сети	317
6.2. Механизмы перераспределения ресурсов в ESX(i).....	319

6.2.1. CPU	320
6.2.2. Memory	323
6.2.3. Disk	340
6.2.4. Net	340
6.3. Мониторинг достаточности ресурсов	340
6.3.1. Источники информации о нагрузке	341
6.3.2. Какие счетчики нас интересуют и пороговые значения	352
6.3.3. Несколько общих рекомендаций	358
6.4. Механизм Alarm	359
6.5. Миграция выключенной (или suspend) виртуальной машины	363
6.6. Storage vMotion – живая миграция файлов ВМ между хранилищами	365
6.7. vMotion – живая миграция ВМ между серверами	366
6.8. Кластер DRS. DPM	372

Глава 7. Защита данных и повышение доступности

виртуальных машин	388
7.1. Высокая доступность виртуальных машин	388
7.1.1. VMware High Availability, HA	389
7.1.2. VMware Fault Tolerance, FT	406
7.2. Управление обновлениями виртуальной инфраструктуры, VMware Update Manager	417
7.2.1. esxupdate и vSphere CLI vihostupdate	417
7.2.2. vSphere Host Update Utility	417
7.2.4. VMware Update Manager	419
7.3. Резервное копирование и восстановление	431
7.3.1. Резервное копирование ESX(i) и vCenter	431
7.3.2. Резервное копирование виртуальных машин	432
7.3.3. VMware Data Recovery	438
7.3.4. Использование VMware Consolidated Backup и vStorage API for Data Protection	445



Глава 2. Настройка сети виртуальной инфраструктуры

В этой главе будет рассказано про то, как работают с сетью серверы ESX(i), 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 только Cisco, модель Nexus 1000V). 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 ESX(i), объекты виртуальной сети

Основное соображение, которое необходимо уяснить: физические сетевые контроллеры сервера ESX(i) 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 ESX(i), то физические сетевые контроллеры являются лишь каналами во внешнюю сеть (Uplink). Через один физический сетевой контроллер в сеть могут выходить и управляющий интерфейс (виртуальная сетевая карта Service Console), и интерфейс для подключения NFS/iSCSI/vMotion/Fault Tolerance (виртуальная сетевая карта VMkernel, гипервизора), и разные виртуальные машины. (Здесь имеется в виду принципиальная возможность. Трафик разных назначений следует разделять по разным физическим сетевым контроллерам, см. раздел, посвященный сайзингу.)

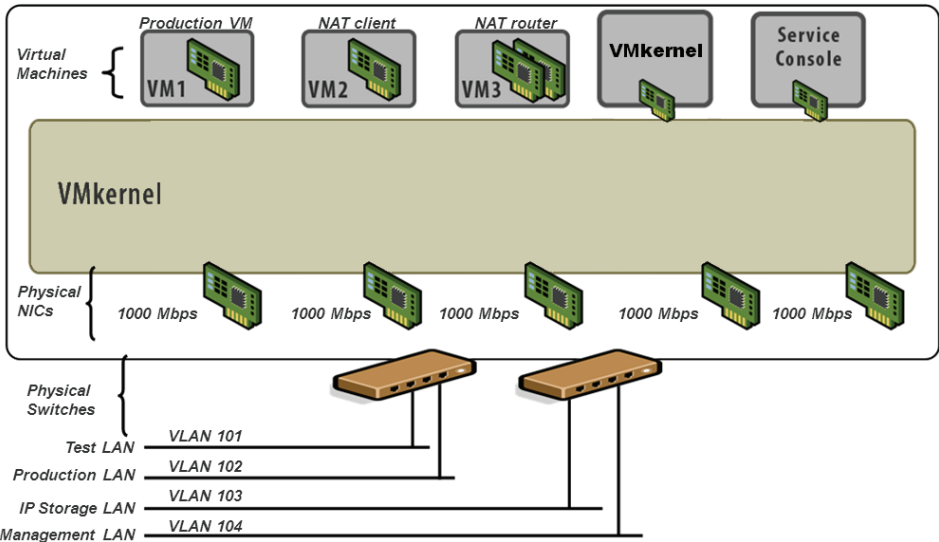


Рис. 2.1. Основные объекты сети «внутри» ESX(i)
Источник: VMware

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

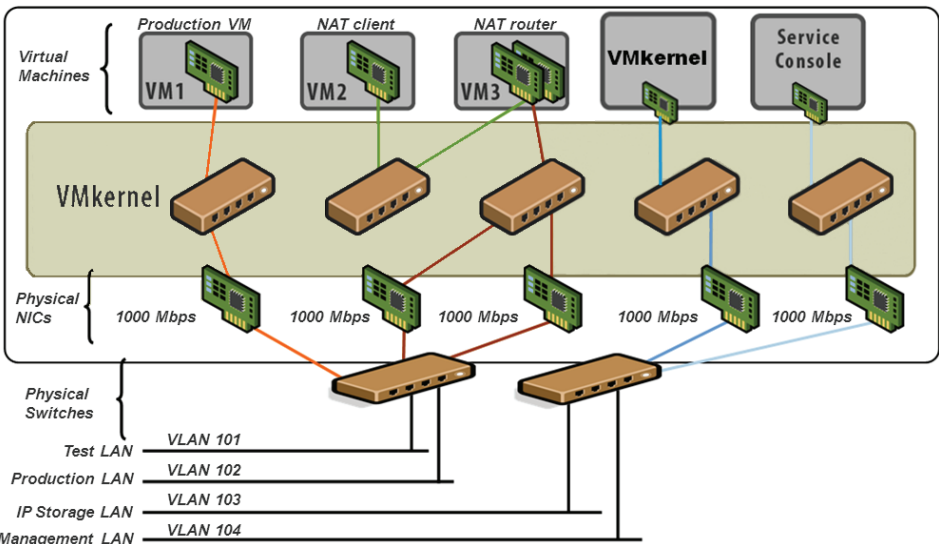


Рис. 2.2. Связь между объектами сети «внутри» ESX(i)
Источник: VMware

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. ESX(i) 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»;
- ❑ виртуальные коммутаторы (vSwitch или вКоммутаторы) – основные объекты сети на ESX(i);
- ❑ группы портов (Port groups) – логические объекты, создаваемые на вКоммутаторах. Виртуальные сетевые контроллеры подключаются именно к группам портов;
- ❑ виртуальные сетевые контроллеры. Они могут принадлежать виртуальным машинам, Service Console и VMkernel.

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

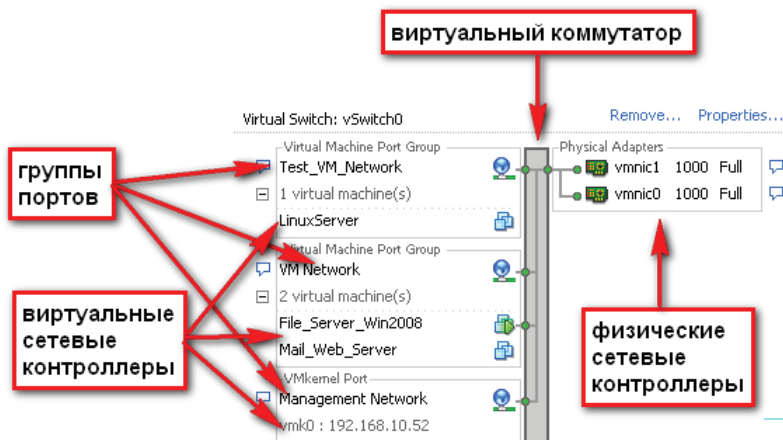


Рис. 2.3. Объекты виртуальной сети на ESX(i) в интерфейсе клиента vSphere

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.

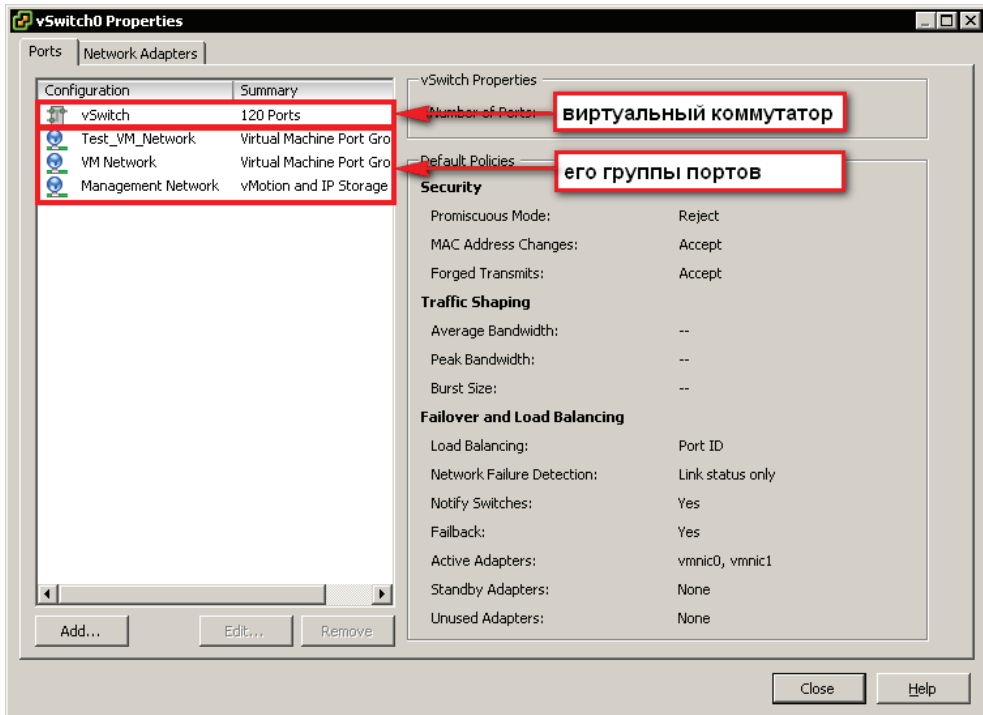


Рис. 2.4. Свойства виртуального коммутатора

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 ESX(i) (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) ⇒ закладка **Network Adapters** ⇒ выбираем нужный *vmnic* и нажимаем **Edit**.

Каждый физический сетевой контроллер нам надо привязать к вКоммутатору. Можно, конечно, и не привязывать – но это имеет смысл, лишь если мы хотим отдать этот *vmnic* напрямую VM. Подробности о таком варианте см. в разделе, посвященном виртуальным машинам, вас интересует функция *VMDirectPath*.

Итак, если мы не планируем использовать VMDirectPath, то не привязанный к vКоммутатору сетевой контроллер сервера у нас не используется никак. Это бессмысленно, поэтому, скорее всего, все vmpnic будут привязаны к тем или иным виртуальным коммутаторам. Правило логичное – один физический сетевой контроллер может быть привязан к одному, и только одному vКоммутатору. Но к одному vКоммутатору могут быть привязаны несколько vmpnic. В последнем случае мы можем получить отказоустойчивую и более производительную конфигурацию сети.

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

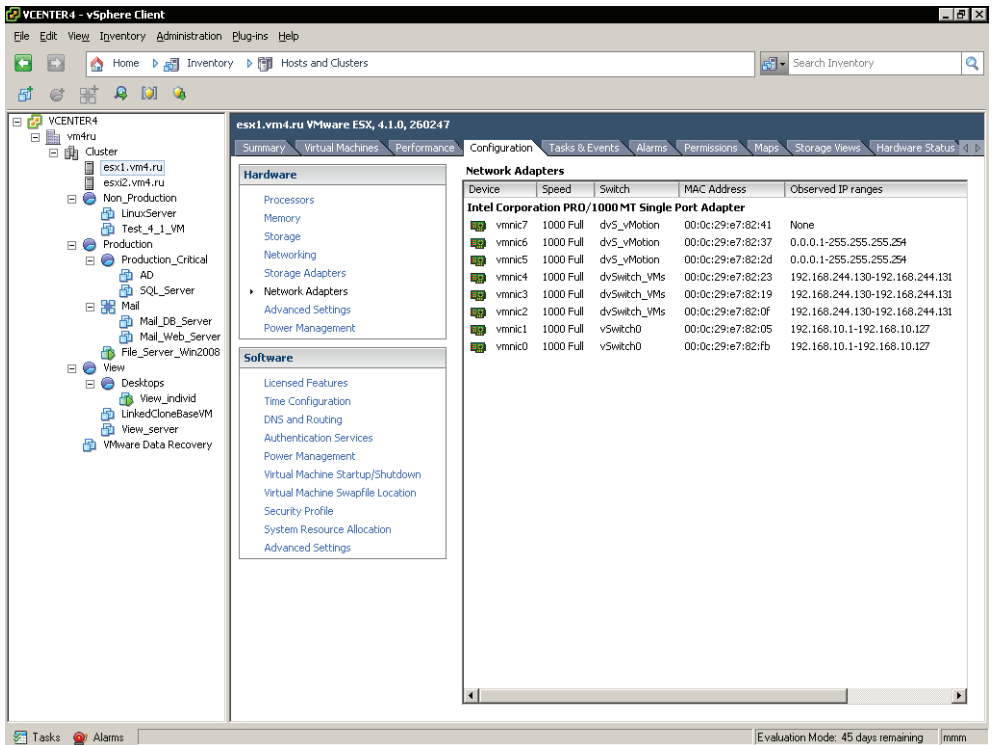


Рис. 2.5. Информация обо всех физических сетевых контроллерах сервера ESX(i)

В столбце **vSwitch** мы видим, к какому виртуальному коммутатору они привязаны. В столбце **Observed IP ranges** – пакеты из каких подсетей получает ESX(i) 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 vmpnic0 и vmpnic1 перехватывают трафик из одной подсети, а vmpnic2, vmpnic3 и vmpnic4 – из другой. Скорее всего, это означает, что они подключены к другому физическому коммутатору или принадлежат другому VLAN на физическом коммутаторе.

Обратите внимание. Физическим сетевым контроллерам ESX(i) дает имя вида **vmnic**. Командой `esxcfg-nics -l` вы выведете на экран список всех физических контроллеров сервера ESX и информацию о них. Фактически эта команда покажет на ту же информацию, что и соответствующее окно графического интерфейса (рис. 2.5).

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

2.1.2. Виртуальные контроллеры Service Console и VMkernel

Первое, чем отличаются виртуальные сетевые контроллеры, – это принадлежность. Принадлежать они могут Service Console (для ESX, у ESXi нет Service Console), VMkernel (самому гипервизору) и 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 Service Console и VMkernel.

Управляющий интерфейс ESX, виртуальный контроллер для Service Console (vswif)

Виртуальный сетевой контроллер для Service Console используется для управления сервером ESX. Один такой контроллер создается установщиком сервера ESX, именно ему принадлежит тот IP-адрес, который вы указывали при установке. Именно на IP-адрес Service Console вы подключаетесь клиентом vSphere, через него с сервером работает vCenter, на этот IP подключаются утилиты для работы с ESX. Также через интерфейс Service Console идут сигналы пульса (heartbeat) при работе кластера VMware HA. Наконец, некоторые варианты резервного копирования виртуальных машин осуществляются через интерфейсы Service Console.

Вам следует резервировать управляющий интерфейс, сделать это можно двумя путями, см. рис. 2.6.

В левой части рисунка вы видите дублированную конфигурацию единственного интерфейса Service Console. Она задублирована за счет того, что к вКмутата

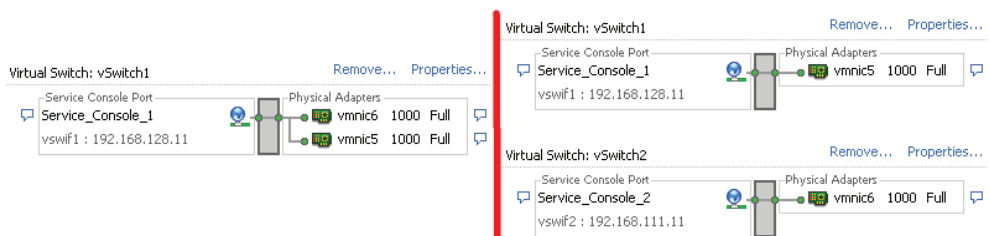


Рис. 2.6. Два варианта дублирования управляющего интерфейса ESX

тору 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 ESX по сети.

В правой части рисунка вы тоже видите дублированную конфигурацию, но дублированную по-другому – созданы два интерфейса Service Console, на разных вКоммутаторах, следовательно, на разных vmpnic. Если выйдет из строя один из каналов во внешнюю сеть (сам vmpnic или порт в физическом коммутаторе, к которому он подключен, или сам физический коммутатор), то один из интерфейсов SC станет недоступен, но останется другой.

Первый вариант удобнее в использовании, но если мы не можем себе позволить выделить два vmpnic на вКоммутатор с Service Console, 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 Service Console 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 SC;
- ❑ проблемы во внешней сети. Во втором примере двум разным интерфейсам SC 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 Service Console. Выбираем, жмем **Next**. Выберем, какие vmpnic привязать к создаваемому сейчас вКоммутатору. **Next**. Указываем имя (Network Label) – это название группы портов.

Это название – для человека, так что рекомендую давать говорящие названия. «Service_Console_2» вполне подойдет.

Однако при чем здесь группа портов, мы ведь создаем виртуальный сетевой интерфейс Service Console? Дело в том, что любой виртуальный сетевой контроллер числится подключенным именно к группе портов, поэтому при создании интерфейса SC из GUI мы создаем и интерфейс Service Console, и группу портов для него.

На стандартном виртуальном коммутаторе интерфейс Service Console всегда занимает свою группу портов целиком (или, что то же самое, группа портов для Service Console 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 Service Console (и, забегая вперед, 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 SC не на новом, а на уже существующем в Коммутаторе. 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 SC, когда он один, – в случае ошибки (например, опечатки в IP-адресе) доступ к управлению ESX по сети станет невозможен. Решается такая проблема из командной строки для ESX или из BIOS-подобного локального интерфейса ESXi. В любом случае понадобится доступ к локальной консоли сервера – физически или по iLO и аналогам.

Обратите внимание. Виртуальный сетевой контроллер для Service Console называется *vswif* – при создании их из GUI им даются имена вида *vswif#*, в командной строке мы управляем этими объектами с помощью `esxcfg-vswif`.

Управляющий интерфейс ESXi

Для ESX управление идет через интерфейс Service Console. Но в составе ESXi нет Service Console. Поэтому для ESXi в качестве интерфейсов управления используются виртуальные сетевые контроллеры VMkernel, те из них, в свойствах которых установлен флажок «Management traffic» (рис. 2.7).

Таким образом, организационные соображения здесь те же самые, что и для интерфейсов Service Console в ESX, но в качестве самих интерфейсов выступают интерфейсы VMkernel с соответствующим флажком в свойствах.

Виртуальный сетевой контроллер для VMkernel (vmk)

И в ESX, и в ESXi мы можем создать виртуальные сетевые адаптеры для VMkernel, для гипервизора. Нужны они для:

- ❑ 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;
- ❑ (только на ESXi) управления сервером (на ESX для этого используется сеть Service Console).

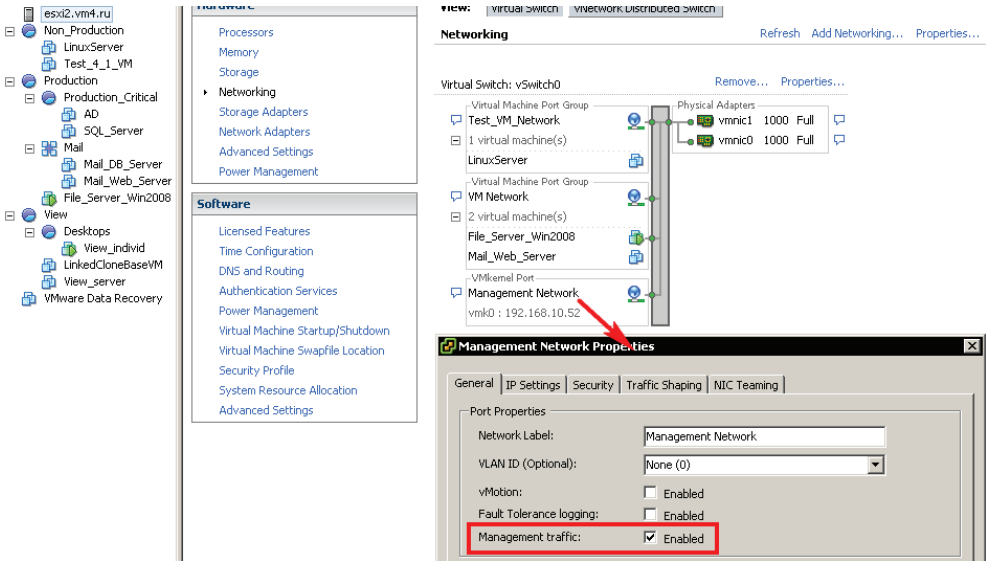


Рис. 2.7. Управляющий интерфейс ESXi

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.

Виртуальные сетевые контроллеры для гипервизора создаются точно так же, как для Service Console (на ESX): **Configuration** ⇒ **Networking** ⇒ и затем:

- ❑ либо **Add Networking** для создания нового вКоммутатора и интерфейса VMkernel на нем;
- ❑ либо для существующего вКоммутатора **Properties** ⇒ **Add** на закладке **Ports**;
- ❑ в случае распределенных виртуальных коммутаторов пройдите **Configuration** ⇒ **Networking** ⇒ кнопка **Distributed Virtual Switch** ⇒ ссылка **Manage Virtual Adapters** ⇒ **Add**.

В любом случае как тип добавляемого интерфейса выбираем **VMkernel**. Укажем имя группы портов и настройки IP для создаваемого интерфейса.

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

Каждый из флажков разрешает передачу соответствующего трафика через данный интерфейс. Технически допускается любая комбинация флажков на од-

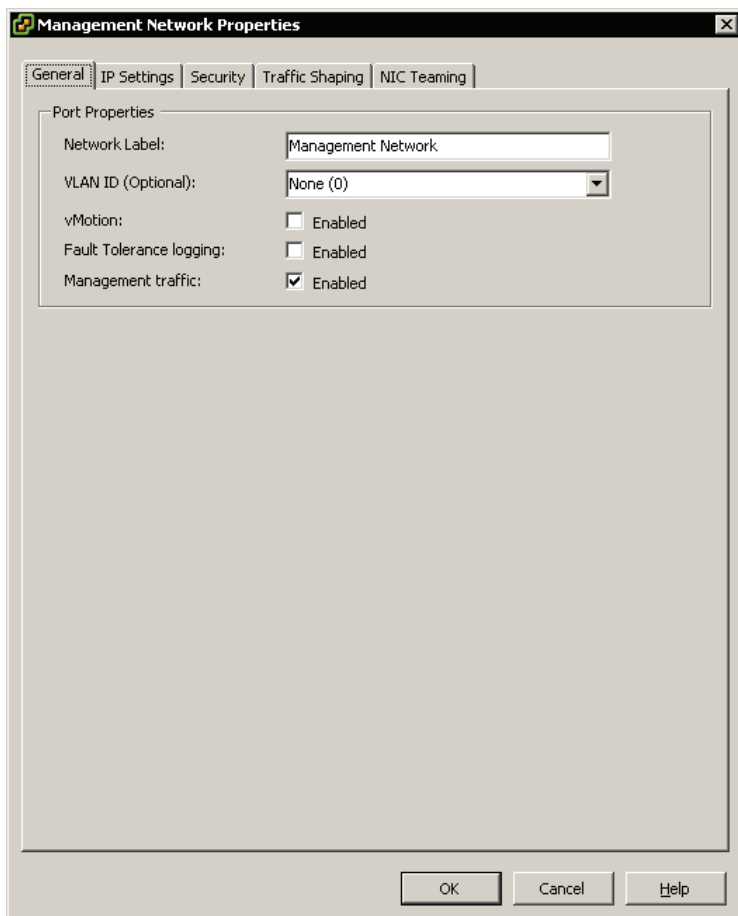


Рис. 2.8. Настройки интерфейса VMkernel на ESXi

ном виртуальном интерфейсе. Однако 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.

К тому же в случае использования ESXi через интерфейс с флажком «Management traffic» по умолчанию не пересылается трафик iSCSI и NFS. 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.

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 `esxconfig-vmknic`.

2.2. Стандартные виртуальные коммутаторы VMware – vNetwork Switch

Если вы перейдете на закладку **Configuration** ⇒ **Networking**, то увидите вашу виртуальную сеть (кстати, из командной строки можно увидеть все то же самое командой `esxcfg-vswitch -l`). Виртуальная сеть – это:

- виртуальные сетевые контроллеры, принадлежащие Service Console, VMkernel и виртуальным машинам;
- группы портов;
- виртуальные коммутаторы;
- физические сетевые контроллеры.

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

Начну с аналогии. ESX(i) с работающими на нем VM можно представить в виде серверной стойки или серверного шкафа. В стойке работают серверы, эти серверы 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 ESX(i), серверы 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 номера портов, входящих в группу, и даже их количество.

Таким образом, для настройки тех же самых VLAN или ограничения пропускной способности сети (traffic shaping) вам достаточно создать на vКоммутаторе 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 VM указать – сетевой контроллер подключен к группе портов «Production». Все. Нам не надо выбирать, к порту с каким номером мы хотим подключить VM. 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 VM. Ограничено только количество

портов на весь vКоммутатор, то есть количество виртуальных сетевых контроллеров (не VM, так как в одной VM может быть несколько виртуальных адаптеров), подключенных ко всем группам портов на этом vКоммутаторе.

Отсюда выводы:

Если вы хотите вывести во внешнюю сеть VM через определенные физические сетевые контроллеры, то вам надо выполнить следующее:

1. Создать vКоммутатор, к нему привязать эти vnic. **Configuration** ⇒ **Networking** ⇒ **Add Networking**.
2. В запущившемся мастере первым вопросом будет тип первой группы портов на создаваемом vКоммутаторе. Следует указать тип «Virtual Machine».
3. Выбрать vnic. Это дело потенциально не простое – о нем пара слов чуть позже. Кстати, вы можете создать vКоммутатор без единого привязанного физического контроллера. Это пригодится для помещения VM в изолированную сеть. Например, для использования того или иного рода прокси-сервера или брандмауэра. Если последнее вас интересует, обратите внимание на VMware vShield Zones (<http://www.vmware.com/products/vshield-zones/>).
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 vКоммутатора, поговорим немного подробнее. Представьте ситуацию, что у вас в сервере несколько vnic, 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. См. рис. 2.9 и 2.10.

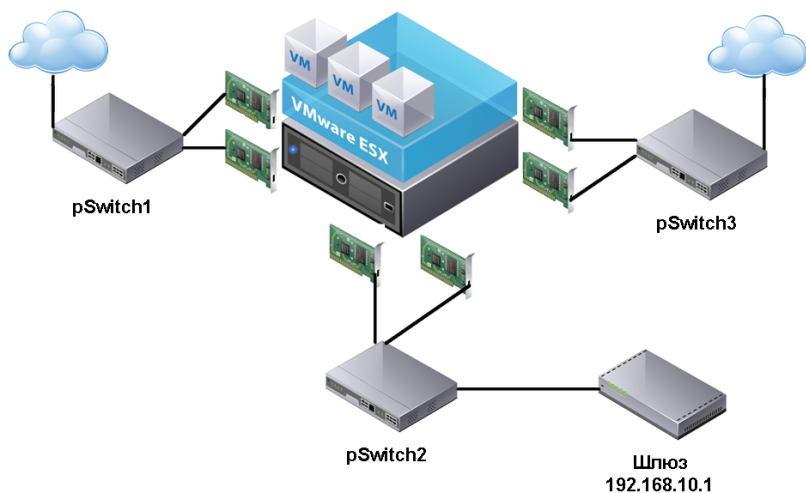


Рис. 2.9. Пример организации сети

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 vmpnic ее надо подключить? Нам может помочь **Configuration** ⇒ **Network Adapters**. См. рис. 2.10.

The screenshot shows the vSphere Client interface for an ESX host. The 'Network Adapters' section is expanded, displaying a table of virtual network interface cards (vmpnic) and their physical connections.

Device	Speed	Switch	MAC Address	Observed IP ranges
Intel Corporation PRO/1000 MT Single Port Adapter				
vmpnic7	1000 Full	dvS_vMotion	00:0c:29:e7:82:41	None
vmpnic6	1000 Full	dvS_vMotion	00:0c:29:e7:82:37	0.0.0.1-255.255.255.254
vmpnic5	1000 Full	dvS_vMotion	00:0c:29:e7:82:2d	0.0.0.1-255.255.255.254
vmpnic4	1000 Full	dvSwitch_VMs	00:0c:29:e7:82:23	192.168.244.130-192.168.244.131
vmpnic3	1000 Full	dvSwitch_VMs	00:0c:29:e7:82:19	192.168.244.130-192.168.244.131
vmpnic2	1000 Full	dvSwitch_VMs	00:0c:29:e7:82:0f	192.168.244.130-192.168.244.131
vmpnic1	1000 Full	vSwitch0	00:0c:29:e7:82:05	192.168.10.1-192.168.10.127
vmpnic0	1000 Full	vSwitch0	00:0c:29:e7:82:fb	192.168.10.1-192.168.10.127

Рис. 2.10. Информация о физических сетевых контроллерах ESX(i)

Что нам поможет?

Обратите внимание на столбец **Observed IP ranges** в этом окне. Диапазоны IP-адресов в нем – это подсети, пакеты из которых получает ESX на соответствующем интерфейсе. Это не настройка, не рекомендация – это информационное поле. То есть если вы знаете, что vmpnic, который вам сейчас надо подключить, коммутирован в сеть с известной вам подсетью, то вы сможете сориентироваться по представленной здесь информации.

Итак, глядя на рис. 2.9 и 2.10, к каким vmpnic надо подключить эту ВМ? Правильный ответ – vmpnic0 и vmpnic1. В моем примере они подключены к vКомму-

татору 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 vmpnic#) 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 закладке **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**;
- 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».