Содержание

Введение

Понятие виртуализации появилось еще в 60-х годах прошлого века. Уже тогда этим вопросом занималась компания IBM. Но тогда виртуализация не нашла применения, так возможности компьютеров и так использовались под завязку. Даже после появления персональных компьютеров ситуация не изменилась, виртуализация не начала своего развитие, так как идея на тот момент заключалась в запуске одной программы на одном устройстве, вследствие чего использование ресурсов было низким. Возможно, так все бы и продолжалось дальше, если бы не энергетический кризис, когда цена на электроэнергию возросла по всему миру, из-за которого возникла необходимость экономии ресурсов.

В конце 20 века, компания VMware на одном аппарате Intel запустила несколько операционных систем, а значит и приложений. То есть, была проведена виртуализация компьютера, а это позволило затраты на электроэнергию распределить на несколько операционных систем уже на одном комплекте аппаратного обеспечения. Таким образом, виртуализация позволила рационализировать нагрузку.

Как всё это работает? Для обеспечения виртуализации на сервере устанавливается ОС с уровнем виртуализации или создается тонкий слой программного обеспечения между сервером и ОС. В результате чего, виртуализация вводит ОС в заблуждение, чтобы она могла опознать ее как собственное аппаратное обеспечение. Это делается из-за слишком большого количества приложений, созданных под конкретную ОС. Таким образом, у виртуальной машины появляется ряд преимуществ, таких как:

- инкапсуляция – это сбор данных или функций в единый компонент. Сбор данных или функций осуществляется с помощью программы, которая маскируется под отдельную физическую машину в результате чего ОС вместо набора различных устройств видит набор разных файлов;

- изоляция означает, что все приложения, работая на одном устройстве, работают независимо друг от друга и опознают себя, как разные устройства. При этом, если зависает или падает одна ОС, это никак не влияет на работу других ОС и приложений;

- совмещение означает создание отдельного кластера, где ОС и все системы, которые с ней работают, имеют все функции отдельного компьютера. Это позволяет виртуальной, а не реальной машине взаимодействовать со всеми ОС и приложениями, которые работают на базе Intel x86;

- независимость от аппаратного обеспечения, поскольку операционная система и приложения устанавливаются на VM, конфигурация которой с течением времени не изменяется, вам незачем беспокоиться о проблемах совместимости с аппаратными средствами, циклах модернизации и поддержке устаревшего оборудования. Даже сложные многоуровневые корпоративные приложения можно перемещать между виртуальными серверами

без переустановки. Причём большинство систем виртуальных машин позволяют переносить конкретную виртуальную машину с одной системы на другую в рабочем режиме серверов. Это также позволяет делиться ресурсами, дисковым пространством и процессорной мощностью. Так, для приложения, которому требуется большое количество дискового пространства, нет необходимости добавлять диски к физическому серверу — их можно реконфигурировать в процессе работы. Поскольку большинство VM можно легко перемещать между машинами без перенастройки, вы полностью свободны в своих действиях, когда вам понадобится модернизировать или продлить срок эксплуатации существующего сервера. Отделение серверов от лежащего в их основе оборудования также означает, что вы можете перемещать VM в зависимости от требований к нагрузке.

Нынешнее развитие позволяет виртуализировать не только машины самого дата-центра, но и сети и системы хранения, связанные с ним. Для того, чтобы открыть свой собственный ЦОД необходимо всего лишь определиться с необходимой конфигурацией оборудования и обратиться в хостинговую компанию, на базе которой и будет создан центр обработки данных.

Чтобы сделать систему более доступной и удобной, обычно создают два ЦОДа: основной и резервный. Резервный нужен в качестве резервной копии основного дата-центра, чтобы при выходе из строя главного, быстро восстановить работоспособность, посредством переноса виртуальных машин с основного на резервный. Так же резервный служит для тестирования приложений: после тестирования виртуальной машины, она переносится на основной, где эксплуатируется.

Одним из главных преимуществ виртуализации является уменьшение количества необходимого физического оборудования, что приводит к уменьшению затрат, в том числе к снижению сопутствующих расходов (электричество, охлаждение и т.д.). Кроме того, при использовании виртуализации значительно увеличивается операционная гибкость и производительность системных администраторов.

Ярким примером этих преимуществ является использование вместо 50 физических серверов всего 4 виртуальных. Рост производительности в этом случае увеличивается до 80% (у физических 5-10%). А также снижается расход энергии за счет оптимизации нагрузки. При этом за счет снижения количества оборудования и выполняемых задач, значительно возрастает продуктивность работы системных администраторов.

Именно поэтому в современном мире наблюдается высокая потребность в виртуализации, что способствует быстрым темпам развитии данной области, и появлению новейшего обеспечения, предназначенного для реализации задач виртуализации.

1. Общая информация

Виртуализация — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе.

Виртуальная компьютерная система, также называемая **виртуальной машиной** (ВМ), — это строго изолированный контейнер ПО, содержащий операционную систему и приложение. Каждая автономная виртуальная машина полностью независима. Наличие нескольких ВМ на одном компьютере обеспечивает работу нескольких операционных систем и приложений на одном физическом сервере.

Тонкий уровень ПО, называемый гипервизором, отделяет виртуальные машины от сервера и по мере необходимости динамически выделяет вычислительные ресурсы каждой виртуальной машине.

Основные свойства виртуальных машин

ВМ имеют указанные ниже характеристики, которые обеспечивают ряд преимуществ.

Разбиение

- Выполнение нескольких операционных систем на одном физическом компьютере;
- Разделение системных ресурсов между виртуальными машинами.

Изоляция

- Изоляция сбоев и нарушений системы безопасности на аппаратном уровне;
- Сохранение уровня производительности с помощью расширенных средств управления ресурсами.

Инкапсуляция

- Полное сохранение состояния виртуальной машины в виде файлов;
- Перемещение и копирование виртуальных машин аналогичны операциям с файлами.

Независимость от оборудования

- Инициализация и перенос любой виртуальной машины на любой физический сервер.

Применение

- Защита информации и ограничения возможностей программ;
- Исследование производительности ПО или новой компьютерной архитектуры;
- Эмуляция различных архитектур;
- Оптимизация использования ресурсов мейнфреймов и прочих мощных компьютеров;

- Моделирование информационных систем с клиент-серверной архитектурой на одной ЭВМ (эмуляция компьютерной сети с помощью нескольких виртуальных машин);
- Упрощение управления кластерами — виртуальные машины могут просто мигрировать с одной физической машины на другую во время работы;
- Тестирование и отладки системного программного обеспечения.

1.1. Типы виртуализации

1.1.1. Виртуализация серверов

На сегодняшний день ресурсы большинства серверов используются менее чем на 15%, что приводит к росту числа серверов, а сама инфраструктура становится все более сложной. Виртуализация серверов может решить эти проблемы благодаря возможности запускать на одном физическом сервере несколько операционных систем в виде виртуальных машин, каждая из которых имеет доступ к вычислительным ресурсам сервера.

Следующий этап — объединение кластера серверов в один консолидированный ресурс, который повышает общую эффективность и снижает расходы. Кроме того, виртуализация серверов обеспечивает ускорение развертывания рабочих нагрузок, а также повышение производительности приложений и доступности.

1.1.2. Виртуализация сети

Виртуализация сети — это полное воспроизведение физической сети программным методом. Приложения в виртуальных сетях работают точно так же, как и в физических. Виртуализированные сети обеспечивают подключение рабочих нагрузок к логическим сетевым устройствам и службам, таким как логические порты, коммутаторы, маршрутизаторы, брандмауэры, средства балансировки нагрузки, сети VPN и т.д. Виртуальные сети аналогичны физическим с точки зрения надежности и возможностей и при этом обладают множеством дополнительных эксплуатационных преимуществ, таких как независимость от оборудования.

1.1.3. Виртуализация настольных компьютеров

Предоставление настольных компьютеров в качестве управляемой услуги повышает скорость реагирования на изменяющиеся потребности бизнеса и новые возможности. Такая модель помогает компаниям снизить расходы и повысить уровень обслуживания, поскольку виртуализированные настольные компьютеры и приложения одинаково доступны сотрудникам главного офиса и филиалов, а также удаленным, внештатным и мобильным сотрудникам, которые используют для работы планшеты iPad и Android.

1.2. Платформы виртуализации

Существует много различных платформ для виртуализации. Каждая из них обладает своими достоинствами и недостатками. В данной главе будет описано несколько наиболее распространенных платформ.

1.2.1. MS Hyper-V

Корпорация Microsoft в 2003 году выпустила свой первый продукт Virtual PC для ПК. В состав платформы виртуализации входят Windows 2008 Server R2 с компонентом Hyper-V, Microsoft Application Virtualization (App-v), Microsoft Virtual Desktop Infrastructure (VDI), Remote Desktop Services, System Center Virtual Machine Manager.

Эта технология виртуализации от MS, финальная версия которой была выпущена летом 2008 года. С выходом Win2k8R2 Hyper-V получил новые возможности — Live Migration, динамическая память, улучшены ряд инструментов и поддержка оборудования.

Технология Microsoft Hyper-V - это система аппаратной виртуализации на основе гипервизора, предоставляющая гостевым системам прямой доступ (т.е. без участия промежуточных виртуальных драйверов, замедляющих работу) к устройствам сервера (диск, память, процессор и т.д.). Данная возможность уменьшает расходы, благодаря чему достигается высокая скорость работы.

Hyper-V представляет собой в виде некой основы платформы виртуализации для серверов на базе процессоров с архитектурой x64. Распространяется двумя способами: как часть Windows Server 2008 (доступна в полном варианте и Server Core) или в составе независимого бесплатного продукта Hyper-V Server, как отдельное решение для установки на «голое железо» (по сути, представляет собой урезанный вариант Server Core, в котором установлена одна роль (без возможности изменения) и ограничены инструменты управления.

Между разными вариантами Hyper-V присутствуют и другие отличия, но в бесплатном варианте доступно все необходимое для построения сервера виртуализации. Это поддержка технологии Live Migration, консолидация серверов и кластеризация узлов.

Как уже говорилось, Hyper-V построен по принципу гипервизора с микроядром. Архитектура предполагает, что монитор виртуальных машин (МВМ) устанавливается прямо поверх аппаратного обеспечения, в отличие от того, где МВМ работает в среде хостовой ОС. Такой подход к построению МВМ позволяет достичь более высокой скорости работы, так как производит исключение накладных расходов, связанные с работой хостовой операционной системы.

Благодаря использованию в Hyper-V синтетических драйверов, которые не требует дополнительной эмуляции виртуальных устройств, обмены

данными при операциях ввода/вывода могут организовываться гораздо быстрее по сравнению с традиционными решениями виртуализации.

Качество Hyper-V

Чем ближе платформа виртуализации подходит к работе по типу физического сервера, тем проще можно развернуть виртуальные рабочие нагрузки и полагаться на их результат.

Hyper-V может разграничиваться по разделам. Раздел — это некая единица разграничения, которая поддерживается гипервизором, в ней как раз и работают операционные системы. У каждого гипервизора есть один раздел родителя, на котором запускается Windows Server или Linux Server. Виртуализационный стек включается на разделе родителя и имеет прямой доступ к физическим ресурсам машины. После чего, раздел родителя создает дочерние разделы, на которых непосредственно размещаются гостевые операционные системы.

Дочерний раздел тоже может создавать свои дочерние разделы. С помощью API гипервизора, который есть в Hyper-V.

У дочерних разделов нет доступа к физическим ресурсам, они имеют виртуальное представление ресурсов, которые называются виртуальными серверами.

При попытке обращения к виртуальным серверам идет перенаправление через VMBus к устройствам раздела родителя, которые с свою очередь начинают обрабатывать этот запрос. VMBus — это логический канал, с помощью которого разделы взаимодействуют между собой. Ответ приходит снова через VMBus.

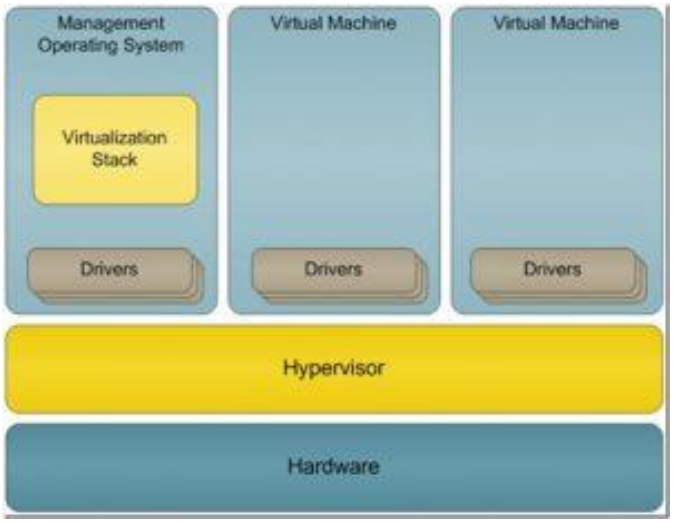

Рис.1. Архитектура Hyper-V

1.2.2. KVM

Технология виртуализации KVM (Kernel-based Virtual Machine) продвигается компанией RedHat и является «основной» в этом дистрибутиве и его клонах. Требует поддержку аппаратной виртуализации Intel VT или AMD V. Это означает, что KVM может использоваться далеко не на каждом компьютере: старые и некоторые из новых CPU (например, Intel Atom) не подходят.

KVM – это инфраструктура виртуализации для ядра Linux, которая поддерживает платформенно-зависимую виртуализацию на процессорах с аппаратными расширениями для виртуализации. Первоначально он поддерживал процессоры x86, но в настоящее время к ним добавился широкий спектр процессоров и гостевых операционных систем, в том числе множество вариаций Linux, BSD, Solaris, Windows, Haiku, ReactOS и AROS Research Operating System (есть даже модифицированная версия QEMU, способная использовать KVM для работы с Mac OS X).

KVM является открытым программным обеспечением, распространяется по лицензии GPL v2 и состоит из модуля ядра, модуля взаимодействия с процессором и компонента пользовательского режима (на базе QEMU). Для лучшей производительности, процессор должен поддерживать набор инструкций для виртуализации (Intel VT или AMD-V).

KVM используется как в корпоративных решениях (Redhat) так и для малых объёмов виртуализации. Уже несколько лет эта технология является основной для RHEL (Red Hat Enterprise Linux) вместо XEN.

Существуют две редакции KVM – платная (RHEV) и бесплатная, причём в бесплатной доступен почти весь функционал. Также бесплатную версию можно сравнить с коммерческими продуктами.

KVM (kernel virtual machine) расшифровывается как виртуальная машина ядра, то есть гипервизор включен в ядро, за счет чего достигается очень высокая производительность виртуализации.

KVM не использует самоэмуляцию. Программа, которая работает в пространстве пользователя, использует интерфейс kvm, осуществляя через него настройку адресного пространства гостевой виртуальной системы, использует её I/O ресурсы, отображая образ гостевой системы на образ хоста.

В архитектуре KVM, виртуальная система исполняется как родной процесс, который запланирован стандартными средствами ОС Linux. При этом каждый виртуальный процессор отображается как стандартный Linux процесс. Благодаря этому KVM имеет возможность использовать все возможности ядра Linux.

Управление эмуляцией устройств осуществляется при помощи модифицированной версии qemu, обеспечивающей эмуляцию BIOS и всех стандартных шин, а также набор системных устройств (контроллеры IDE и SCSI, сетевые карты и т.д.).

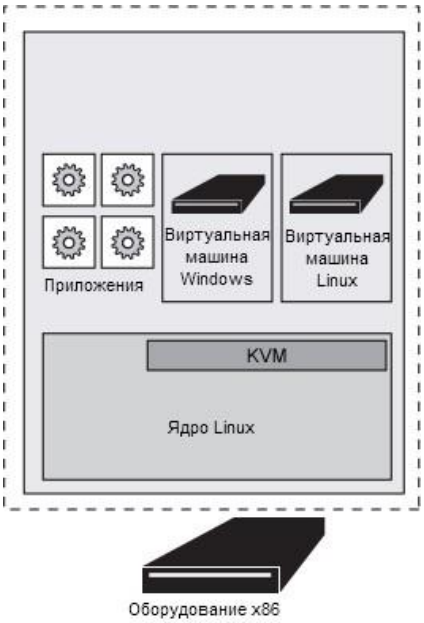

Рис.2. Архитектура KVM

Для взаимодействия с данным гипервизором можно использовать разные инструменты, а именно:

- virt-manager
- virsh
- python-libvirt
- разные утилиты (virt-top, virt-install, virt-view)

Все вышеописанные продукты используют библиотеку libvirt, но предоставляют разные функции. Так, virt-manager - это графическая оболочка, с помощью которой упрощаются операции создания-удаления, остановки-запуска виртуальных машин. Написана она на языке Python, и для запуска требует либо графическое окружение на сервере, либо проброс X over SSH.

Virsh является утилитой командной строки, а точнее "командной строкой" KVM.

При запуске мы попадаем в интерфейс взаимодействия с гипервизором, где нам доступно около 200 разных команд. Данная команда полезна для дебаггинга, а также для особых задач, которые недоступны из интерфейса графической оболочки.

Python-libvirt является обвязкой к libvirt на языке Python. В данную библиотеку включены многие (но не все) методы libvirt, что позволяет использовать вызов напрямую из Python-скриптов и программ.

Утилиты (virt-top, virt-install, virt-view) позволяют получить доступ к наиболее востребованным функциям KVM прямо из командной строки.

Например, virt-install облегчает создание новой ВМ.

Ещё одна утилита – virt-top. Она в режиме реального времени (или опционально в файл) выводит данные о виртуальных машинах. В ней вам доступна информация о процессоре, памяти, дисковой и сетевой подсистеме.

На сегодняшний день KVM предоставляет большинство возможностей коммерческих продуктов бесплатно, а также имеет открытый интерфейс, что позволяет создавать продукты на его основе и добавлять необходимые функции.

1.2.3. VMware ESXi

VMware – одна из ведущих компаний на рынке платформ виртуализации. Программные технологии виртуализации VMware запатентовала в 1998 году и выпустила профессиональные продукты для виртуализации различного уровня: от VMware Workstation, предназначенного для настольных ПК, до VMware ESX Server, позволяющего консолидировать физические серверы предприятия в виртуальной инфраструктуре.

Виртуализация VMware на данный момент пользуется большой популярностью благодаря своим функциональным возможностям и производительности.

Для установки на «голое железо» предлагается VMware ESXi. Это самостоятельный продукт, являющийся основой для установки гостевых операционных систем, а совместно с VMware vSphere — средством для построения виртуальной инфраструктуры и управления виртуальными ресурсами. По сути, ESXi — это сильно урезанная версия Linux, которая содержит гипервизор (VMkernel) и такие консоли управления, как vCLI (vSphere CLI), PowerCLI (PowerShell интерфейс к vCLI), SSH и DCUI (Direct Console User Interface).

Ранее ESXi представляет собой бесплатный и урезанный вариант ESX. Но время ESX прошло, следующие версии VMware VSphere будут включать поддержку исключительно ESXi, а все преимущества ESX перед ESXi сошли на нет. Так что на сегодняшний день разработчики рекомендуют переходить на ESXi.

Таким образом, VMware ESXi является гипервизором нового поколения, не зависящим от ОС, который легко позволит развертывать решения для виртуализации. С помощью этого гипервизора, предъявляющего минимальные требования к конфигурации, можно запускать производственную среду за считанные минуты, а сам гипервизор можно масштабировать для запуска самых ресурсоемких приложений. Гипервизор VMware ESXi позволяет организовать оперативную работу центров обработки данных, обеспечивая повышенную безопасность, надежность и управляемость (доступен также в виде системы, встроенной в серверное аппаратное обеспечение).

Главное отличие ESXi от ESX заключается в архитектуре.

- Независимость от операционной системы и оптимизация производительности виртуальной среды.

- Уникальный уровень безопасности благодаря небольшому размеру в 32 МБ и ориентированной на виртуализацию архитектуре.

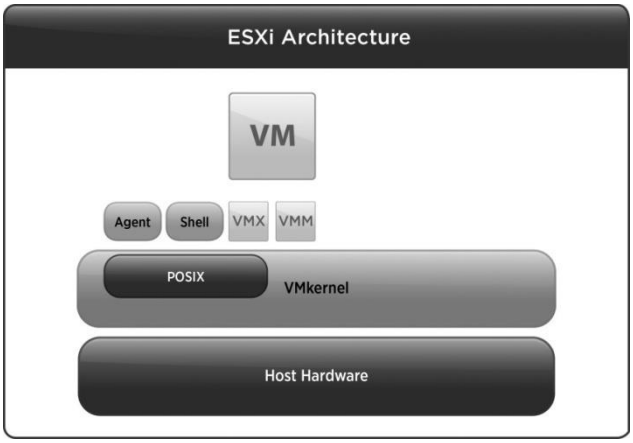

Рис.3. Архитектура ESXI

Занимая всего лишь 32 МБ в основной ОС, VMware ESXi устанавливает новую планку безопасности благодаря меньшей потенциальной «площади атаки». Такой небольшой размер данной платформы и высокая надежность позволяют адаптировать VMware ESXi непосредственно в стандартные серверы с архитектурой x86. Благодаря этому можно развертывать бездисковые серверы, подключенные к общему хранилищу, устраняя наиболее подверженный сбоям компонент сервера и сокращая энергопотребление и требования к системе охлаждения.

VMware ESXi работает со стандартными протоколами мониторинга и легко управляется с помощью VMware vSphere Client. VMware ESXi, входящий в состав пакета VMware vSphere, а также поддерживает сервер VMware vCenter и API-интерфейс VMware vSphere для интеграции с программными решениями сторонних производителей.

В ESXi агенты работают прямо в VMkernel, при необходимости модули сторонних разработчиков (мониторинг, драйвера) также выводятся на гипервизор. Уменьшение слоев означает большую надежность и безопасность, следовательно, меньше возможностей для атак.

Продукт от VMware отличает поддержка большого количества гостевых ОС (Windows, Linux, Solaris, FreeBSD, Netware и многие другие).

Установка дистрибутива на голое железо очень проста (можно реализовать с помощью стандартного варианта с привода или через PXE), к тому же начиная с версии 4.1 поддерживаются сценарии, которые позволяют автоматизировать процесс инсталляции ПО, настройку сети и подключения к vCenter Server.

Немаловажно наличие специального конвертера VMware vCenter Converter, позволяющего использовать в ESXi образы MS Virtual Server, Virtual PC, Hyper-V, а также физические серверы и образы дисковых разделов.

Виртуализация VMware помогает сократить капитальные расходы за счет консолидации серверов и эксплуатационные расходы за счет автоматизации. Имеет возможность аварийного восстановления в виртуализированной инфраструктуре с помощью улучшенных и упрощенных решений по аварийному восстановлению на базе VMware® vCenter Site Recovery ManagerTM.

Дальнейшая часть посвящена работе с продуктом компании VMware.

2. Установка VMware vSphere ESXi

VSphere и vSphere с Operations Management обеспечивают виртуализацию с последовательным управлением, специально созданным для получения максимальной производительности, доступности и эффективности вашей инфраструктуры и приложений с начала работы. Архитектура VMware vSphere состоит из следующих компонентов:

- Базовый гипервизор vSphere ESXi, который устанавливается на каждом физическом сервере для размещения виртуальных машин.
- Один экземпляр сервера управления vCenter Server, обеспечивает централизованное управление несколькими узлами vSphere.
- Компонент управления операциями vRealize Operations, оптимизирует vSphere, контролируя производительность и управляя мощностью.
- С помощью vRealize Log Insight проводится анализ журнала и управлением им, что предоставляет эффективные панели мониторинга с глубокой оперативной видимостью.

2.1. Установка vSphere ESXi

ESXi- это операционная система, с включенным в нее гипервизором, который и обеспечивает виртуализацию.

Хост (Host) - физический сервер, на котором установлен ESXi.

VMkernel- это название гипервизора, являющегося частью ESXi.VMkernel неотъемлемая часть ESXi, поэтому их можно воспринимать как синонимы, поэтому "интерфейс управления VMkernel " означает тоже самое, что и "интерфейс управления ESXi".

Перед установкой ESXi необходимо убедиться, что ваше устройство поддерживает его, а аппаратные и системные ресурсы отвечают его требованиям. Все их можно найти на сайте VMware.

Варианты установки ESXi:

ESXi может быть установлен несколькими способами. Чтобы обеспечить наилучшее развертывание vSphere, перед началом установки тщательно изучите параметры. Установки ESXi рассчитаны на широкий диапазон размеров развертывания. В зависимости от выбранного вами метода установки доступны различные опции для доступа к установочным материалам и загрузки установщика.

- Интерактивная установка - рекомендуется для небольших развертываний менее пяти хостов. Программа установки загружается с CD- или DVD-диска, с загрузочного USB-устройства или с помощью PXE загружаете из сети, после чего ESXi вручную устанавливается на каждый хост.
- Скрипт установки - эффективный способ развернуть несколько хостов ESXi с автоматической установкой. Он содержит параметры

конфигурации хоста, поэтому его можно использовать для настройки нескольких хостов с одинаковыми настройками. Этот скрипт должен храниться в местоположении, к которому хост может получить доступ по HTTP, HTTPS, FTP, NFS, CDROM или USB.

- Автоматическая установка (AutoDeploy) - позволяет эффективно создавать большое количество хостов с помощью PXE-загрузки. На сервер загружается образ ESXi, откуда все хосты скачивают его себе. С помощью этой функции так же удобно обновлять ESXi, для чего необходимо просто загрузить новую версию на сервер, обновление на хостах произойдет автоматически.
- Автоматическое развертывание с кэшированием - можно использовать vSphere Auto Deploy для предоставления хоста ESXi и настроить профиль хоста, который заставит хост сохранить образ и конфигурацию ESXI на локальном диске, удаленном диске или USB-накопителе. Если сервер Auto Deploy недоступен, хост использует изображение на диске.

Настройка инсталляций с помощью ESXi Image Builder CLI

ESXi Image Builder CLI - это набор команд CLI PowerShell, который вы можете использовать для создания установочного образа ESXi с индивидуальным набором обновлений и патчей ESXi. Вы также можете включить сторонние драйверы сети или хранилища, которые выпущены между выпусками vSphere.

Образ ESXi, созданный с помощью Image Builder, можно развернуть одним из следующих способов:

- Записав на установочный DVD;
- Через vCenter Server, используя функциюAuto Deploy.

2.1.1. Интерактивная установка

Для установки программного обеспечения ESXi на жесткий диск SAS, SATA, SCSI или USB используйте ESXi CD / DVD или USB-накопитель.

Подготовка:

- Установить ISO-инсталлятор ESXi в одном из следующих мест:
  - На CD или DVD-диск,
  - На флэш-накопителе USB.
- Или можно использовать PXE установщик ESXi
- Убедиться, что аппаратные часы сервера настроены на UTC.
- Подключить монитор и клавиатуру к аппарату, на котором будет установлено программное обеспечение ESXi. Также можно использовать приложение удаленного управления.
- При возможности отключить сетевое хранилище. Это действие сокращает время, которое требуется установщику для поиска доступных дисков. Обратите внимание, что при отключении сетевого хра-

нилища все файлы на отключенных дисках становятся недоступными при установке.

- Убедитесь, что ESXi Embedded отсутствует на хост-машине. ESXi Installable и ESXi Embedded не могут существовать на одном хосте.

Установка:

1. Вставьте установочный CD / DVD-диск ESXi в дисковод CD / DVD-ROM или подключите флэш-накопитель USB и перезапустите устройство.

2. Установите в BIOS загрузку с устройства CD-ROM или флэш-накопителя USB.

3. На странице «Выбор диска» выберите диск для установки ESXi и нажмите «Ввод». Нажмите F1 для получения информации о выбранном диске.

4. Выберите тип клавиатуры для хоста (тип клавиатуры можно изменить после установки в консоли)

5. Введите пароль root для хоста. Можно оставить пароль пустым, но чтобы защитить систему этого делать не стоит. Вы можете изменить пароль после установки с помощью консоли.

6. Нажмите «Ввод», чтобы начать установку.

7. По завершении установки извлеките установочный компакт-диск, DVD-диск или флэш-накопитель USB.

8. Нажмите «Ввод», чтобы перезагрузить хост.

9. Как во втором шаге установите первым загрузочное устройство, на котором был установлен ESXi.

2.1.2. Установка с помощью vSphereAutoDeploy

VSphere Auto Deploy позволяет снабжать сотни физических хостов программным обеспечением ESXi. Хосты загружаются из сети с центрального сервера Auto Deploy. После завершения начальной загрузки и настройки хосты управляются сервером vCenter так же, как и другие хосты ESXi.

Принцип работы:

1. Загружается целевой хост ESXi.

2. Этот хост делает DHCP-запрос.

3. DHCP-сервер отвечает с IP-параметрами и расположением TFTP-сервера, который содержит загрузчик.

4. ESXi обращается к серверу TFTP и запрашивает файл загрузчика, который указал DHCP-сервер.

5. TFTP-сервер посылает загрузчик хосту ESXi, который исполняет его. Начальный загрузчик догружает дополнительные компоненты с TFTP-сервера.

6. Загрузчик ищет конфигурационный файл на TFTP-сервере, скачивает ядро и другие компоненты ESXi с HTTP-сервера, который размещен на TFTP-сервере, и запускает ядро на хосте ESXi.

7. Установщик ESXi запускается в интерактивном режиме, либо используя kickstart-скрипт, который указан в конфигурационном файле.

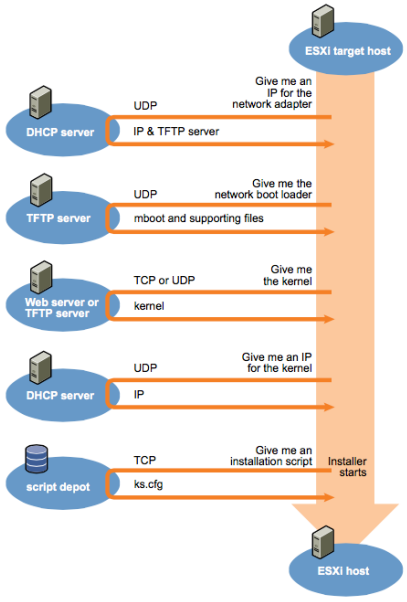

Рис.4. Принцип установки ESXi с помощью AutoDeploy

Для того, чтобы установить ESXi с помощью VSphereAutoDeploy необходимо:

- УстановитьAutoDeployвvCenterServer, если она не была установлена.
- Настроить vCenter: создать контейнер для серверов, назначить ключ лицензии.
- Настроить DHCPи TFTP.
- Установить PowerCLI.
- Настроить AutoDeploy для первого сервера, настроить профили хоста. Профиль хоста (HostProfiles) - функция, позволяющая применять нужные настройки сервера автоматически при каждой загрузке.
- Настроить AutoDeploy для последующих серверов.

После этого каждый хост, находящийся в сети будет автоматически подключаться к серверу AutoDeploy, устанавливая с него ESXi.

3. Виртуальные машины

3.1. Виртуальные машины и их параметры

Виртуальная машина - это компьютер с программным обеспечением, которое, подобно физическому компьютеру, управляет операционной системой и приложениями. Виртуальная машина состоит из набора файлов спецификации и конфигурации и поддерживается физическими ресурсами хоста. На каждой виртуальной машине есть виртуальные устройства, которые обеспечивают ту же функциональность, что и физическое оборудование, более портативное, более безопасное и удобное в управлении.

Виртуальная машина состоит из нескольких файлов, которые хранятся на запоминающем устройстве. Ключевыми файлами являются файл конфигурации, файл виртуального диска, файл настроек NVRAM и файл журнала. Параметры виртуальной машины настраиваются через vSphere Web Client, один из интерфейсов командной строки vSphere (PowerCLI, vCLI) или vSphere Web Services SDK.

3.1.1. Инфраструктура виртуальных машин

Инфраструктура, поддерживающая виртуальные машины, состоит как из двух уровней программного обеспечения: виртуализации и управления.

VCenter Server позволяет объединять и управлять ресурсами нескольких хостов, а также эффективно контролировать физическую и виртуальную инфраструктуру и управлять ею. Вы можете управлять ресурсами для виртуальных машин, предоставлять виртуальные машины, планировать задачи, собирать журналы статистики, создавать шаблоны и многое другое. VCenter Server также предоставляет vSphere vMotion ™, vSphere Storage vMotion, vSphere Distributed Resource Scheduler (DRS), vSphere High Availability (HA) и vSphere Fault Tolerance. Эти службы обеспечивают эффективное и автоматизированное управление ресурсами и высокую доступность для виртуальных машин.

VMware vSphere Web Client - это интерфейс для серверов vCenter Server, ESXi и виртуальных машин. С помощью vSphere Web Client вы можете удаленно подключиться к vCenter Server. VSphere Web Client - это основной интерфейс для управления всеми аспектами среды vSphere. Он также обеспечивает консольный доступ к виртуальным машинам.

В иерархии сервера vCenter центр данных является основным контейнером хостов ESXi, папок, кластеров, пулов ресурсов, vSphere vApps, виртуальных машин и т.д.

Хранилища данных представляют собой виртуальные представления базовых физических ресурсов хранения в центре обработки данных. Хранилище данных - это место хранения (например, физический диск или LUN на RAID или SAN) для файлов виртуальной машины. Хранилища данных скрывают особенности основного физического хранилища и пред-

ставляют собой унифицированную модель ресурсов хранения, необходимых виртуальным машинам.

3.1.2. Жизненный цикл виртуальной машины

Создание и развертывание виртуальной машины в центре данных производится наиболее удобным способом: создание новой виртуальной машины и установка на нее ОС, клонирование имеющейся ВМ или создание из нее шаблона для последующего использования.

С помощью мастера создания виртуальной машины vSphere Web Client и редактора свойств можно добавлять, настраивать или удалять большинство аппаратных средств, параметров и ресурсов виртуальной машины. Мониторинг показателей ЦП, памяти, диска, сети и хранилища осуществляется с помощью диаграмм производительности в vSphere Web Client. Снапшоты позволяют фиксировать состояние виртуальной машины, включая память виртуальной машины, параметры и виртуальные диски. При необходимости можно вернуться к предыдущему состоянию виртуальной машины.

vSphere vApps позволяет управлять многоуровневыми приложениями. vSphere Update Manager используется для выполнения организованных обновлений виртуального оборудования одновременно.

Когда виртуальная машина больше не нужна, ее можно удалить из инвентаря, не удаляя ее из хранилища данных, или удалить виртуальную машину и все ее файлы.

3.1.3. Параметры виртуальной машины и ресурсы

Каждое виртуальное устройство выполняет ту же функцию для виртуальной машины, что и аппаратное обеспечение на физическом компьютере.

Виртуальная машина может работать в любом из нескольких мест, таких как хост ESXi, дата-центр, кластер или пул ресурсов. Многие параметры и ресурсы, которые настраиваются, имеют зависимости и отношения с этими объектами.

Каждая виртуальная машина имеет ресурсы процессора, памяти и диска. Виртуализация CPU отвечает за производительность и, преимущественно, запускается непосредственно на процессоре. По возможности используются основные физические ресурсы. Уровень виртуализации выполняет инструкции только по мере необходимости, чтобы виртуальные машины работали так, как если бы они работали непосредственно на физической машине.

Все последние операционные системы поддерживают виртуальную память, позволяя программному обеспечению использовать больше памяти, чем физически имеет компьютер. Кроме того, гипервизор ESXi обеспечивает поддержку избыточной памяти виртуальной машины, где объем

гостевой памяти, настроенный для всех виртуальных машин, может превышать объем физической памяти хоста.

Виртуальные диски и дополнительное пространство к существующим дискам можно добавлять даже когда виртуальная машина работает.

Виртуальные машины VMware имеют опции, представленные в таблице 1.

Таблица 1 – Опции виртуальных машин.

Опции	**За что отвечают**
Общие настройки	Просмотр и изменение имени виртуальной машины и настройка расположения файла конфигурации и рабочего места виртуальной машины.
Инструменты VMware	Управление питанием виртуальной машины и запуск скрипта VMware Tools. Обновление инструментов VMware во время циклического энергопотребления и синхронизация гостевого времени с хостом.
Расширенные опции	Отключение ускорения и включение ведения журнала, настройка отладки и статистики, изменение местоположения файла подкачки. Изменение чувствительности к задержкам и добавление параметров конфигурации.
Управление энергопотреблением	Управление параметрами питания гостевых ОС. Приостановка работы виртуальной машины или перевод гостевой операционной системы в режим ожидания.
Маска CPUID	Включение/Отключение NX/XD. Отключение NX/XD увеличивает совместимость vMotion между хостами.
Горячее подключение к памяти / процессору	Включение или отключение горячего подключение к CPU и памяти. Можно добавить ресурсы памяти или ЦП к виртуальной машине во время ее работы.
Параметры загрузки	Устанавливается время загрузки при включении виртуальных машин и настройка восстановления после неудачной загрузки.
Волоконный канал NPIV	Управление доступом к виртуальным машинам для LUN на основе виртуальной машины. Виртуализация N-port ID (NPIV) предоставляет возможность совместного использования одного физического адаптера Fibre Channel HBA между несколькими виртуальными портами, каждый с уникальными идентификаторами.
Параметры vApp	Включить или отключить функции vApp. С помощью него можно настроить политику распределения IP или профиль сетевого протокола для vApp.

3.1.4. VMwareTools

VMware Tools - это набор утилит, которые устанавливаются в операционной системе виртуальной машины.

VMware Tools повышает производительность виртуальной машины и делает возможными многие удобные функции в продуктах VMware.

Например:

- Более высокая производительность графики и Windows Aero в операционных системах, поддерживающих Aero
- Функция Unity, которая позволяет приложению на виртуальной машине появляться на главном рабочем столе подобно любому окну приложения
- Общие папки между хостом и гостевыми файловыми системами
- Копирование и вставка текста, графики и файлов между виртуальной машиной и рабочим столом хоста или клиента
- Улучшенная производительность мыши
- Синхронизация часов в виртуальной машине с часами на рабочем столе хоста или клиента
- Сценарии, помогающие автоматизировать операции гостевой операционной системы
- Периодически собирает информацию о сети, дисках, а также об использовании памяти из гостевой операционной системы и отправляет его на ESXi-хост.
- Проверяет доступность виртуальных машин и операционных систем каждую секунду.

3.2. Создание виртуальных машин

VMware предлагает несколько методов для обеспечения виртуальных машин vSphere. Оптимальный метод для вашей среды зависит от таких факторов, как размер и тип вашей инфраструктуры и цели, которые вы хотите достичь.

3.2.1. Создание новой виртуальной машины

Если никакие другие виртуальные машины в вашей среде не имеют требований, которые необходимы, например конкретная операционная система или аппаратная конфигурация, то необходимо создавать новую виртуальную машину. В этом случае можно настроить виртуальное оборудование, включая процессоры, жесткие диски и память.

Процедура:

1. Запуск процесса создания новой виртуальной машины.

2. Выбор имени виртуальной машины и папки, где она будет находиться.

3. Выбор ресурса: хост, кластер, vApp или пул ресурсов, где будет запускаться виртуальная машина.

4. Выбор хранилища данных, где будут храниться файлы конфигурации виртуальной машины и ее виртуальные диски.

5. Настройка совместимости: выбор версии EXSi-хоста, в зависимости от хостов вашей среде.

6. Выбор гостевой операционной системы: выбранная ОС влияет на поддерживаемые устройства и количество виртуальных процессоров, доступных ВМ. Операционная система установлена не будет, только использована информация для установки подходящих параметров по умолчанию, таких как объем необходимой памяти.

7. Настройка оборудования ВМ. Во время процесса создания для виртуальной машины настроен диск по умолчанию. При необходимости можно удалить этот диск и добавить новый жесткий диск, выбрать существующий диск или добавить диск RDM.

8. Завершение создания. Перед развертыванием ВМ можно проверить ее настройки.

9. Установка гостевой ОС. Осуществляется так же как и установка операционной системы на физическом компьютере. Гостевую ОС на виртуальную машину можно установить с сервера PXE или с помощью ISO-образа с носителя.

3.2.2. Создание виртуальной машины из шаблона

Чтобы сэкономить время, можно создать виртуальную машину, которая является копией настроенного шаблона.

Процедура:

1. Выбор шаблона. После выбора шаблона развертывания виртуальной машины есть возможность дополнительно настроить гостевую операционную систему и оборудование виртуальной машины.

2. (Необязательно) Настройка гостевой операционной системы.

3. (Необязательно) Настройка аппаратного обеспечения этой виртуальной машины.

4. (Необязательно) Включение виртуальной машины сразу после завершения ее создания.

5. Выбор имени и папки виртуальной машины

6. Выбор ресурса.

7. Выбор хранилища данных. Тут есть несколько вариантов:

- Тот же формат, что и у источника - использует то же хранилище, что и исходная машина.
- Толстая настройка Lazy Zeroed - создание виртуального диска по умолчанию, при котором место для него выделяется сразу при создании, но очищается только при первой записи с виртуальной машины.

- Толстая настройка Eager Zeroed - создание виртуального диска по умолчанию, выделение место и его стирание данных происходит сразу при создании.
- Тонкая настройка - виртуальный диск использует только столько пространства хранилища данных, сколько ему нужно изначально, но при необходимости может увеличивать до максимальной емкости, выделенной ему.

8. Завершение создания виртуальной машины из шаблона. Перед развертыванием виртуальной машины, можно просмотреть настройки виртуальной машины.

3.2.3. Клонирование существующей виртуальной машины

Клонирование виртуальной машины создает виртуальную машину, которая является копией оригинала. Новая виртуальная машина настроена с тем же виртуальным оборудованием, установленным программным обеспечением, а также другими свойствами, которые были сконфигурированы для исходной виртуальной машины.

Процедура:

1. Запуск клонирования.
2. Выбор виртуальной машины для клонирования.
3. Выбор имени виртуальной машины и папки для ее хранения.
4. Выбор ресурса: хост, кластер, vApp или пул ресурсов, где будет запускаться ВМ.
5. Выбор хранилища данных.
6. Выбор настроек клонирования. Возможно выборочно настроить гостевую ОС, настроить оборудование виртуальной машины, включить ВМ после ее создания.
7. Завершение клонирования. Перед развертыванием виртуальной машины, можно просмотреть настройки виртуальной машины.

3.3. Виртуальные ресурсы

Большинство свойств виртуальных ресурсов виртуальной машины можно добавить или настроить во время процесса создания виртуальной машины или после создания виртуальной машины и установки гостевой операционной системы в настройках.

Не все устройства доступны для каждой виртуальной машины. Хост, на котором работает виртуальная машина, и гостевая операционная система должны поддерживать устройства, которые необходимо добавить, или конфигурации, которые нужно внести.

3.3.1. Виртуальный процессор

Для повышения производительности виртуальной машины можно добавлять, изменять и настраивать ресурсы ЦП. VMware использует следую-

щую терминологию. Понимание этих терминов может помочь вам спланировать стратегию распределения ресурсов CPU:

ЦПУ – процессор или процессор - это часть компьютерной системы, которая выполняет команды компьютерной программы и является основным элементом, выполняющим функции компьютера. Процессор содержат ядра.

Разъем процессора – физический разъем на материнской плате компьютера, который принимает один физический процессор. Многие материнские платы могут иметь несколько сокетов, которые, в свою очередь, могут принимать многоядерные процессоры (CPU). VSphere Web Client вычисляет общее количество виртуальных сокетов из числа ядер и ядер на сокет, который вы выбираете.

Core (ядро) – содержит блок, каждый из которых отвечает за выполнение одного потока команд. Ядра могут независимо запускать программы или потоки. Одно или несколько ядер могут существовать на одном процессоре.

Corelet – ядро процессора AMD архитектурно эквивалентно логическому процессору. В отличие от традиционного процессорного ядра, у корелетов отсутствует полный набор частных выделенных ресурсов выполнения, и они совместно используют некоторые ресурсы выполнения с другими ядром.

Потоки – некоторые ядра могут одновременно запускать независимые потоки инструкций. В существующих реализациях ядра могут запускать один или два потока программного обеспечения за один раз, мультиплексируя функциональные блоки ядра между программными потоками, если это необходимо. Такие ядра называются двойными или многопоточными.

Обмен ресурсами - происходит определение приоритета или важности виртуальной машины или пула ресурсов. Если виртуальная машина имеет вдвое больше заданных ресурсов, чем другая виртуальная машина, она имеет право потреблять вдвое больше этого ресурса, когда эти две виртуальные машины конкурируют за ресурсы.

Распределение ресурсов - при необходимости будет автоматически меняться параметры распределения ресурсов процессора, такие как общие ресурсы, резервирование и ограничение, когда доступная емкость ресурсов не соответствует требованиям.

Виртуальная симметричная многопроцессорная обработка vSphere (Virtual SMP) - функция, которая позволяет одной виртуальной машине иметь несколько процессоров.

3.3.2. Память

Параметры ресурса памяти для виртуальной машины определяют, какая часть памяти хоста выделяется виртуальной машине. Размер виртуальной аппаратной памяти определяет, сколько памяти доступно для прило-

жений, запущенных на виртуальной машине. Виртуальная машина не может использовать большее количество ресурсов памяти, чем ее настроенный объем виртуальной аппаратной памяти. Узлы ESXi ограничивают использование ресурсов памяти максимальным количеством, полезным для виртуальной машины, поэтому вы можете принять значение по умолчанию для ресурсов неограниченной памяти.

3.3.3. Сетевой контроллер

Сетевые функции ESXi обеспечивают связь между виртуальными машинами на одном хосте, между виртуальными машинами на разных хостах, а также между другими виртуальными и физическими машинами. Сетевые функции также позволяют управлять хостами ESXi и обеспечивают связь между службами VMkernel (NFS, iSCSI или vSphere vMotion) и физической сетью. При настройке сети для виртуальной машины вы выбираете или изменяете тип адаптера, сетевое подключение и подключаете ли вы сеть при включении виртуальной машины.

3.3.4. Конфигурация виртуального диска

Добавить виртуальные диски к виртуальным машинам и добавить дополнительное пространство к существующим дискам, возможно даже когда виртуальная машина работает. Большинство параметров виртуального диска устанавливается во время создания виртуальной машины или после установки гостевой операционной системы.

Данные виртуальной машины можно хранить на новом виртуальном диске, существующем виртуальном диске или подключенном SAN LUN. Виртуальный диск, который отображается как один жесткий диск для гостевой операционной системы, состоит из одного или нескольких файлов в файловой системе хоста. Вы можете копировать или перемещать виртуальные диски на одном и том же хосте или между хостами.

Для виртуальных машин, работающих на хосте ESXi, хранить данные виртуальной машины можно непосредственно на SAN LUN, а не в файле виртуального диска. Эта возможность полезна, если приложения запускаются на своих виртуальных машинах, которые должны определять физические характеристики запоминающего устройства.

Когда вы сопоставляете LUN с томом VMFS, vCenter Server или хост ESXi создает файл raw mapping (RDM), указывающий на необработанный LUN. Инкапсуляция информации о диске в файл позволяет vCenter Server или хосту ESXi блокировать LUN, чтобы только одна виртуальная машина могла писать на него. Этот файл имеет расширение .vmdk, но файл содержит только информацию о диске, которая описывает сопоставление с LUN в системе ESXi. Фактические данные хранятся в LUN. Вы не можете развернуть виртуальную машину из шаблона и сохранить ее данные на LUN. Вы можете хранить только свои данные в файле виртуального диска.

Количество свободного места в хранилище данных постоянно меняется. Убедитесь, что вы оставляете достаточно места для создания виртуальной машины и других операций виртуальной машины. Тонкая настройка позволяет создавать разреженные файлы с блоками, которые выделяются при первом доступе, что позволяет перенаправить хранилище данных. Разреженные файлы могут продолжать расти и заполнять хранилище данных. Если хранилище данных заканчивается на диске во время работы виртуальной машины, это может привести к остановке работы виртуальной машины.

4. Виртуальная сеть

4.1. Основы и объекты виртуальной сети.

При использовании физического сервера, с установленной ОС, на нем для подключения к сети настраивается физические сетевые контроллеры, ip-адреса, VLAN. В случае же использования виртуальных машин, возникает проблема, ведь с помощью одного физического контроллера должны работать несколько виртуальных машин. Именно поэтому, физический сетевой контроллер не является "активным сетевым устройством", т.е. у него нет своего IP-адреса, а MAC-адрес фигурирует только в технических сообщениях. А вот для виртуальных машин и себя самого гипервизор создает виртуальные сетевые контролеры (VMkernel). Таким образом, физический сетевой контроллер - связь с внешней сетью, через который могут подключаться все имеющиеся виртуальные контроллеры.

Связующим звеном между виртуальными и физическими сетевыми контроллерами являются виртуальные коммутаторы. Схема связи виртуальных объектов с физическими показана на рисунке 5.

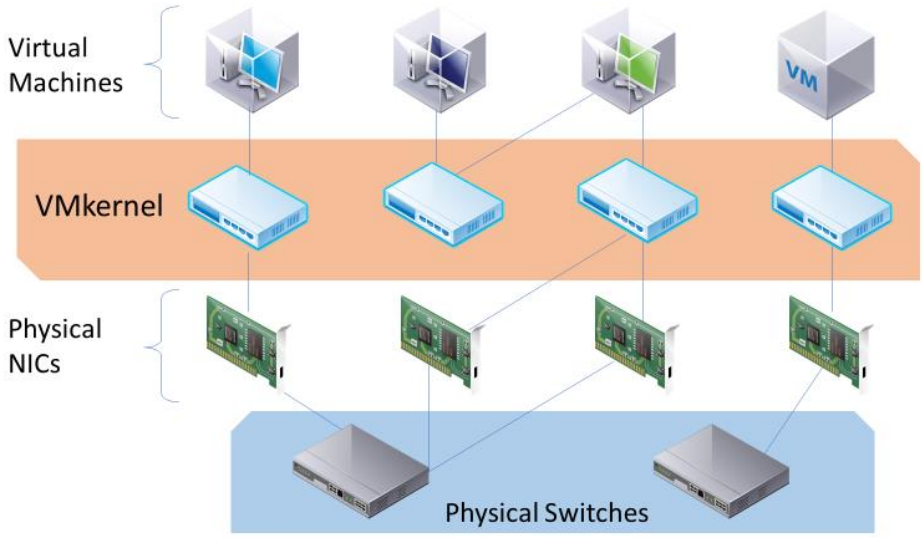

Рис.5. Связь между объектами сети «внутри» ESXi

К объектами виртуальной сети относятся:

- Физические сетевые контроллеры (networkinterfacecard, NIC) - контроллеры, установленные в сервере. Гипервизор дает им имена вида vmnic#. Они так же называются "канал во внешнюю сеть" и "аплинк";
- Виртуальный коммутатор (vSwitch);
- Группы портов (Portgroups) - логические объекты, создоваемые на виртуальных коммутаторах, к которым подключаются виртуальные сетевые контроллеры;
- Виртуальные сетевые контроллеры.

В клиенте VSphere связь между объектами виртальной сети показана очень наглядно, отображены группы портов (VirtualMachinePortGroup), виртуальные сетевые контроллеры и связанные с помощью виртуальных коммутаторов (серые прямоугольники) физические сетевые контроллеры (PhysicalAdapters) (см. рис.6).

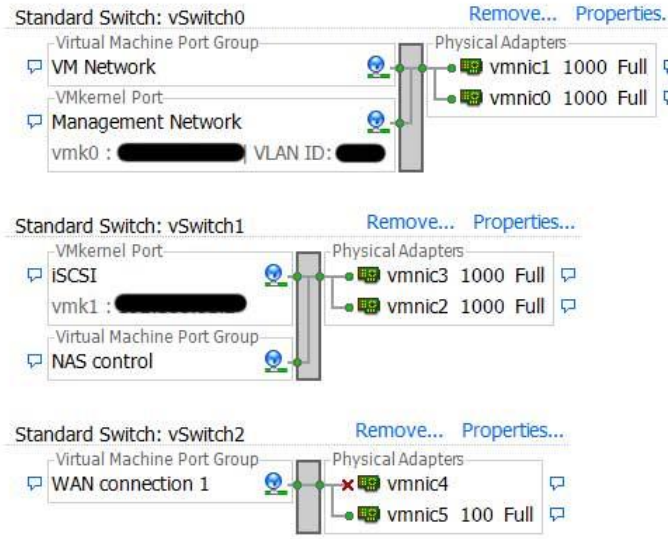

Рис.6 Отображение связей между элементами в интерфейсе VSphere.

4.2. Физический сетевой контроллер

Физические контроллеры используются исключительно как каналы во внешнюю сеть, то, как говорилось ранее, они не имеют своего IP-адреса, настроить на них можно только скорость и дуплекс. Так же, физические адаптеры должны быть привязаны к виртуальным коммутаторам. При этом, к одному физическому контроллеру может быть привязан только один vSwitch, но к одному виртуальному свичу может быть привязанно несколько сетевых коммутаторов, повышающих откаузойчевость и производительность сети.

В связи с ограниченностью функционала данных элементов сети и невозможность подключнеия к ним, возможности применения каких-либо способов защиты виртуальной сети нет. Поэтому перейдем к следующему элементу.

4.3. Виртуальный сетевой контроллер VMkernel

Виртуальный сетевой контроллер может принадлежать виртуальной машине или самому гипервизору (VMkernel).

Функции виртуальных сетевых интерфейсов гипервизора (vmk#):

- Обращение по IP-адресу виртуального сетевого контроллера VMkernel для управления гипервизором;

- Передача содержимого оперативной памяти при переносе виртуальной машины (vMotion);
- Подключение дисковых ресурсов iSCSI и NFS;
- Передача процессорных инструкций на резервную виртуальную машину при использовании FaultTolerance.

Виртуальных сетевых контроллеров управления может быть несколько, они могут быть привязаны к нескольким физическим сетевым котроллерам, что позволяет обеспечить высокую отказоустойчивость системы. Эти элементы так же не отвечают за безопасность виртуальной сети, поэтому нет необходимости придумывать какие-то способы защиты с использованием сетевых контроллеров гипервизора.

4.4. Виртуальные коммутаторы vSwitch.

Ключевыми компонентами виртуальной сети являются виртуальные коммутаторы vSwitch. Это L2-коммутаторы, которые обеспечивают связь между виртуальными сетевыми контроллерами и физическими адаптерами.

VSwitchбывают двух видов:

- StandardvSwitch - стандартный виртуальный коммутатор;
- vNetworkDistributedvSwitch (vDS) - распределенный виртуальный коммутатор.

Виртуальные коммутаторы работают под управлением гипервизора и отвечают за все сетевые операции хоста, в том числе они обеспечивают прохождение управляющего трафика. Все остальные сетевые компоненты подключаются с помощью Portgroup (см. рис. 7).

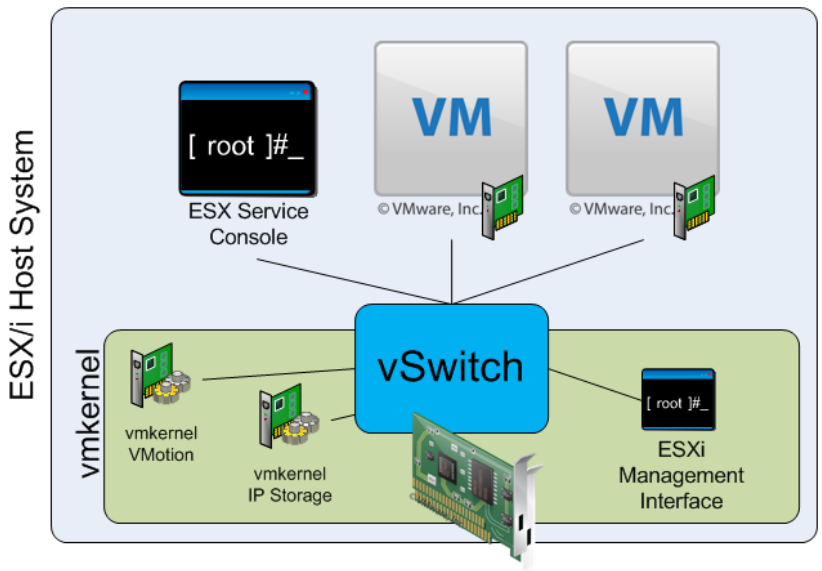

Рис. 7. Использование vSwitch.

В отличии от физических L2-коммутаторов, vSwitch не изучает MAC-адреса проходящего трафика, не участвует в SpanningTree и не может со-

здать сетевой петли, т.к. нет прямого соединения между свичами, взаимодействие происходит только через внешнюю сеть.

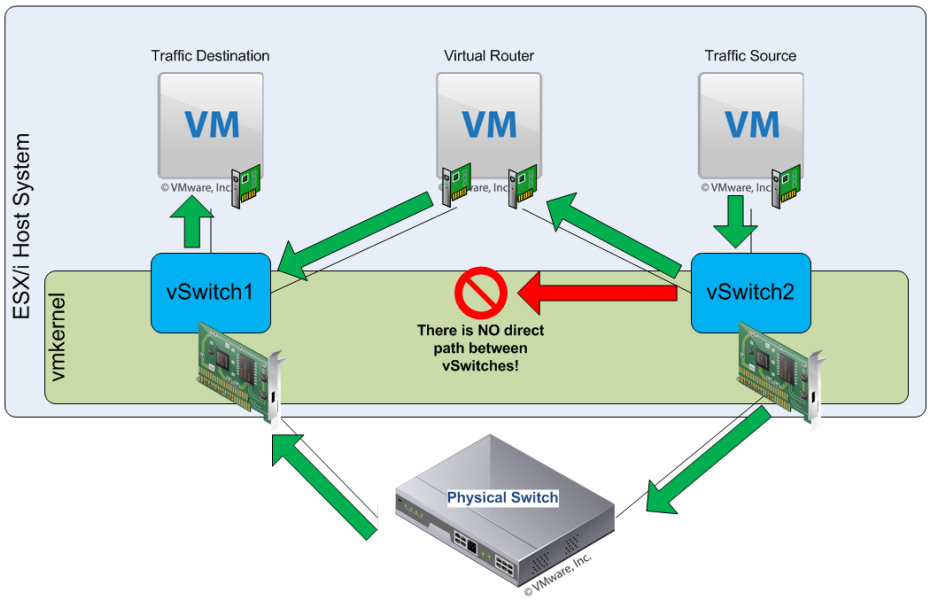

Рис. 8. Схема взаимодействия между vSwitch.

vSwitch работает на канальном уровне сетевой модели OSI. Как и любое другое устройство второго уровня, он отвечает за доставку пакетов соседним устройствам, но не производит маршрутизацию, в следствии чего, при возникновении необходимости осуществления пересылки пакетов в сегмент сети, напрямую не подключенный к vSwitch, придется использовать дополнительное устройство, выполняющее роль маршрутизатора (виртуальное или физическое).

Одно из главных отличий, что виртуальных сетяхvSwitch берет на себя обеспечение отказоустойчивости (не требуется никаких дополнительных сетевых интерфейсов, как в физической сети) (см. рис. 9). Это сокращает количество используемых потов физического коммутатора с 8 до 2-х.

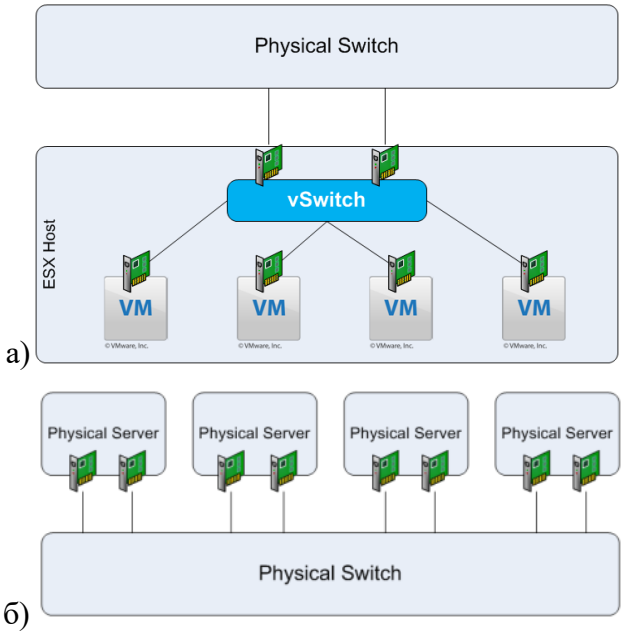

Рис. 9. Обеспечение отказоустойчивости в а) виртуальных сетях б) физических сетях.

4.5. PortGroup

PortGroup - это логические объекты, аналог объединения портов в группы в физических сетях. При настройке реального коммутатора на портах настраивается ограничение пропускной способности, группы портов добавляются в VLAN, в виртуальной же сетевой инфраструктуре пропускная способность и VLAN указывается для группы портов, а вносить количество портов, входящих в группу, и их номера не нужно. Количество виртуальных портов, входящих в группу будет расти автоматически с добавлением ВМ, они не ограничены, но ограничено количество портов на весь vSwitch, т.е. на количество виртуальных сетевых коммутаторов.

4.6. Распределенный коммутатор vNetworkDistributedvSwitch

Сетевой коммутатор в vSphere состоит из двух логических разделов: плоскости данных и плоскости управления. Плоскость данных реализует коммутацию пакетов, фильтрацию, тегирование, и так далее. Плоскость управления является структурой управления, которая используется для настройки функциональных возможностей плоскости данных. vSphereStandardSwitch содержит плоскость данные и управления, в следствии чего необходимо настраивать и поддерживать каждый стандартный коммутатор в индивидуальном порядке.

vSphereDistributedSwitch разделяет плоскость данных и плоскость управления. Управляется распределённый коммутатор с помощью системы vCenterServer, которая позволяет управлять сетевой конфигурацией вашей среды на уровне ЦОД. Плоскость данных остается локально на каждом хо-

сте, который связан с распределенным коммутатором. Секция плоскости данных распределенного коммутатора называется hostproxyswitch. Конфигурация сети, которую вы создаете на vCenterServer (плоскость управления) автоматически применится на всех hostproxyswitches (плоскость данных).

При помощи dvSwitch можно создавать последовательные конфигурации сети для физических интерфейсов, виртуальных машин и услуг VMkernel. Распределенный виртуальный коммутатор решает эту проблему, тем, что это один логический коммутатор, работающий на всех серверах. Это упрощает настройку политики безопасности и позволяет при перемещении виртуальной машины с сервера на сервер оставаться ей на прежнем порту распределенного коммутатора.

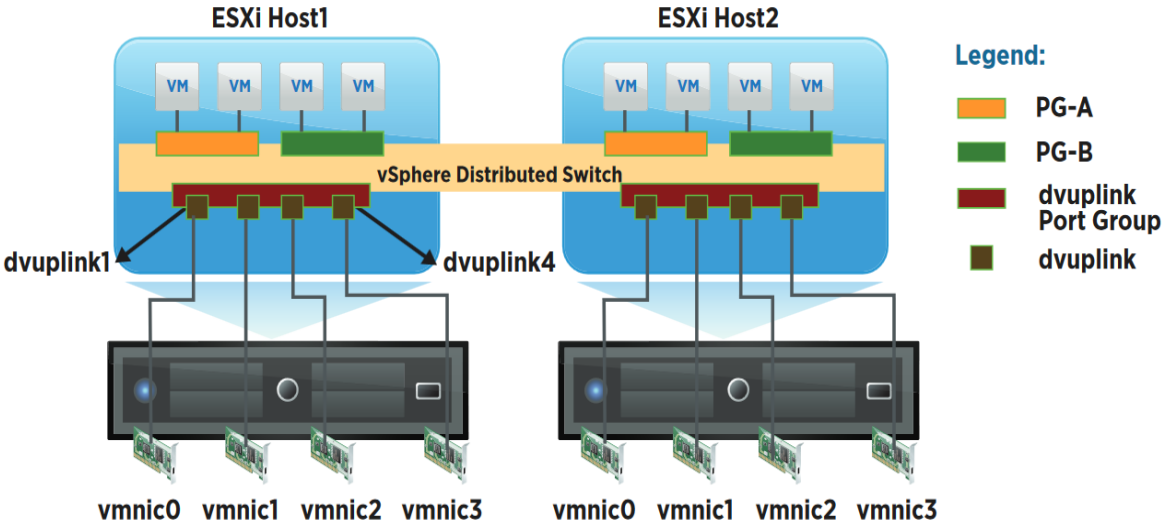

Рис. 10. Использование dvSwitch

Uplinkportgroup или dvuplinkportgroup определяется во время создания распределенного коммутатора и может иметь один или несколько uplinks.Uplink представляет собой шаблон, который используется для настройки физических соединений узлов сети, а также восстановления после сбоев и балансировки нагрузки политик. На уровне хоста, каждый физический сетевой интерфейс подключен к порту uplink с определенным идентификатором. Если задать политики балансировки нагрузки и отказоустойчивости через uplink, то данные политики автоматически распространятся на hostproxyswitches или плоскость данных. Таким образом можно осуществить переход на другой ресурс и загрузить конфигурацию на NICs всех хостов, связанных распределенным коммутатором.

Распределенные группы портов обеспечивают подключение к сети виртуальных машин и предоставление VMkernel трафика. Они идентифицируются, используя сетевую метку, уникальную для текущего центра обработки данных. На объединённых NICs задаются нужные политики: балансировка нагрузки, VLAN, безопасность, trafficshaping и т. д.. Uplinkport-

groups и настройки распределенных групп портов на vCenterServer автоматически распространяются на все хосты распределенного коммутатора через их hostproxyswitches. Таким образом можно настроить группу виртуальных машин и распределить эти настройки на остальные виртуальные машины той же распределенной группы портов.

Преимущества dvSwitch:

- Централизованное управление;
- Private VLAN - расширение стандарта VLAN, позволяющее дополнительно ограничить видимость (здесь) виртуальных машин внутри одного VLAN;
- Двухсторонний trafficshaping;
- Расширенные возможности использования VLAN trunk;
- Network IO Control - возможно гибко настраивать пропускную способность;
- Балансировка нагрузки между физическими сетевыми контроллерами;
- LinkLayerDiscoveryProtocol(LLDP) - протокол сбора данных о сетевых устройствах;
- NetFlow - сбор статистики трафика;
- QoS (802.1p)- поддержка тэгов приоретизации трафика.

5. Сеть хранения данных SAN (StorageAreaNetwork)

Основной движущей силой, стимулирующей развитие сетей хранения данных, является непрекращающийся рост объемов информации, к которой требуется высокоскоростной доступ. Современные технологии хранения данных позволяют объединить отдельные массивы в высокопроизводительные сети.

Сеть хранения данных SAN — это система, которая позволяет организовать распределенный доступ к устройствам хранения данных между серверами и рабочими станциями, независимая от локальной и беспроводной сети.

Основой данной сети является протокол FibreChannel. Передача данных межу серверами-коммутаторами осуществляется при помощи оптических каналов связи. Это позволяет обеспечить высокую готовность систем для станций с большой интенсивностью запросов.

Более прогрессивная и функциональная технология SAN все чаще вытесняет традиционные, и позволяет во много раз увеличить производительность и надежность системы хранения данных. К недостаткам, пожалуй, можно отнести только высокую стоимость системы.

Технологически SAN состоит из следующих компонентов:

1. Узлы, ноды (nodes)
- Дисковые массивы (системы хранения данных) — хранилища (таргеты [targets]),
- Серверы — потребители дисковых ресурсов (инициаторы [initiators]).

2. Сетевая инфраструктура
- Коммутаторы (и маршрутизаторы в сложных и распределённых системах),
- Кабели.

Транспортную основу SAN составляет протокол FibreChannel, использующий как медные, так и волоконно-оптические соединения устройств.

Преимущества использования SAN на базе FibreChannel:

- Масштабируемость (сети хранения данных обладают способностью к масштабированию в аспекте объема и скорости передачи данных);
- Сегрегация хранилищ (хранилище отделено от сервера приложений и в то же время обеспечивает защиту и целостность данных, например, можно разделить данные отделов, которые физически будут находиться в одном пуле устройств хранения данных);
- Централизация и управление хранилищем (сети хранения данных позволяют консолидировать устройства хранения, оптимизировать их использование и управление. Речь идет о том, что дублировать данные, которые размещены вне инфраструктуры SAN, нет необхо-

димости. Еще одним преимуществом является возможность распределения хранилищ. Это позволяет избежать ситуации, когда у одного сервера объем хранилища избыточен, а второму серверу объема не хватает);

- Поддержка устаревших устройств;
- Поддержка большего количества устройств;
- Расстояние (шина SCSI поддерживает расстояния между устройствами хранения и узлами порядка десятков метров, а FibreChannel — десятков километров).

5.1. FC (FiberChannel)

Интерфейс **FibreChannel** — это технология межсистемного взаимодействия, которая объединяет в себе возможности высокоскоростного ввода-вывода и сетевого обмена данными. Основывается на использовании центрального хранилища данных и расширения существующих технологических функций.

В терминологии FibreChannel устройства называются узлами (nodes). Каждый узел FibreChannel имеет уникальное 64-разрядное имя WWN (WorldWideName), которое назначается производителем.

5.1.1. Топологии FibreChannel

1. Топология “точка–точка” (point-to-point)

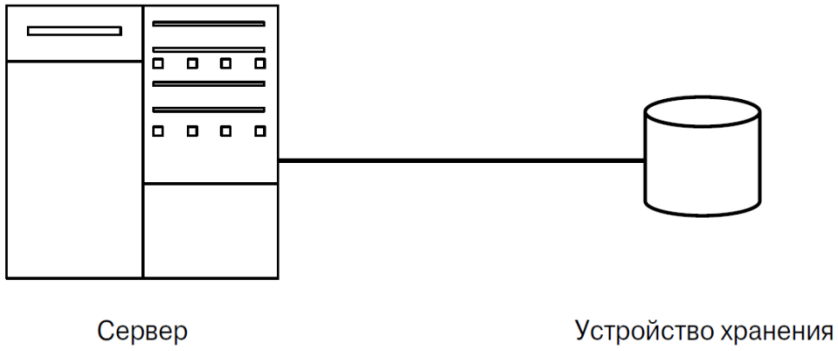

Рис.11 Топология “точка–точка”

В этом случае сервер обычно подключается к выделенной подсистеме хранения, причем данные не используются совместно.

Для реализации топологии “точка–точка”, как минимум, необходимы:

- сервер, адаптер FibreChannel (адаптер шины),
- устройство хранения (например, жесткий диск или накопитель на магнитной ленте), оснащенное интерфейсом FibreChannel.

2.Кольцо с разделяемым доступом (arbitrated-loop)

Кольцо — это схема логического подключения устройств, при котором данные передаются по логически замкнутому контуру. В кольце с раз-

делением доступа (arbitrated-loop) протокол описывает порядок, в котором узел получает разрешение на передачу данных.

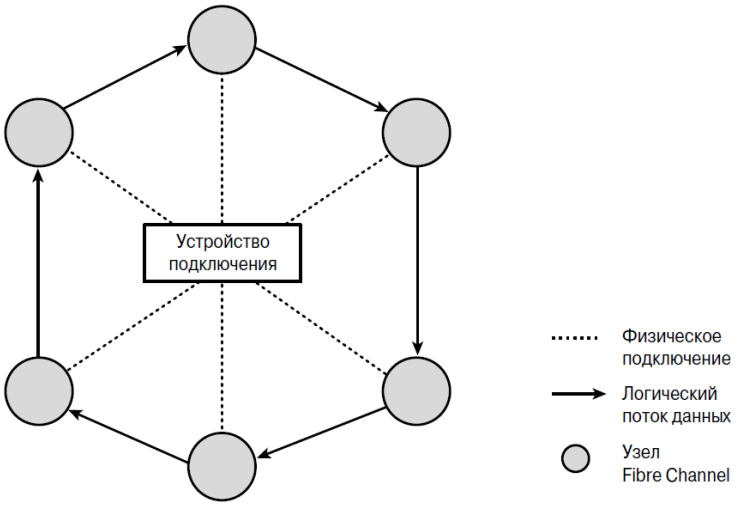

Рис.12 Топология “кольцо с разделяемым доступом”

Кольцо FibreChannel с разделением доступа (FibreChannel-arbitrated-loop — FC-AL) может быть реализовано на базе различных устройств хранения (жестких дисков, накопителей на магнитной ленте), серверов, адаптеров шины и устройств для их подключения. В качестве устройств подключения могут выступать концентраторы или коммутаторы FibreChannel.

Команды FibreChannel поддерживают согласование и доступ к кольцу для передачи данных. Кроме того, предоставляются команды для назначения адресов портов кольца с разделением доступа (arbitrated-loop-port-addresses — AL-PA) различным узлам кольца. Каждый узел в управляемом кольце FibreChannel имеет контур для собственного отключения от кольца и сохранения непрерывности кольца в случае ошибки.

Кольца FibreChannel с разделением доступа могут адресовать до 127 портов. Один из этих портов зарезервирован для подключения к коммутатору связной архитектуры, остальные 126 портов доступны для предоставления узлам.

3. Коммутируемая связная архитектура (switched-fabric)

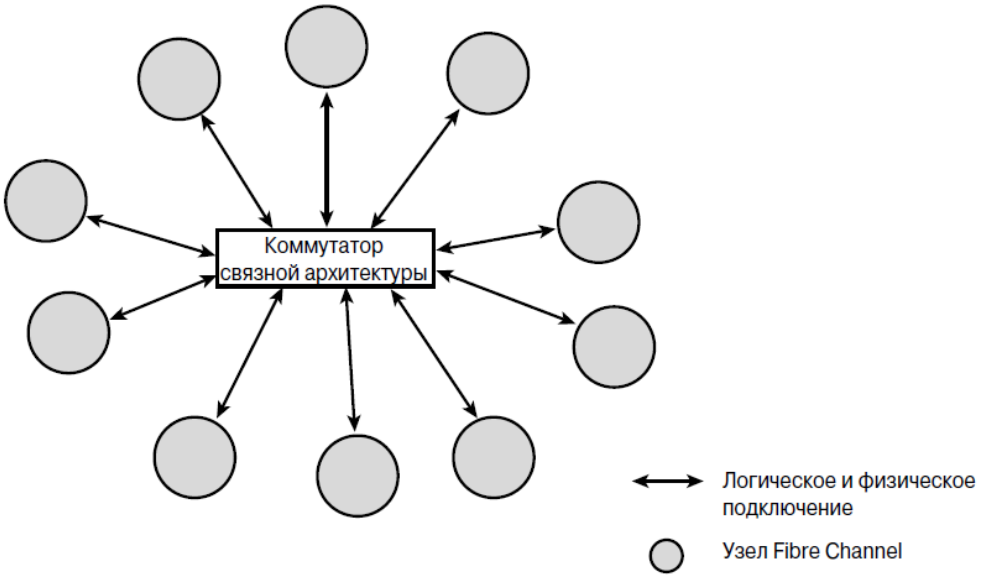

Рис.13 Коммутируемая связная архитектура

В топологии коммутируемой связной архитектуры FibreChannel (FibreChannelswitched-fabric) каждое устройство имеет логическое подключение к любому другому устройству. Обеспечение физического подключения устройств по топологии “каждый с каждым” потребовало бы огромных затрат, так как для N устройств необходимо N^2 портов и физических подключений. В реальности каждое устройство подключается к коммутатору, а коммутатор поддерживает логические подключения между всеми своими портами.

Несколько узлов (устройств хранения и компьютерных систем) подключены к коммутатору FibreChannel. Коммутатор —это высокоскоростное устройство, которое обеспечивает подключение по схеме “каждый с каждым” и обрабатывает несколько одновременных подключений. Коммутаторы могут быть подключены каскадно (в виде иерархии) или в виде сети, что позволяет формировать более сложные конфигурации. Очень часто строится трехуровневая иерархия, на самом нижнем уровне которой размещены кольца с разделением доступа FibreChannel), подключенные к малопроизводительным коммутаторам. Эти коммутаторы, в свою очередь, подключаются к высокоскоростным коммутаторам, обеспечивающим максимально возможную пропускную способность.

5.1.2. Протокол FibreChannel

FibreChannel — это набор стандартов, разработанных в Национальном институте стандартизации США. Интерфейс FibreChannel предоставляет высокопроизводительное последовательное подключение между хостом и единицами хранения, а также между самими единицами хранения. Стандарт позволяет обеспечить высокоскоростную передачу данных в се-

тях с топологией “точка–точка” и кольцо. Более того, FibreChannel предоставляет все эти возможности вместе с проверкой ошибок.

В стандарте FibreChannel определено пять функциональных уровней: от FC-0 до FC-4:

Уровень FC-0

Определяет физические характеристики интерфейса и носителя. В частности, посредством FC-0 определяются спецификации уровней сигналов, носителя и получателей/отправителей. Уровень FC-0 позволяет использовать несколько интерфейсов, что дает возможность выбирать разные скорости передачи данных и различные передающие среды.

Уровень FC-1

Определяет схемы кодирования и декодирования данных, сигналов и специальных символов, а также управление ошибками. Кроме того, уровень FC-1 отвечает за обслуживание линий связи. Уровень FC-1 использует схему кодирования, которая называется 8B/10B.

Схема проектировалась для того, чтобы обеспечить следующее:

- эффективную синхронизацию данных;
- расширенное обнаружение ошибок;
- эффективное обнаружение управляющих символов;
- упрощенное проектирование аппаратного обеспечения приемников/передатчиков.

Данные FibreChannel всегда передаются группами по 4 байта, которые называются словами передачи (transmissionwords).

Уровень FC-2

Определяет передачу данных от одного узла к другому, т.е. непосредственно транспортный механизм. Уровень FC-2 формирует кадры, определяет классы обслуживания и службы регистрации связной архитектуры или портов.

Уровень FC-2 определяет:

- иерархию передачи данных FibreChannel, которая включает в себя упорядоченные множества, кадры, последовательности и обмены;
- управление потоком FibreChannel;
- протоколы FC-2;
- классы обслуживания FC-2.

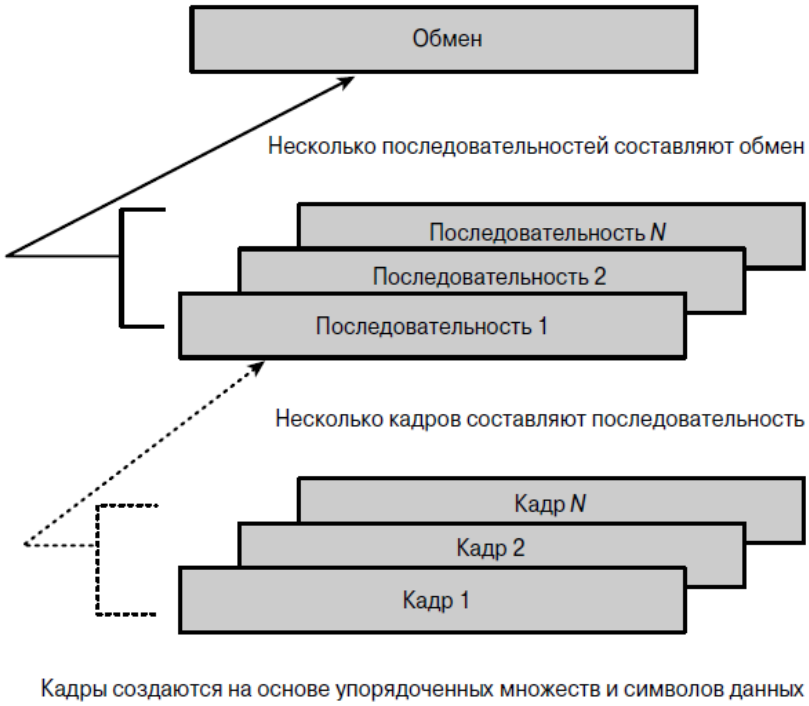

Рис.14Иерархия передачи данных FibreChannel

FibreChannel данные передаются с помощью кадров (frame). Кадры создаются из упорядоченных множеств и символов данных. Несколько кадров группируются вместе и формируют последовательность, а несколько последовательностей формируют обмен (exchange).

Уровень FC-3

Определяет общие службы, включая службы по управлению и общему транспортному механизму. Уровень FC-3 — общий для всех портов узла. Уровни FC-1, FC-2 и FC-4 реализованы отдельно для каждого порта.

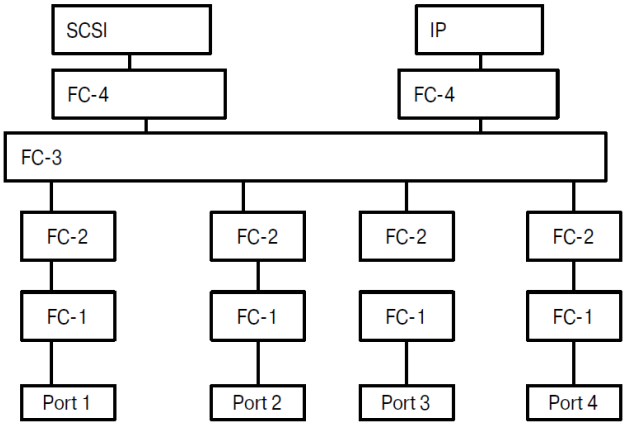

Рис.15 Уровни FC

Некоторые функции, реализованные на уровне FC-3:

- Транкинг (trunking) или канальное уплотнение (striping), при котором параллельные линии связи и порты “сворачиваются” для обеспечения большей пропускной способности между узлами.
- Многоабонентская доставка, при которой единая передача данных может быть направлена одновременно к нескольким портам.
- Свободный поиск (hunting), при котором несколько портов используют одинаковый псевдоним.

Уровень FC-4

Определяет связывание протоколов верхнего уровня с FibreChannel:

- SCSI;
- IP;
- IPI (Intelligent Peripheral Interface);
- HIPPI (High-Performance Parallel Interface);
- IEEE 802.2;
- SBCCS (Single-Byte Command Code Sets);
- AAL5 (ATM AdaptationLayer);
- FC-LE (LinkEncapsulation).

5.2. Fibre-Channel-over-Ethernet

Fibre-Channel-over-Ethernet, или FCoE, это новый протокол (транспортный уровень), определенный стандартом в комитетеT11. FCoE переносит фреймы Fibre ChannelчерезEthernet, инкапсулируя кадрыFibre Channel в jumbo-frames-Ethernet-a.

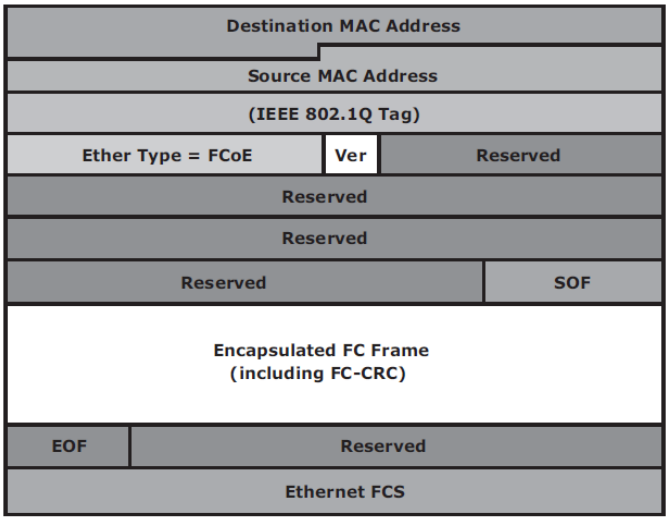

Рис. 16 Кадр FCoE

Кадр FCoE представляет собой кадр Ethernet, который содержит блок данных протокола FCoE. Первые 48-бит в кадре используются для указания MAC-адреса назначения, а следующие 48 бит указывают MAC-адрес ис-

точника. 32-битный IEEE 802.1Q тег поддерживает создание нескольких виртуальных сетей (VLAN) через единую физическую инфраструктуру. Все FCoE фреймы используют уникальный EtherType (0×8906), который позволяет коммутатору понять, что к нему пришел не просто Ethernet, а FC. Следующие 100-биты зарезервированы. Далее следуют 8-бит начала кадра, а затем сам FC-фрейм. 8-бит - конец кадра. Далее 24 зарезервированных бита. Кадр заканчивается 32 битами, посвященных функции FCS, которая обеспечивает обнаружение ошибок для кадра Ethernet.

Инкапсуляция кадра FC происходит через mapping FC кадров в кадры Ethernet. Спецификация протокола FCoE заменяет уровни FC-0 и FC-1 на Ethernet. Это обеспечивает возможность нести уровень FC-2 к уровню FC-4 по Ethernet.

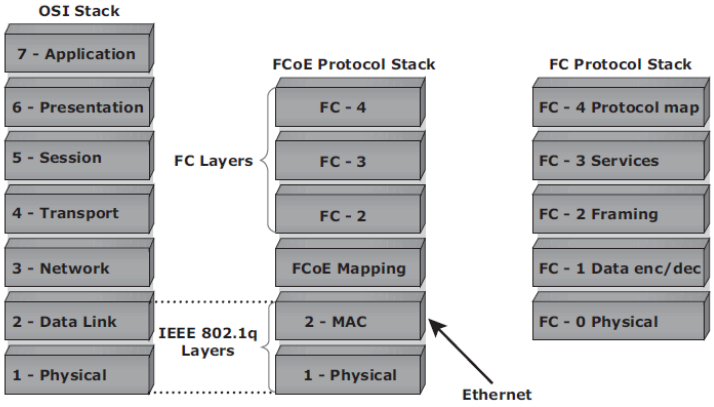

Рис.17 Принцип работы FCoE

ОбщепринятыйEthernetпроходит с потерей данных, т.е. фреймы могут быть отброшены или потеряны во время передачи. ConvergedEnhancedEthernet (CEE), или Ethernetбез потерь, обеспечивает новую спецификацию к существующему стандарту Ethernet, который исключает потери.

Ethernet без потерь требует определенных функциональных возможностей:

Приоритет на основе управления потоком (PFC)

FC управляет загруженностью на канальном уровне путем использования credit-basedflow (без потерь).Ethernetиспользует dropflowcontrol механизм, который при загруженности отбрасывает фреймы. Эта проблема устраняется с помощью кадра управления Ethernet PAUSE (IEEE 802.3x)для создания CEE.

Получатель может послать отправителю PAUSE request, когда буфер получателя переполняется. После получения этого фрейма отправитель прекращает передачу фреймов. Недостатком использования данного метода, является то, что он действует на всей линии связи.

PFC создает восемь отдельных виртуальных каналов на одной физической линии связи и позволяет любому из этих каналов, быть приостановленным и перезапущенным независимо друг от друга. PFC обеспечивает механизм паузы, основываясь на приоритетах пользователей или классов обслуживания.

Расширение выбора передачи (ETS)

Усовершенствованный выбор передачи обеспечивает общую структуру управления для назначения полосы пропускания для различных классов трафика, таких как LAN, SAN и InterProcessCommunication (IPC). Когда конкретный класс трафика не использует свою выделенную полосу пропускания, ETS позволяет другим классам трафика, чтобы использовать доступную полосу пропускания.

Уведомление о перегрузке (CN)

Уведомление о перегрузке обеспечивает управления перегрузками из конца в конец для протоколов, которые не имеют встроенных механизмов управления перегрузкой. CNобеспечивает механизм обнаружения и оповещения о загрузках для перемещения потока трафика. Уведомления о загруженности позволяет коммутатору посылать сигнал к другим портам, которые должны остановить или замедлить их передачи.

5.3. Протокол iSCSI

Протокол iSCSI (Internet SCSI) позволяет реализовать одно или несколько подключений TCP/IP между устройствами, обменивающимися командами SCSI, ответами и информацией о состоянии. Другими словами, iSCSI представляет собой протокол для соединения «точка-точка», инкапсулирующий команды SCSI, ответы и информацию о состоянии.

Основные компоненты iSCSI:

- Initiator (host)
- Target (storage or iSCSI gateway)
- IP-based network

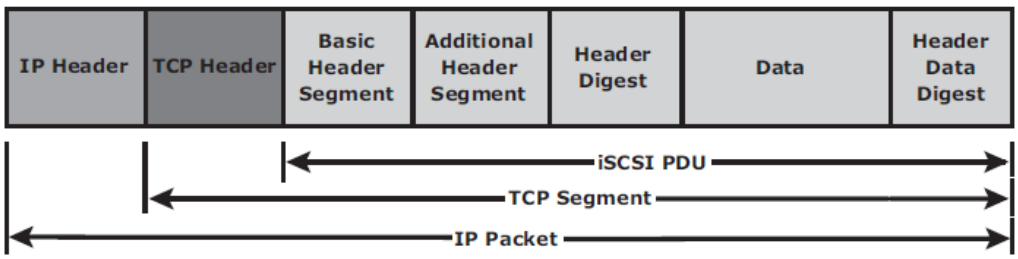

Рис. 18 Инкапсуляция протокола iSCSI

5.3.1. Инкапсуляция протокола iSCSI

Компоненты IP, TCP, iSCSI и SCSI работают вместе в процессе инкапсуляции. Пакет iSCSI содержит данные и команду SCSI для стека TCP/IP. Заголовок iSCSI содержит информацию об извлечении и интерпретации команды SCSI, которая размещена в пакете. Заголовок пакета TCP отвечает за гарантированную и последовательную доставку пакетов. Пакет TCP содержит данные и полезную нагрузку (payload) пакета IP. Заголовок IP используется в процессе маршрутизации.

5.3.2. Уровни протокола iSCSI

Протокол iSCSI размещен поверх существующих уровней протоколов TCP, IP и низкоуровневых аппаратных протоколов, поддерживающих TCP/IP (например, Ethernet и GigabitEthernet). SCSI – это протокол уровня приложения. Протокол iSCSI предоставляет услуги протоколу приложений SCSI и применяет TCP/IP для обеспечения гарантированной доставки, маршрутизации и т.д.

Все устройства iSCSI (целевые и устройства-инициаторы) имеют два разных имени. Адрес iSCSI, который состоит из адреса IP, порта TCP и имени iSCSI и имеет формат “<имя домена>: <номер порта>:<имя iSCSI>”. Имя iSCSI, которое имеет наглядный формат, например «Hmh_FQN. Поставщик_диска. Модель _диска. Номер».

Протокол iSCSI устанавливает сеансы связи между инициатором и целевым устройством. Один или несколько сеансов протокола TCP могут использоваться одним или несколькими сеансами iSCSI. После установки сеанса две стороны (инициатор и целевое устройство) обмениваются такими параметрами, как безопасность, размер буфера и возможность отправки незапрошенных данных. Сеанс iSCSI может закрываться стандартным образом: посредством завершения регистрации или в связи с возникновением ошибки. Независимо от количества использованных сессий TCP, протокол iSCSI гарантирует, что команда SCSI и ответы на нее будут доставлены в правильном порядке. Обратите внимание: протокол TCP гарантирует последовательную доставку для определенного сеанса TCP, но не обеспечивает синхронизации передаваемых данных между различными сеансами TCP. Таким образом, синхронизации сеансов TCP возлагается на протокол iSCSI.

Можно перечислить ряд требований к протоколу iSCSI:

- Разные команды SCSI могут передаваться через различные сеансы TCP;
- Все данные и параметры, соответствующие определенной команде, должны передаваться в рамках того же сеанса TCP, по которому передавалась команда SCSI.

В протоколе iSCSI определена концепция тега инициатора. Все ответы будут иметь соответствующий тег инициатора, который высылается вместе с первоначальной командой. Инициатор должен обеспечить уникальность тегов и исключить возможность повторного использования тега, пока инициатор не получит все ответы на соответствующую команду. Тег должен быть уникальным в рамках инициатора.

Протоколом iSCSI определена концепция нумерации команд, которая обеспечивает последовательную доставку команд через несколько сеансов TCP.

Кроме того, в протоколе iSCSI определен механизм **CRC** типа «точка-точка». Проверка **CRC** на втором уровне (например, на уровне GigabitEthernet) или на третьем уровне (контрольные суммы TCP/IP) может оказаться ненадежной, особенно если на пути передачи пакетов размещены другие устройства IP, например трансляторы сетевых адресов, маршрутизаторы и т.д. Поставщики подсистем хранения всегда с большой осторожностью относились к методам проверки целостности данных.

Протокол iSCSI имеет свои недостатки. При его использовании возникают проблемы, связанные с безопасностью, управлением сетевыми «заторами» и качеством обслуживания.

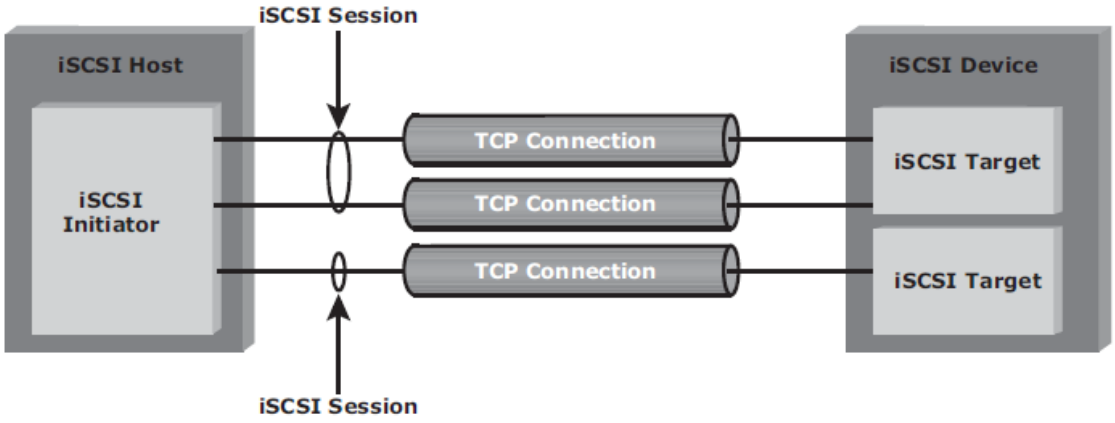

Рис. 19iSCSI сессия

Сессия устанавливается между инициатором и таргетом. Сессия идентифицируется sessionID (SSID), который включает в себя часть initiatorIDи targetID.

Сеанс может быть предназначен для:

- Обнаружения доступных таргетов инициаторами и местоположением конкретного таргета в сети;
- нормальная работа ISCSI (передача данных между инициаторами и таргетами).

Для этого необходимо одно или более TCP соединение внутри каждой сессии. Каждое такое соединение имеет уникальный идентификатор соединения (CID).

iSCSIсессия устанавливается через iSCSI login-процесс, который начинается, когда инициатор устанавливает TCP соединение с требуемым таргетом по порту 3260 или специальному таргет-порту. Во время login-фазы, инициатор и таргет аутентифицируют друг друга и договариваются о разных параметрах.

После успешного прохождения этой фазы, сеанс iSCSI вступает в фазу полной функциональности для обычных операций SCSI. На этом этапе, инициатор может посылать команды SCSI и данные в различные LUNs.

Финальная фаза – фаза разрыва соединения (выход из системы). Инициатор отвечает за начало процедуры выхода из системы. Тем не менее, таргет может также вызвать завершение, отправив сообщение ISCSI, что указывает на возникновение внутреннего состояния ошибки. После того, как запрос на выход из системы отправляется инициатором и принят таргетом, никакие дальнейшие запросы и ответы не могут быть отправлены.

5.3.3. Топологии iSCSI

Две реализации:

- Native (топология не имеет компонентов FC; инициаторы могут быть непосредственно или через сеть IP подключены к хранилищам),
- Bridged (с использованием FC).

Native iSCSI Connectivity

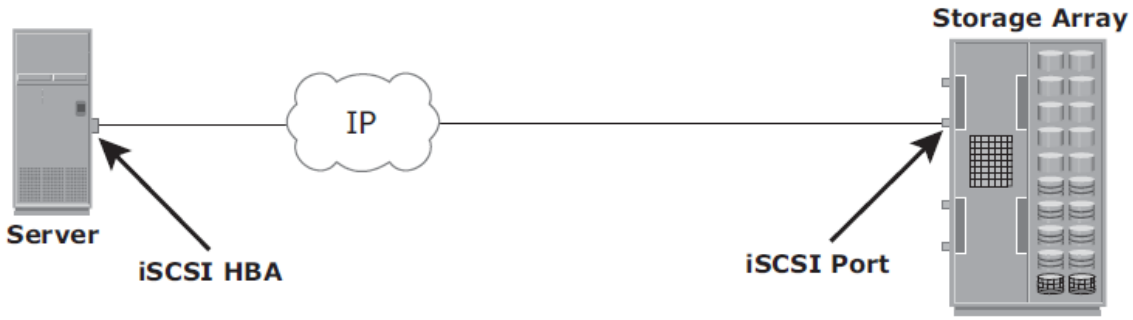

Рис.20 NativeiSCSIConnectivity

Компоненты FC не требуются для подключения к ISCSI, если ISCSI развернут с поддержкой массива.

На рисунке массив имеет один или более iSCSI портов с ip-адресами и подключенными к стандартному Ethernet коммутатору.

После подключения инициатора к сети, он может получить доступ к LUNам в хранилище. Один порт может обслуживать несколько инициаторов, пока он может справляться с генерируемым хостами трафиком.

Bridged iSCSI Connectivity

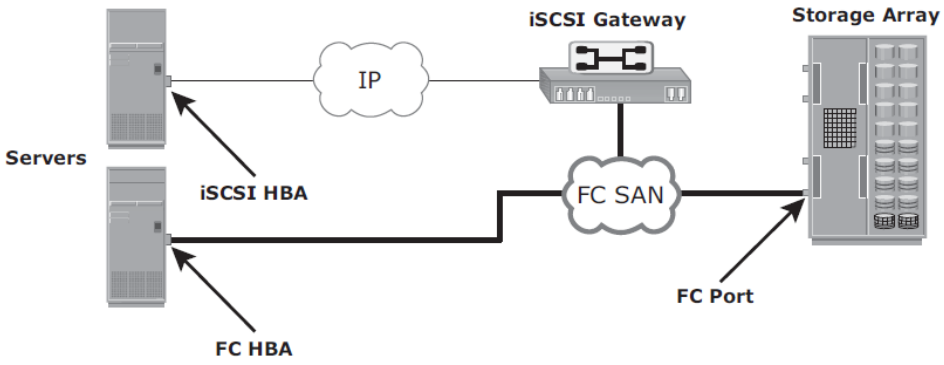

Рис. 21 Bridged iSCSI Connectivity

В этом случае массив не имеет iSCSI порты. Таким образом, внешнее устройство, называемое шлюзом или мультипротокольным маршрутизатором, должно быть использовано для облегчения обмена данными между хостом ISCSI и FCstorage. Шлюз преобразует IP-пакеты вFC кадры и наоборот. Мостовые устройства содержат как FC и Ethernet порты для облегчения связи между средами FC и IP.

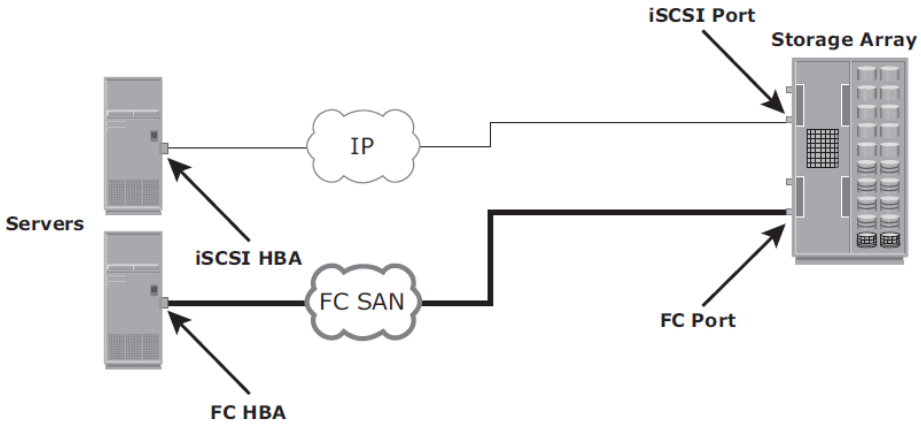

Рис. 22 Combining FC and Native iSCSI Connectivity

Наиболее распространенная топология представляет собой сочетание FC и native ISCSI. Как правило, массив хранения поставляется с двумя портами FC и ISCSI, которые позволяют ISCSI и FC соединение в той же среде.

5.4. NetworkFileSystem (NFS)

NFS является протоколом клиент-сервер для обмена файлами, который обычно используется в системах UNIX.NFS первоначально была основана на установленииUDP-соединения. Она использует машинно-независимую модель для представления данных пользователя. Также использует удален-

ный вызов процедур (RPC) в качестве способа межпроцессного взаимодействия между двумя компьютерами.

Протокол NFS предоставляет набор RPCs для доступа к удаленной файловой системе для следующих операций:

- Поиск файлов и каталогов,
- Открытие, чтение, запись и закрытие файла,
- Изменение атрибутов файлов,
- Изменение ссылки на файлы и каталоги.

NFS создает соединение между клиентом и удаленной системой для передачи данных. NFS (NFSv3 и ранее) является stateless протоколом - это означает, что она не поддерживает какой-либо таблицы для хранения информации об открытых файлах и связанных с ними указателей. Таким образом, каждый вызов предоставляет полный набор аргументов, чтобы получить доступ к файлам на сервере. Эти аргументы включают ссылку дескриптор файла в файл, определенную позицию для чтения или записи, а также версии NFS.

В настоящее время используются три версии NFS:

NFS version 2 (NFSv2): использует протокол UDP для обеспеченияs tateless-network-connectionмежду клиентом и сервером. Особенности, такие как блокировка, обрабатываются вне протокола.

NFS version 3 (NFSv3): наиболее часто используемая версия, которая использует UDP или TCP, на основе stateless протокола. Она включает в себя некоторые новые функции, такие как 64-битный размер файла, асинхронная запись, а также дополнительные атрибуты файлов для уменьшения предвыборки.

NFS version 4 (NFSv4): использует TCP и основана на stateful-протоколе. Он предлагает повышенную безопасность.

Обмен данными между клиентом и сервером NFS

- Клиенту (пользовательскому процессу) безразлично, получает ли он доступ к локальному файлу или к NFS файлу. Ядро занимается взаимодействием с железом через модули ядра или встроенные системные вызовы.
- Модуль ядра kernel/fs/nfs/nfs.ko, который выполняет функции NFS клиента отправляет RPC запросы NFS серверу через модуль TCP/IP. NFS обычно использует UDP, однако более новые реализации могут использовать TCP.
- NFS сервер получает запросы от клиента в виде UDP датаграмм на порт 2049. Несмотря на то, что NFS может работать с преобразователем портов, что позволяет серверу использовать динамически назначаемые порты, UDP порт 2049 жестко закреплен за NFS в большинстве реализаций.

- Когда NFS сервер получает запрос от клиента, он передаётся локальной подпрограмме доступа к файлу, которая обеспечивает доступ к локальному диску на сервере.
- Результат обращения диску возвращается клиенту.

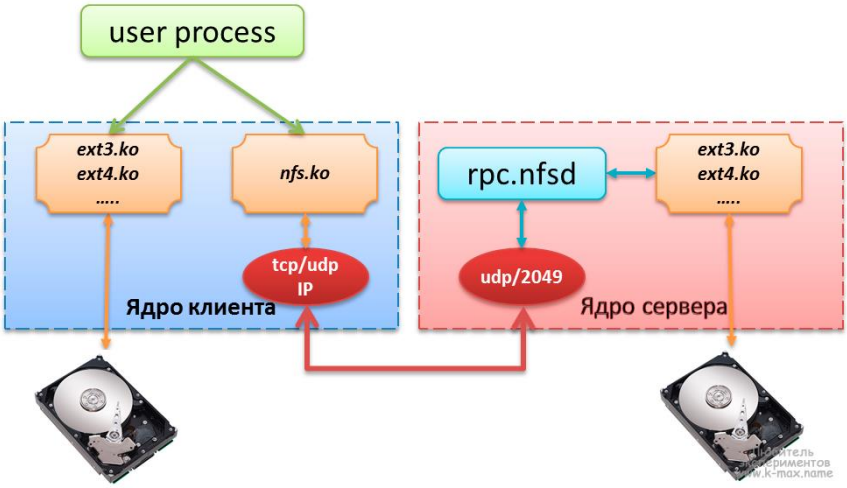

Рис. 23 Обмен данными между клиентом и сервером NFS

6. Механизмы безопасности

6.1. Защита гипервизора

Гипервизор ESXi защищен сразу после развертывания, он готов “из коробки”. Но мы можем и усилить его защиту, в том числе и при помощи режима lockdown и прочего встроенного функционала. Так же, мы можем настроить мастер-хост, на который будут ссылаться другие хосты, сравнивая свои конфигурационные файлы.

Ниже изложены способы повышения надежность и защиту гипервизора, рекомендованные разработчиками:

1. Ограничение доступа к ESXi

По умолчанию оболочка ESXi и службы SSH не запущены. Поэтому в DCUI (Direct Console User Interface) может войти только пользователь с root-правами. Если нам необходим доступ, например, по SSH, то настоятельно рекомендуется ограничить временные лимиты дабы исключить неавторизованный доступ. И только те пользователи, которые имеют необходимые привилегии, могут получить доступ к ESXi хосту для работы непосредственно с ним.

2. Зарегистрированные пользователи и малое количество привилегий

По умолчанию, root-пользователь ответственен за широкий спектр задач. Вместо того, чтобы предоставлять каждому администратору root-доступ на ESXi хосте, намного разумнее создать большее количество аккаунтов, имеющее различные привилегии по отношению к конфигурации хоста при помощи настройки разграничения управления доступом в vCenter Server. Можно создать собственную новую роль, назначить ей определенные привилегии и передать эту роль определенным пользователям (или группе пользователей).

3. Сведение к минимуму количество открытых портов на фаерволе ESXi.

Хост автоматически открывает порт, когда поднимается соответствующий сервис. Необходимо использовать CLI-интерфейс для проверки состояния портов, ибо некоторые из них нам уже не нужны, а любой открытый порт -- это уже дыра в безопасности).

4. Автоматизация управления хостами ESXi

Автоматизация -- основа управления. Все, что может быть настроено вручную, так же может быть настроено и скриптами. Особенную актуальность это приобретает при настройке не одной виртуальной машины или хоста, а нескольких. Альтернативой скриптам являются профили: можно обозначить некоторый хост мастером, а другие -- ссылающимися на него и

применить политики мастера на ссылающихся. Можно сделать это напрямую, а можно через утилиту от VMware Auto Deploy.

5.Воспользоваться преимуществом режима lockdown

В режиме lockdown доступ к хостам ESXi может быть получен только через vCenter Server. Начиная с версии vSphere 6.0 можно использовать обычный или строгий режим lockdown, а так же можно назначить пользователей, которым, несмотря на данный общий запрет, можно будет подключаться напрямую. Обычный режим подразумевает, что войти в консоль ESXi все же есть, так как служба DCUI не останавливается, но только при условии, что ни через vCenter, ни через веб-сервис войти невозможно вследствие каких-то отказов, а так же пользователь является либо администратором хоста и/или пользователем, прописанным в параметре DCUI Access хоста. В строгом режиме же данная служба останавливается, что означает, коли SSH и ESXi Shell не включены или exception-пользователи не имеются, то хост полностью недоступен и придется полностью переустанавливать хост.

6. Проверка интеграции VIB пакетов

У каждого VIB пакета есть свой уровень доверия. Текущее ранжирование уровней таково: VMwareCertified (Отбор в эту группу самый строгий, пакеты должны пройти такую же серию проверок, как и продукты, которые VMware делают для своих продуктов. На текущий момент только драйвера IOVP публикуются под таким уровнем. Главное преимущество таких пакетов в том, что VMware самостоятельно обрабатывает заявки в техподдержку по кейзам, связанным с ними), VMwareAccepted (Тестируется компанией-партнером, но VMware дает одобрение результаты тестов. При заявке в тех поддержку, они перенаправляют их в соответствующий отдел компании-партнера. Таких пакетов много больше, но они все еще не настолько многочисленны как CommunitySupported), PartnerSupported (Практически идентичны VMwareAccepted за исключением того, что: а) VMware не проверяет результаты тестирования; б) Под таким уровнем публикуются новые или необычные технологии компаний-партнеров) и CommunitySupported (Самый низкий уровень доверия, делается либо отдельным независимым программистом (или группой) или компанией, не являющейся партнером. Разумеется, VMware не обеспечивает поддержки, связанной с такими пакетами). Относительно пакетов используется простая иерархическая логика: мы можем установить VIB на хосту ESXi только если уровень доверия пакета будет выше или как минимум такой же как и уровень хоста. Чтоб установить менее доверенный пакет, необходимо снизить уровень самого хоста.

7. Управления сертификатами ESXi

В vSphere версии 6.0 и всех последующих , VMCA (VMware Certificate Authority) обеспечивает каждый хост ESXi подписанным сертификатом при помощи корневого центра сертификации, согласно стандартным настройкам. В случае если политики внутри вашей компании требуют замены, то вы спокойно можете использовать ваши собственные или независимые центры сертификации.

8. Аутентификация смарт-картой

Начиная с версии 6.0, вы можете использовать для аутентификации смарт-карты вместо имени пользователя и пароля. Однако, вам все равно потребуется вводить ПИН-код для успешного получения доступа.

9. Блокировка аккаунтов ESXi

Начиная с vSphere версии 6.0 поддерживается блокировка аккаунтов, подключающихся по SSH или vSphere Web Services SDK. Согласно стандартным настройкам, после десяти неудачных попыток войти в аккаунт, последний будет забанен. Разблокировка происходит через 2 минуты. Стоит отметить, что DCUI и ESXi Shell не поддерживает данную функцию.

6.2. Защита виртуальных машин

Виртуальные машины – это среды, в которых запускаются гостевые операционные системы и приложения. Все виртуальные машины VMware изолированы друг от друга. Эта изоляция позволяет нескольким виртуальным машинам безопасно запускаться, и в тоже время разделяет аппаратное обеспечение и гарантирует их непрерывную работу.

Даже пользователь с полномочиями системного администратора в гостевой операционной системе виртуальной машины не может нарушить этот уровень изоляции, чтобы получить доступ к другой виртуальной машине без полномочий, явно предоставленных системным администратором ESXi.

В виду изоляции виртуальной машины, если гостевая ОС, запущенная на ВМ выходит из строя, то другая ВМ продолжит работать на том же хосте.

Выход из строя гостевой ОС не влияет на:

1. Способность пользователей подключаться к другим виртуальным машинам
2. Способность других виртуальных машин получать необходимые ресурсы
3. Производительность и работу других виртуальных машин.

Поскольку VMkernel является связующим звеном физических ресурсов, и весь физический аппаратный доступ происходит через VMkernel, виртуальные машины не могут обойти этот уровень изоляции. Так же, как физическая машина связывается с другими машинами в сети через сетевую

плату, виртуальная машина связывается с другими виртуальными машинами, работающими в том же узле через виртуальный свитч. Далее, виртуальная машина связывается с физической сетью, включая виртуальные машины на других узлах ESXi, через физический сетевой адаптер.

Эти характеристики применяются к изоляции виртуальной машины в сетевом контексте:

- Если виртуальная машина не делит виртуальный свитч ни с какой другой виртуальной машиной, то она будет полностью изолирована от виртуальных сетей в узле;
- Если никакой физический сетевой адаптер не сконфигурирован для виртуальной машины, то виртуальная машина является полностью изолированный от любых физических сетей;
- Если Вы используете те же средства защиты (брандмауэры, антивирусное программное обеспечение, и т.д.), чтобы защитить виртуальную машину от сети так же, как и для физической машины, то виртуальная машина так же безопасна как физическая машина.

Вы можете защитить ВМ, настроив резервирование ресурсов и верхние пределы настройки хоста. Например, посредством подробного управления ресурсами, доступного в ESXi вы можете настроить ВМ так, что она будет получаться как минимум 10% ресурсов процессора машины-хоста, но никогда более чем 20%.

Резервирование ресурсов и их ограничение защищают ВМ от ухудшения производительности, которое могло бы возникнуть, если бы другая ВМ потребила бы слишком много общего ресурса. Например, если одна из ВМ на хосте выведена из строя DoS-атакой, то ограничение ресурсов этой машине позволит предотвратить атаку (ограничения ресурсов до такой степени, чтобы это не могло повлиять на другие ВМ)

Таким же образом, ограничение ресурсов каждой машине гарантирует, что в случае высокой потребности в ресурсах машиной, которая находится под DoS-атакой, все остальные виртуальные машины будут продолжать иметь достаточно ресурсов для их нормального функционирования и работы.

6.3. Безопасность виртуальной сети

Виртуальные коммутаторы, в отличии от физических, не выполняют динамическое соединение, необходимое для проведения межстанционных нападений. Они так же опускают двойные оформленные пакеты, что обуславливает неэффективность такого рода атак. vSwitch так де не позволяют пакетам покидать свой широковещательный домен, тем самым сводя на нет брутфорс-атаки, которые опираются на перегрузку коммутаторов, чтобы позволить передать пакеты на другие VLANдомены.

Но это совершенно не означает, что виртуальные машины в полной безопасности и им ничего не угрожает.

Именно поэтому, для максимальной защищенности виртуальных сетей, необходимо использовать комплексное обеспечение безопасности. Сюда входит использование межсетевых экранов, сегментация, защита физического коммутатора и безопасность виртуальных свичей (стандартного и распределенного).

6.3.1. Общие рекомендации:

✓ Отсоединить все неиспользуемые платы NIC, чтобы не было простого способа попасть в сеть.

✓ Убедиться в том, что хост-платформа, которая подсоединяет гипервизор и гостей к физической сети, безопасна путем установки разрешений для файлов, управления пользователями и группами, и настроив логгирование и синхронизацию времени.

✓ Шифрование всего трафика между клиентами и хостами, между системами управления и гипервизором, и между гипервизором и хостами с помощью протокола SSL.

✓ Обеспечение безопасности IP-коммуникации между двумя хостами с помощью шифрования и проверки подлинности каждого IP-пакета.

✓ Исключить использование самоподписанных сертификатов или сертификатов по-умолчанию — они уязвимы для атак типа «злоумышленник в середине».

✓ Помещение виртуальных свичей в смешанный режим для целей мониторинга и включение фильтрации MAC-адресов для предотвращения MAC-спуфинга.

6.3.2. Межсетевые экраны

Администраторы безопасности используют брандмауэры для защиты сети или отдельных компонентов в сети от вторжений.

Межсетевые экраны управляют доступом к устройствам в пределах их периметра, путем закрытия портов и явного или неявного определения исключений.

Основные способы фильтрации и анализа трафика МСЭ:

1. StatelessPacketFiltering

StatelessPacketFiltering является одним из старейших, но наиболее используемым способом фильтраций. Данный способ фильтрации просматривает сетевой трафик, ограничивает или блокирует доступ на основе ACL, анализирует заголовки протоколов транспортного и сетевого уровня. Stateless фильтрация обычно реализуется за счет оборудования 3 уровня, например, маршрутизатора.

Анализ пакетов происходит по следующим параметрам:

- IP-адрес источника и получателя,

- Порт источника/назначения (80, 443, 23…),
- Флаги (для TCP),
- Тип протокола,
- Биты типа обслуживания (ToS).

МСЭ с таким видом фильтрации не следят за состоянием подключений и могут пропустить в защищаемую сеть TCP-пакет от узла, с которым не открыто активных сессий. Так же к недостаткам данного способа фильтрации можно отнести слабую аутентификацию и отсутствие анализа данных IP-пакета.

2. Statefulpacketfiltering

Способ фильтрации Stateful, работает от сетевого уровня до уровня приложения, отслеживая все активные сессии, проходящие через брандмауэр (например, CiscoAdaptiveSecurityAppliance). Данный метод фильтрации в отличие от StatelessPacketFiltering сохраняет в памяти устройства информацию о прошлых сессиях. Опираясь на сохраненные данные, МСЭ анализирует полученный пакет на соответствие с предыдущим, после чего выносит соответствующее решение(блокировать/разрешить). Например, если начальное соединение было разрешено, то любые дополнительные соединения транспортного уровня, также должны быть разрешены. Для отслеживания сессий, во время передачи трафика заполняется таблица состояния соединений. Таблица состояния соединений динамически изменяется в режиме реального времени. Что является отличным решением для мониторинга сессий.

Главное отличие этих двух видов фильтрации состоит в том, что динамическая фильтрация проверяет всю сессию целиком, а статическая проверяет каждый пакет, не сопоставляя с предыдущим.

Важно помнить, что брандмауэр ESXi в ESXi 5.5 и более поздних версиях не позволяет вести фильтрацию трафика VMotion. Поэтому необходимо установить правила на внешнем брандмауэре, чтобы гарантировать, что никакие входящие соединения не могут быть сделаны к гнезду VMotion.

Виды межсетевых экранов в среде виртуальных машин:

- Брандмауэры между физическими машинами, такими как vCenterServer и ESXihost.
- Брандмауэры между одной виртуальной машиной, и другой, например, между виртуальной машиной, действующей в качестве внешнего веб-сервера и виртуальной машиной, подключенной к внутренней сети компании.
- Брандмауэры между физической машиной и виртуальной машиной, например, когда вы устанавливаете брандмауэр между физической картой сетевого адаптера и виртуальной машиной.

6.3.3. Сегментация

Сегментация - это хранение различных зон виртуальной машины в пределах хоста на различных сегментах сети. Если изолировать каждую зону виртуальной машины на своем собственном сегменте сети, сведется к минимуму риск утечки данных из одной виртуальной машины зоны к следующему. Сегментация предотвращает различные угрозы, в том числе AddressResolutionProtocol (ARP) подмены, в которой злоумышленник манипулирует ARP-таблицей, переназначая MAC и IP-адреса, что позволяет получить доступ к сетевому трафику к и от хоста. Злоумышленники используют ARP-спуфинг для создания человека в середине (MITM-атаки), выполнения отказа в обслуживании (DoS-атаки) или срыве работы виртуальной сети.

Планирование сегментации снижает вероятность передач пакетов между зонами виртуальной машины, что предотвращает сниффинг-атак, которые требуют отправки сетевого трафика жертве. Кроме того, злоумышленник не сможет использовать ненадежную службу в одной виртуальной зоне машины для доступа к другим зонам виртуальных машин на хосте. Можно реализовать сегментацию с помощью одного из двух подходов. Каждый подход имеет свои преимущества.

1. Используются отдельные физические сетевые адаптеры для зон виртуальных машин, чтобы гарантировать, что зоны изолированы. Ведение отдельных физических сетевых адаптеров для зон виртуальных машин, вероятно, самый безопасный способ и наименее склонный к возникновению неправильной конфигурации после создания начального сегмента.
2. Настраиваются виртуальные локальные сети (VLAN), для защиты сети. VLAN-сети являются стандартной схемой сетей IEEE со специальными методами тегирования трафика, которые позволяют маршрутизацию пакетов только для тех портов, которые являются частью одного и того же VLAN. При правильной настройке сетей VLAN обеспечивают надежный способ защиты виртуальных машин от случайных или злонамеренных вторжений.

6.3.4. Безопасность физических коммутаторов

Защита физического коммутатора для каждого ESXi-хоста предотвратит получение злоумышленником доступа к хосту и его виртуальным машинам. Для этого необходимо убедиться, что на портах физического свича отключен протокол STP(SpanningTreeProtocol) и опция несогласования настроена для каналов между внешним физическим коммутатором и виртуальным коммутатором в режиме VirtualSwitchTagging (VST). А также убедиться, что PortFastсконфигурирован для всех физических портов коммутатора, к которым подключены ESXi хосты. Для виртуальных машин, которые строят мост или выполняют маршрутизацию, требуется периоди-

чески проверять, что на первом физическом порту коммутатора настроен BPDUGuard, PortFast отключен и spanningtreeprotocol включен. В Vsphere 5.1 и более поздних, чтобы предотвратить (DoS) атаки, вы можете включить гостевой BPDU фильтр на ESXi хосте. Так же, на физическом коммутаторе требуется отключить DynamicTrunkingProtocol (DTP).

1. Безопасность стандартных виртуальных коммутаторов

Как и физические сетевые адаптеры, сетевой адаптер виртуальной машины может передавать кадры, которые позволяют, подключиться с другой машины или выдавать себя за другую машину, чтобы получить сетевые кадры, предназначенные для этой машины. Кроме того, как и физические сетевые адаптеры, виртуальные коммутаторы сети могут быть сконфигурированы таким образом, чтобы принимать кадры, предназначенные для других машин. Оба эти сценария представляют угрозу безопасности.

При создании стандартного коммутатора, необходимо добавить группы портов, ввести политику трафика системы для виртуальных машин и VMkernel-адаптеров, подключенных к коммутатору.

В рамках добавления группы портов VMkernel или виртуальной машины к стандартному коммутатору, ESXi настраивает политику безопасности для портов в группе. Вы можете использовать эту политику безопасности, чтобы гарантировать, что хост запрещает гостевой операционной системе виртуальной машины обнаруживать другие машины в сети.

СтандартныйvSwitch можно защитить от атак уровня L2, ограничивая некоторые из режимов MAC-адрес, используя параметры безопасности коммутаторов. Каждая виртуальная машина сетевой адаптер имеет начальный адрес MAC и эффективный MAC-адрес.

- Начальный адрес MAC - Исходный MAC-адрес, присваиваемый при создании адаптера. Хотя начальный MAC-адрес может быть изменен из-за пределов гостевой операционной системы, он не может быть изменен в гостевой операционной системе.
- Эффективный MAC-адрес - MAC-адрес, который фильтрует входящий сетевой трафик с MAC-адрес назначения, который отличается от эффективного MAC-адреса. Гостевая операционная система отвечает за установление эффективного MAC-адрес и обычно соответствует эффективному MAC-адрес исходного MAC-адреса.

После создания виртуального коммутатора, эффективный MAC-адрес и начальный адрес MAC одинаковы. Гостевая операционная система может изменить эффективный MAC-адрес на другое значение в любое время. Если операционная система изменяет эффективный MAC-адрес, его сетевой адаптер получает сетевой трафик, предназначенный для нового MAC-адреса.

При передаче пакетов через сетевой адаптер гостевой операционной системы, как правило, коммутатор размещает свой собственный эффективный MAC-адрес в исходном MAC поле адреса кадров Ethernet. Он помещает MAC-адрес для принимающего сетевого адаптера в целевом поле адреса MAC. Принимающий адаптер принимает только пакеты, если MAC-адрес назначения в пакете соответствует его собственному эффективному MAC-адресу.

Операционная система может передавать кадры с подменой исходного MAC-адреса. Это означает, что операционная система может проводить злонамеренные атаки на устройства в сети, выдавая свой сетевой адаптер, за авторизируемый в приемной сети.

Политика безопасности распределенных групп портов и портах включает в себя следующие варианты:

- Беспорядочный режим (Promiscuous mode operation)- позволяет перевести виртуальный коммутатор в режим зеркалирования всех портов на все (режим работы концентратора). При установке в значение Reject (по умолчанию) хост отключает режим концентратора на коммутаторе. При установке в значение Accept хост включает режим концентратора на коммутаторе.
- изменение MAC-адреса (MAC addresschanges) - влияет на трафик, получаемый виртуальной машиной. При установке в значение Accept (по умолчанию) хост позволяет виртуальной машине получать пакеты, если ее эффективный MAC-адрес отличается от начального MAC-адреса. При установке в значение Reject хост не допускает пакеты до виртуального сетевого адаптера виртуальной машины, если ее эффективный MAC-адрес отличается от начального MAC-адреса. При этом гостевая операционная система не может распознать, что ее отключили из-за смены MAC-адреса.
- Поддельная передача (Forgedtransmissions) - влияет на трафик, передаваемый виртуальной машиной. При установке в значение Accept (по умолчанию) хост позволяет виртуальной машине передавать пакеты, если ее эффективный MAC-адрес отличается от MAC-адреса источника, указанного в заголовке пакета. При установке в значение Reject хост не позволяет передавать пакеты виртуальной машине, если ее эффективный MAC-адрес отличается от MAC-адреса источника, указанного в заголовке пакета. При этом гостевая операционная система не может распознать, что ее отключили из-за подмены MAC-адреса.

2. Безопасность распределенного vSwitch

- Для распределенных портовых групп со статической привязкой, следует убедиться, что функция Auto Expand отключена (включена по умолчанию в Vsphere 5.1 и более поздних версий).

- Необходимо убедится в том, что все частные идентификаторы VLAN любого распределенного коммутатора полностью документированы.
- Если используется VLAN-тегирование на dvPortgroup, идентификаторы VLAN должны соответствовать идентификаторам на внешних VLAN-коммутаторах в курсе вверх по течению. Если идентификаторы VLAN не отслеживаются полностью, ошибочное повторное использование идентификаторов может разрешить трафик между неадекватными физическими и виртуальными машинами. Точно так же, неправильные или отсутствующие идентификаторы VLAN могут привести к тому, что трафик не будет проходить между физическими и виртуальными машинами.
- Следует проверить, что неиспользуемые порты не существуют в виртуальных группах портов, связанных с VsphereDistributedSwitch.
- Метки всех dvSwitch. Распределенные коммутаторы, связанные с ESXi требуют заполнения поля имени коммутатора. Эта метка служит в качестве функционального дескриптора для коммутатора, так же, как имя хоста, связанного с физическим коммутатором. Метки у распределенных коммутаторов указывает на функцию или IP-подсеть коммутатора. Например, вы можете пометить коммутатор как внутренний, чтобы указать, что это только для внутренней сети между частным виртуальным коммутатором без каких-либо физических сетевых адаптеров, связанных с виртуальной машиной.
- Отключение проверки работоспособности сети для распределенных коммутаторов, если он не используется. Проверка работоспособности сети по умолчанию отключена. После ее включения пакеты содержат информацию о хосте, коммутаторе и портах, что злоумышленник может потенциально использовать. Эту проверку стоит использовать только для устранения неполадок, и отключать ее, когда поиск и устранение неисправностей завершены.

Защита виртуального трафика от подмены и перехвата 2-го уровня атак путем настройки политики безопасности на группы портов или портах производится так же, как и защита виртуального трафика при использовании стандартного коммутатора.

6.4. Безопасность в vSphere 6.5

В последней версии vSphere безопасности уделено огромное внимание, она буквально является центром данного релиза. Основное внимание в нем уделено безопасности управления. А ключом безопасности является масштабируемость и автоматизация.

6.4.1. Шифрование ВМ

Реализацию шифрования виртуальных машин разрабатывали уже достаточно давно, но только в этой версии ее использование стало возмож-

ным, при этом данное введение не несет в себе негативных последствий для работы.

Шифрование ВМ происходит на уровне гипервизора. Поскольку в качестве I/O-интерфейса выступает контроллер виртуального диска, шифрование происходит почти мгновенно с помощью модуля в ядре, еще до отправления к слою хранения ядра. Все файлы виртуальной машины оказываются зашифрованы.

Преимущества:

- ✓ Независимость от типа гостевой ОС и хранилища данных, т.к. шифрование происходит на уровне гипервизора.
- ✓ Шифрование осуществляется через политики, которые могут применяться к многим ВМ.
- ✓ Шифрование не управляется изнутри виртуальной машины, поэтому в ее памяти не хранятся ключи.
- ✓ Управление ключами осуществляется на основе отраслевого стандарта KMIPc 1.1. это дает возможность выбора и гибкость.
- ✓ Для шифрования используется алгоритм AES-Ni, благодаря чему ускоряется процесс шифрования и повышается уровень безопасности.
- ✓ Шифрование vMotion.Во время миграции используется сгенерированный сервером vCenter однократно и случайным образом 256-битный ключ и генерируется 64-разрядный «код».Ключ шифрования и код упакованы в спецификации миграции, отправляемой на оба хоста. В этот момент все данные виртуальных машин vMotion шифруются с ключом и кодом, гарантируя невозможность воспроизведения данных.

6.4.2. Secure Boot

Суть механизма безопасной загрузки в vSphere 6.5 заключается в том, что загрузка подписанного кода гипервизора ESXi контролируется со стороны UEFI firmware, а также не разрешается исполнение неподписанных пакетов. Это не позволяет постороннему ПО модифицировать ядро гипервизора.

UEFI (Unified Extensible Firmware Interface) - это замена традиционному BIOS в серверах и настольных ПК. Механизм Secure Boot - это один из "протоколов" UEFI, который предоставляет возможности контроля загрузки подписанного кода за счет хранения цифровых сертификатов, которые хранятся в микрокоде (firmware) компьютера (signature database).

Для обеспечения безопасной загрузки используются следующие компоненты:

- **Загрузчик ESXi (boot loader)** - он убеждается в том, что цифровая сигнатура кода не была изменена. Загрузчик подписан сертификатом Microsoft UEFI Public CA. Он также содержит VMware public key, с помо-

щью которого проверяются компоненты VM Kernel, Secure Boot Verifier, а также пакеты VIB.

- **Ядро VM Kernel** - это также подписанная часть кода. Первое, что делает VM Kernel, это запускает Secure Boot Verifier.
- **Secure Boot Verifier** - он также хранит VMware public key и проверяет аутентичность всех VIB-пакетов, которые загружаются во время загрузки ESXi.
- **VIB-пакеты (vSphere Installation Bundles)** - эти пакеты, помимо реализуемых ими сервисов, содержат файл XML descriptor и файл цифровой подписи (digital signature file). Во время загрузки ESXi в памяти создается "карта" содержимого каждого из VIB-пакетов, соответственно не требуется проверять каждый из его файлов, а достаточно проверить подпись пакета целиком (это быстрее работает).

Процесс загрузки хоста ESXi с точки зрения Secure Boot:

1. Включение питания.
2. UEFI Firmware валидирует загрузчик ESXi Boot Loader на предмет соответствия сертификату Microsoft внутри микрокода UEFI.
3. ESXi Boot Loader валидирует компонент VM Kernel на предмет соответствия сертификату в Boot Loader.
4. VM Kernel запускает компонент Secure Boot Verifier.
5. Secure Boot Verifier валидирует каждый VIB-пакет на соответствие сертификату VMware, который находится в хранилище Secure Boot Verifier.
6. Запускаются необходимые сервисы управления (DCUI, hostd и т.п.).

7. Распределение ресурсов. VMware Distributed Resource Scheduler (DRS)

На работающей на сервере виртуальной машине 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-кластера. Также, нам необходимо наблюдать, достаточно ли ресурсов выдается нашим виртуальным машинам. Если нет – определять, что является так называемым «узким местом».

Теперь стоит разобраться, что такое кластер. В системе VMware, кластер — это группа хостов (физических серверов), которые связаны между собой сетью, и управляемые единым сервисом и совместно выполняющие определенные функции, как один большой организм.

На платформе от VMware — vSphere можно построить 2 вида кластеров: HA (High-availability) кластер и DRS (Distributed Resource Scheduler) кластер, которые работают на уровне виртуальной машины.

VMware Distributed Resource Scheduler (DRS) — это встроенный балансировщик, служащий для группировки узлов ESXi в кластеры ресурсов и уравновешивания рабочих нагрузок между кластерами в виртуальной среде. С его помощью в группах становится возможно проводить развертывание новых ресурсов и выполнять автоматический перенос виртуальных машин во время технического обслуживания, не нарушая работу служб.

7.1. Настройки распределения ресурсов для ВМ.

Для начала, необходимо описать настройки, которые позволят обеспечить или снизить количество ресурсов, которое выделяется для одной виртуальной или группы машин в пуле ресурсов.

7.1.1. Настройки для процессоров

Для процессоров виртуальных машин мы можем задавать настройки три настройки:

Reservation – это количество мегагерц, гарантированно закрепленных за данной ВМ в момент ее включения. Но, необходимо обратить внимание на то, что резерв – это блокирующая настройка. Иными словами, если для обеспечения резерва виртуальной машины не хватает МГц у сервера, то ВМ просто не включится и выдаст соответствующее сообщение об ошибке.

Limit – это max количество МГц, выделенное на все процессы для данной виртуальной машины. По умолчанию, процессорные ресурсы ВМ ограниченны только физически. Одно виртуальное ядро не может получить больше мегагерц, чем предоставляет одно ядро физического процессора.

Но следует помнить, ресурсы процессора выдаются ВМ по необходимости. Это значит, что если нагрузка маленькая, то и требуемое процессорное время будет выделятся в небольшом количестве, не смотря на высокий резерв, который мы задали. Но при этом у нее будет возможность при необходимости задействовать и больше, забрав часть у других виртуальных машин. По определению, сумма резервов всех работающих ВМ не превышает физического количества ресурсов сервера.

Shares. Иногда бывают ситуации, когда ресурсов сервера хватает для покрытия резервов всех работающих ВМ, но недостаточно для того, чтобы соответствовать их потребностям и ВМ не достигали своих лимитов. В таком случае можно сделать настройкиShares. ВМ получит столько ресурсов, сколько долей составляет share одной машины относительно всех возможных share.Эта величина является безразмерной.

Рассмотрим пример: 4 однопроцессорные ВМ стоят на одном ядре сервер. Shares у каждой по 1000, а всего 4000. Следовательно, доля каждой ВМ – четверть. Это значит, что для одной из этих ВМ может быть выделена четверть ресурсов этого ядра.

Итак, механизм shares работает, когда ВМ уже превысила свой резерв, еще не достигла своего лимита, и не хватает ресурсов на все претендующие на них ВМ.

7.1.2. Настройки для памяти

Reservation – это количество мегабайт физической оперативной памяти, гарантированно закрепленных для текущей виртуальной машины при её включении. Вся остальная необходимая память выделяется из файла swap (подкачка).

Гостевая ОС видит количество памяти, которое называется верхней границей, её можно настроить во вкладке Hardware. Если необходимо ограничить ВМ сверху, следует менять не Limit, а количество памяти на вкладке Hardware. Именно ее называют ram в контексте лицензирования vSphere.

Выделенная физическая память не может превышать некоторого количества мегабайт, которое называется limit.Из файла подкачки будет обязательно выдано всё, что осталось до hardwarememory, даже если на сервере хватает свободной оперативной памяти.

В большинстве случаев, нет необходимости использовать настройку Limit. Если нужно выделить для ВМ меньше памяти, лучше менять значение hardwarememory.

Настройка Shares для памяти аналогична настройке Shares процессоров. Она определяет, какое количество ресурсов от суммы Shares получит данная ВМ.

Необходимо помнить, если сервер не имеет достаточное количество памяти для удовлетворения резерва ВМ, то данная машина не включится.

Но при этом если памяти хватает только для резерва, а для сверх резерва нет, то необходимую физическую оперативную память можно взять из файла подкачки. Используется два механизма подкачки – файл подкачки гостевой ОС и файл подкачки VMkernel, создаваемый для каждой ВМ при ее включении. При включении ВМ создается файл .vswp, именно в него ESXi адресует часть памяти ВМ, если физической не хватает. В наихудшем случае, эту часть составляет вся память сверх резерва до hardwarememory. Тогда размер файла подкачки -Hardwarememory – reservation. Следовательно, если reservation=0, то при включении каждой виртуальной машины будет создаваться файл, размер которой равняется её оперативной памяти. Отсюда следует два вывода:

- ВМ не включится, если недостаточно места для создания файла подкачки на хранилище;
- Один из способов освободить некоторое количество места на разделах VMFS– увеличитьreservation для памяти ВМ: при увеличении резерва, уменьшается место, резервируемое под файл .vswp, файл подкачки VMkernel.

7.2. Пулы ресурсов

Когда количество виртуальных машин исчисляется десятками, их число будет изменяться.

Создаются новые машины, какие-то машины удаляются, клонируются. Для каждой отдельной виртуальной машины отслеживать эти настройки неудобно. Поэтому для удобства создали пулы ресурсов, в них используются настройки для групп ВМ.

Настройки пула ресурсов схожи с настройками распределения ресурсов для ВМ -Limit, Reservation и Shares. Единственное отличие – наличие флажка ExpandableReservation. При его включении становится возможно «брать взаймы» свободныеreservation у родительского пула.

Например, существует пул ресурсов Main, в который входят два дочерних - Child1 &Child 2. В пулах Child1&Child2поместилинесколько ВМ, для которых требуется настроитьreservation. Из описанного ранее известно, что для включения машин с резервом необходимо иметь достаточно ресурсов. Значит и в пулах, которые содержат эти ВМ, должны существовать свои reservation в нужном количестве.

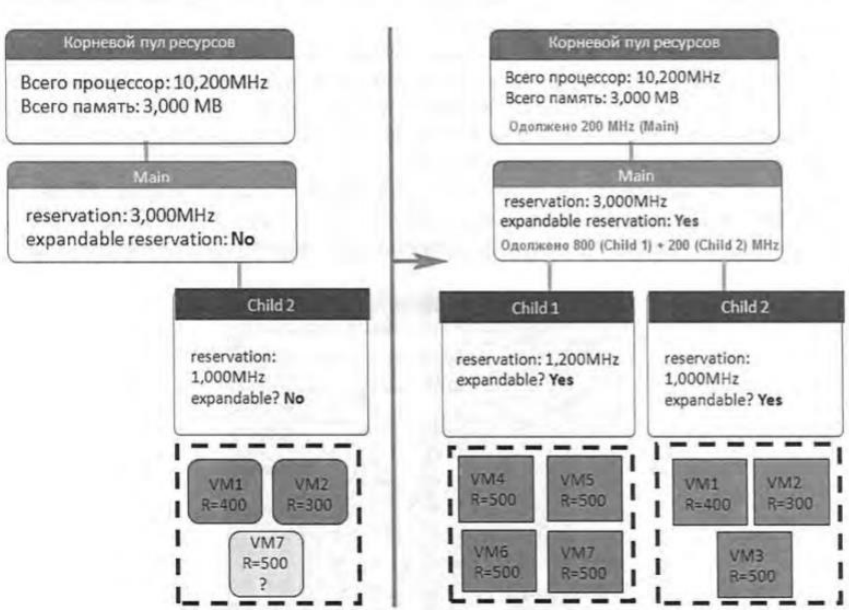

Рис 24. Модель ExpandableReservation.

В дочернем пуле Child 2 всего 1000 МГц, а виртуальные машины VM1 &VM2 занимают в сумме 700 МГц, значит не удастся включитьVM7 с резервом в 500 МГц. Но у родительского пула Main есть свободные reservation в достаточном количестве. Значит есть возможность включить VM7 за счет родительского пула.

На правой части рисунка включен ExpandableReservation для пулов Main и Child 2. Теперь дочерний сможет «одолжить» свободные МГц у родительского. На левой части рисунка 4. пул Child 1 тоже решил «попросить» ресурсы у родительского, но их оказалось недостаточно. В таком случае Maincам одалживает reservation у своего корневого пула.

Включенный ExpandableReservation хорош тем, что рассчитывать точное количество reservation для дочерних пулов становиться необязательно. В случае нехватки они смогут одолжить необходимое количество у родительских пулов.

Для сервера можно создавать пулы ресурсов вне кластера или для DRS-кластера. Если сервера в кластере без функции DRS, то ни для серверов, ни для кластера пулы создать будет нельзя.

7.3. Способы перераспределения ресурсов в ESXi

Выше были описаны настройки распределения ресурсов - limit, shares и reservation. Теперь можно говорить о том, благодаря каким механизмам ESXi умеет эти настройки использовать и устройство распределения ресурсов сервера между виртуальными машинами.

Процессоры бывают трех типов (С точки зрения ESXi):

- физические – процессоры, «сокеты»;

- логические (LCPU) – ядра физического процессора. ОднаLCPU = одна очередь команд;
- виртуальные (VCPU) – процессор виртуальной машины (max 32).

Пояснение: в пятой версии vSphere при настройке виртуальных процессоров для ВМ выбирается число «виртуальных процессоров» и «число ядер в каждом виртуальном процессоре». Произведение этих чисел – столько потоков сможет использовать данная виртуальная машина. То есть, если у виртуальной машины один «виртуальный сокет» и он четырехядерный, и если у виртуальной машины четыре «одноядерных виртуальных сокетов», то можно говорить о том, что у нее четыре vCPU, так как в обоих случаях с точки зрения производительности получается четыре идентичных потоков.

Также необходимо упомянуть о том, что один vCPU работает на одном LCPU. Это значит, что на одном ядре может работать виртуальная машина с одним виртуальным процессором. Следовательно, даже если имеется однопроцессорный четырёхядерный сервер, на котором работает одна ВМ с одним процессором, то она задействуют только одно ядро. Но, если есть ВМ с четырьмяvCPU, то она задействует все четыре ядер – ESXi в обязательном порядке развозит процессоры одной машины по разным ядрам. Также на одном ядре может работать несколько разных ВМ с разнымиvCPU.

Теперь рассмотрим настройки HyperthreadedCoreSharing и CPUAffinity, которые есть у ВМ.

HyperthreadedCoreSharing управляет использованием логических процессоров. Когда включается гипертрейдинг, то на логические процессоры делится каждое физическое ядро. Если у нас включен hyperthreaded, то каждое физическое ядро порождает два логических, что делает возможным в теории выполнить некоторые команды в два потока.

Три варианты настроек:

- Any – на каждом из двух логических ядер одновременно может работать несколько виртуальных процессоров одной или разных ВМ;
- Internal – используется когда у ВМ несколькоvCPU. В таком случае они могут работать на разных логических ядрах одного физического ядра. Данная настройка доступна только для многопроцессорных ВМ. Если виртуальная машина имеет только один процессор, то значение = None;
- None – однопоточный процесс. Т.е. на данном ядре выполняется только один vCPU. Второе логическое ядро простаивает.

Перейдем к следующей настройке – SchedulingAffinity.

В SchedulingAffinity можно указывать конкретные ядра, на которых будут выполняться vCPU этой ВМ. В отличии от гипервизора, который по

умолчанию может расположить процессор ВМ на любом ядре, данная настройка дает возможность ограничить выбор.

ВМ потеряет способность к живой миграции, если сменить значение, которое стоит по умолчанию. Это значительно ограничивает применимость данной SchedulingAffinity.

7.4. Миграция виртуальной машины

7.4.1. Миграция выключенной ВМ

Если машина находится в состоянии паузы (suspend) или выключена, то ее можно мигрировать и между хранилищами, и между серверами, а также между ними обоими одновременно.

Запустить такую миграцию можно перетаскиванием ВМ на нужный сервер или пунктом Migrate в контекстном меню (с участием vCenter). Если необходимо сделать без участия vCenter, то все немного сложнее:

1. Выключение/перевод в режим паузы
2. Удаление ВМ из иерархии объектов ESXi (RemovefromInventory)
3. Если необходим, перенос файлов ВМ на другое хранилище
4. Регистрация ВМ на нужном сервере.

Так же можно перенести ВМ с сервера на сервер с минимальным простоем, без участия vMotion/StoragevMotion(может быть недоступен):

1. Создание snapshot (снимок состояния) для работающей ВМ
2. Копирование всех её файлов, кроме последнего snapshot, на новое хранилище
3. Перевод ВМ в режим паузы
4. Перенос оставшихся файлов ВМ на другое хранилище
5. Поиск скопированных файлов на другом сервере и выбрать Add to inventory (если сервер один и тот же, то необходимо удалить исходную ВМ).
6. Включение ВМ, удаление снимка состояния и исходной ВМ

7.4.2. Живая миграция между хранилищами

StoragevMotion – живая миграция файлов ВМ между хранилищами. Проще говоря – перенос файлов с хранилища на хранилище без выключения ВМ. Осуществляется поддержка любых типов хранилищ, в том числе и локальных дисков, перенос как всей ВМ, так и частично.

Так же в функционал SvMotion входит:

- Конвертация диска виртуальной машины между форматами thin&thick,
- Копирование RDM в файл vmdk (vmdk в RDM невозможно скопировать).

В чем же заключается суть процесса переноса без выключения?

Во-первых, при инициации StoragevMotion, начинается копирование файлов ВМ на новое место. Так как файлы дисков большие, то с ними могут возникнуть проблемы. Что бы избежать изменения и других проблем применяется следующий подход – все изменения дублируются, т.е. записываются и в исходный и в новый файл-диск.

Во-вторых, файлы типа vmdk, являющиеся основным объемом информации, копируются. Для этого требуется лишь одна итерация, т.к. все данные, которые изменились во время переноса, сразу же были записаны в новую копию.

И в-третьих, когда все файлы скопировались, гипервизор отправляет запросы уже к новым файлам, старые же удаляются за ненадобностью.

Инструкция для запуска процесса и дальнейшие шаги в мастере настроек:

1. Выбрать Migrate в контекстном меню ВМ и Change Datastore или Home/Datastore/хранилище с ВМ/Virtual Machines и перенесите нужную ВМ на другое хранилище
2. Select Datastore – выбор необходимого вам хранилища (Advanced – указание миграции только отдельных дисков ВМ)
3. Disk Format – указание типа диска для ВМ на новом хранилище (например Thick, Thin)

ВМ можно перенести между любыми типами хранилищ, т.к. условий не указано. И для виртуальных машин условий вообще нет. А для сервера только одно – лицензия на StoragevMotion.

7.4.3. Живая миграция между серверами

Процесс живой миграции (vMotion) –переезд включённой (живой) ВМ с сервера на сервер без прекращения работы.

Такая миграция необходима:

- Когда необходим плановый простой сервера,
- Для балансировки нагрузки.

Для начала при запуске миграции, vCenter проверяет осуществление требований для серверов и виртуальной машины, между которыми происходит миграция.

Потом гипервизор копирует все файлы из оперативной памяти на другой ESXi. Так же он ведет список адресов измененных блоков памяти перемещаемой ВМ. Это необходимо, так как наша машина продолжает работать и содержимое, которое мы копируем, меняется в течение времени.

Кроме того, надо помнить, что именно под vMotion, через интерфейс VMkernel передается основной объем памяти. И уже после передачи всей памяти на другой ESXi- ВМ блокируется полностью, и начинают переда-

ваться последовательно страницы памяти, которые были изменены. Их размер невелик и значит время, за которое ВМ заблокируется полностью так же небольшое. Но бывает такое, что этот объем превышает некое пороговое значение (установленное ранее), и тогда ESXI просто повторит итерацию. Именно благодаря этому, даже если пройдет несколько итерации, область памяти, для передачи которой ВМ полностью блокируется, станет весьма небольшой.

Во время такого процесса мы получаем две идентичные ВМ, которые находятся на выбранных серверах. И, значит, для нормального продолжения работы необходимо уничтожить изначальную ВМ и оповестить всех по сети, что машина с этим MAC-адресом теперь доступна уже на другом порту физического коммутатора. Самое интересное, что полный перенос происходит в течении одной секунды.

Вернемся к условиям, которые проверяются центром, перед запуском миграции. Они необходимы для простого и безболезненного переноса(избежать простоя например).

Первое и основное условие, которое налагается на сервер, - процессоры серверов должны быть совместимы с точки зрения vMotion. Это важно, т.к. процессор – единственная подсистема сервера, которую гостевые ОС видят такой, какая она есть физически. Под совместимостью подразумевают набор поддерживаемых инструкции (SSE 3,SSE 4.1, NX/XD и др.), а не одинаковые тактовые частоты, размер кэш-памяти, количество ядер. Простой пример, если существует два разных процессора на разных а приложения используют какую-то из инструкции, что была доступна до, но недоступна сейчас, то приложение упадет.

Второе условие, которое налагается на ВМ и сервера – все задействованные ресурсы должны быть доступны на обоих серверах. Перечислим некоторые из них:

- Файлы ВМ (vmx, vmdk). Т.е. виртуальная машина должна располагаться на общем хранилище;
- Настройка SCSIBusSharing для виртуальных SCSI-контроллеров должна быть обязательна выставлена как None (виртуальные машины-кластеры Майкрософт не могут быть мигрированы);
- Т.к. к ВМ могут быть подключены образы CD-ROM, то и их файлы тоже должны быть общедоступными для обоих серверов;
- Группы портов, к которым подключена ВМ, так же должны существовать на обоих серверах;
- CD-ROM сервера не должны быть подключены к ВМ;
- Не должно быть указано конкретных ядер (не настроено CPUAffinity);

- ВМ не должна быть подключена к виртуальному коммутатору без привязанных сетевых контроллеров (эту проверку можно отключить);
- Необходимо выделить отдельный гигабайтный интерфейс (это и отдельный интерфейс для vMotion) для передачи оперативной памяти ВМ между серверами.

Перейдем к процессу запуска. Это можно сделать, выбрав Migrate в контекстном меню и затем нажать Changehost, либо (что проще) просто перетащить ВМ на нужный ESXi (так можно избежать множества вопросов). Далее останется лишь:

- SelectResourcePool – выбор в какой пул ресурсов перемещать ВМ, если это необходимо;
- vMotionPriority – нужно ли резервировать ресурсы по перемещаемую машину на сервере назначения или же нет. Если вы выберете второй вариант, то сможете произвести процесс миграции и в условиях недостатка ресурсов, но тогда миграция займет времени гораздо больше.

В новой версии vSphere реализована возможность миграции ВМ сразу по нескольким сетевым контроллерам, что позволило повысить скорость миграции ВМ с большими объемами памяти при использовании гигабитных сетевых соединений.

7.5. DRS-кластер

Живая миграция ВМ - это полезная вещь, позволяющая мигрировать ВМ между серверами для балансировки нагрузки и освобождения сервера, когда нужно его перезагрузить. Данный механизм может пригодиться при установке обновлений или аппаратном обслуживании.

Но если инфраструктура достаточно большая, то администратору становиться сложно выполнять эти операции. В таком случае ему в помощь приходитVMwareDistributedResourceScheduler (DRS).

Для чего необходимDRS:

- балансировка нагрузки между серверами (по процессору и памяти);
- автоматическийvMotionВМ с сервера в режиме обслуживания (maintenancemode).

Этим режимом помечается сервер, который необходимо освободить от ВМ перед операциями обслуживания. Выполняет либо администратор, либоVMwareUpdateManager.

DRS для решения этих задач может:

- 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 того, чтобы DRS-кластер мог успешно функционировать, необходимо настроить его, и добавить в него сервера ESXi. Можно включить DRS для уже существующего кластера или вместе с созданием нового.

Для DRS-кластера доступны следующие группы настроек, благодаря которым мы и разберемся в принципах его работы.

В VMwareDRS существует базовая настройка, называющаяся«уровень автоматизации». Данные настройки указывают, что предлагать администратору на согласование, а что выполнять автоматически.

Рассмотри три варианта настройки:

- ручной или Manual – включение или игнорирование ВМ, DRS и vMotion будет зависеть от администратора. Непривилегированный пользователь, при такой работе DRS, не сможет увидеть окно выбора сервера и ВМ не включится;
- полуавтомат или Partially automated – включение ВМ, DRS происходи автоматически, а vMotion только по согласованию;
- полностью автоматический или Fully automated – ВМ, DRS и vMotion включаются автоматически.

Следующей настройкой является Migration Threshold - она указывает степень консервативности или агрессивности работы кластера DRS. Если настроено ближе к Aggressive, тогда будут выдаваться все рекомендации включая маловажные. А если положение ближе к Conservative, тогда будет происходить автоматический переброс ВМ для сбалансирования.

DRS может давать рекомендации для совершения миграции по одной из следующих причин:

- для равномерного распределения нагрузки на процессоры сервера, или резерва по процессору;
- для равномерного распределения нагрузки на память сервера, или резерва по памяти;
- для того, чтобы угодить настройке reservation для пулов ресурсов;
- для исполнения правил настройки кластера DRS - affinity или anti-affinity ;
- для миграции ВМ с сервера, если он переходит в режим standby или maintenance;
- для исполнения рекомендации DPM (Distributed Power Management), при условии использования этого компонента.

Может быть 5 приоритетов миграции (1 - самый высокий, 5 - наименьший). Наивысший приоритет дается для процесса миграции ВМ с более загруженного сервера на менее.

Главные параметры:

- MigrationThreshold (указывает на степень несбалансированности кластера);
- Target host load standard deviation (стандартное отклонение нагрузки);
- Current host load standard deviation (стандартное отклонение текущей нагрузки).

При расчете host load standard deviation в формуле используется расчет нагрузки на каждый сервер: сумма/ресурсы.

Анализируя загрузку CPU и памяти хостов, DRS получает информацию о standard deviation. Система является «несбалансированной», если Current host load standard deviation превышает заданную величину. Тогда для исправления данного состояния DRS производит перераспределение ВМ внутри кластера без задержек. В дальнейшем происходит мониторинг загрузки кластера и исправления возникшего дисбаланса.

Правилаaffinity и anti-affinity дают миграциям высокий приоритет, о которых будет упомянуто далее. Максимальный приоритет может получить рекомендация, если сервер находится в режимеMaintenance. В первую очередь будет происходить миграция именно с такого типа хоста.

Таким образом, если выбрано самое консервативное положение, выполняются только рекомендации от правил affinityи anti-affinity и миграция серверов в режиме обслуживании. В самом агрессивном режиме – будет выравниваться даже небольшая разница в нагрузке на сервера. Рекомендуется применять среднее значение, чтобы избежать перегрузок.

Просчитывая загруженность серверов, миграции, количество ВМ и стандартное отклонение до и после миграции, DRS выбирает наиболее оптимальные варианты переноса машин. Также необходимо учитывать стабильность нагрузки на виртуальную машину за прошедшее время.

DRSGroupsManager – создание группы серверов и ВМ и правил соотношений между ними.

Эта возможность может быть интересна по следующим соображениям:

- На одних и тех же серверах работают однотипные ВМ;
- Соответствующие распределение ВМ и серверов по производительности;
- Если сервера в кластере разной конфигурации, то самые тяжелые ВМ лучше не мигрировать на сервера другой конфигурации, чем на которых такие ВМ включились;
- Если в лицензии указано ограничение в количестве серверов;
- Если некоторые сервера имеют единую точку отказа.

Rules– здесь имеется возможность создатьaffinityи anti-affinity, а также указывать правила говорящие какие группы ВМ принадлежат каким группам серверов.

Создавая правила anti-affinity,указывается несколько виртуальных машин, которые DRSраспределяет по разным серверам. Здесь действует правило: одна ВМ группы = один сервер.

В правиле affinityуказывается произвольное количество ВМ, которыеDRSрасполагает на одном сервере. Это необходимо во избежание нагрузки физической сети и ограничения пропускной способности во время использования некоторых приложений.

Возможны несколько вариантов разделения ВМ и сервера по группам и указания правила их связи:

1. Mustrunonhostsingroup –определенная группа ВМ работает только на определенной группе серверов. Если подходящий сервер недоступен, то включение машины или её миграция не будет осуществлена.;

2. Shouldrunonhostsingroup – определенной группе ВМ необходимо быть на определенной группе серверов. Разрешена работа машины на другом сервере, если нет подходящего;

3. MustNotrunonhostsingroup –определенная группа ВМ может работать на всех серверах, кроме определенной группы;

4. ShouldNotrunonhostsingroup – определенной группе ВМ необходимо быть на любых, кроме указанной группы серверов.

Правила работы механизма:

- Не существует приоритетов правил. Работа ВМ должна удовлетворят сразу всем;
- Но при конфликте правил, в приоритете будет созданное позднее;
- Администратор не имеет прав для осуществления запуска и переноса ВМ с условиями «Mustrun…» или «MustNotrun...», если происходит конфликт правил. При конфликте даже DRS, DPM, HA не будут осуществлять операции:
- DRS не станет мигрировать виртуальную машину с сервера, который переводится в режим обслуживания;
- DRS не перестанет включать или мигрировать на неподходящий сервер даже для балансировки нагрузки;
- HA не включит ВМ, если не доступен сервер удовлетворяющий правилам;
- DPM не выключит серверов, которые из-за правил не сможет быть освобожденным;
- Правилами типа «Should…» - являются рекомендованными в общих случаях, т.к. у них нет строгих рамок и ограничении, а значит и конфликтов не возникает;

- Правила типа «Should…»являются рекомендованными, поэтому DRS может не брать их в расчет при дисбалансе нагрузки на процессоры или память серверов кластера. DRSлишь выдаст предупреждение о невыполнении правила, но переносить ВМ не будет.

VirtualMachineOptions – настройка DRS, которая дает или отнимает возможность индивидуально настроить уровень автоматизации для каждой виртуальной машины. Для того чтобы уменьшить влияние на такие ВМ со стороны DRS, такую настройку предлагают не ставить в режим автомат. Уравновешивание нагрузки будет происходит при переносе менее критичных ВМ.

VMwareEVC или EnhancedvMotionCompatibility – выполнение правил vMotion необходимо, так как для выполнения функции DRS использует механизм vMotion.Совпадение типов процессоров по поддерживаемым инструкциям – одно из основных правил. Для этого необходимо просто включить функцию EVC.

При включенииEVCвсе сервера DRS«приводятся к единому знаменателю» по функциям процессора. Это происходить если отключить те функций, которых нет хотя бы на одном сервере кластера.

Swapfilelocation – место, где по умолчанию будут хранится vmkernelswap-файл (файл подкачки). Такой файл может располагаться как на общем хранилище так и вне его. Это не повлияет на процесс миграции ВМ.

DistributedPowerManagement

Теперь перейдем к настройке DistributedPowerManagement (DPM). Это дополнительная функция кластера, при которой DRSсравнивает нагрузку на сервера. Если она будет небольшая, то есть возможность перенесения всех ВМ на одну часть сервера. При этом незадействованные сервера перейдут в режим stanby, до момента их надобности. Анализ DPMоснован на данных, которые собирает и анализирует кластер и происходит с интервалом в 5 минут. DPM основывает свой анализ на данных, полученных в течении 40-минут.

Нормальная нагрузка на сервер 63±18% по мнению DPM. Включение серверов происходит, когда превышается допустимый лимит (81%), а выключение – при падении ниже 45%.

Благодаря данной функции происходит грамотная нагрузка серверов и экономия ресурсов.

Существует два механизма включения серверов:

- BMC/IPMI (указать параметры доступа к BaseboardManagementController, работающих по протоколу IPMI)
- Wake-on-LAN (WOL).

Для первого механизма необходимо:

- IP-адрес BMC/IPMI должен быть доступен с сервера vCenter,
- на BMC должен быть создан пользователь, который будет иметь право прав включать сервер.

BMC/IPMI-контроллера по сути являются миниатюрными компьютерами, имеющие свой сетевой порт. Они имеют доступ к управлению оборудованием «главного» сервера. Данные контроллеры работают, даже если сервер неработоспособен и BMC/IPMI остается доступен по сети. Через них и можно включить сервер.

С помощью механизма Wake-On-Lan (WOL) можно попытаться включить настройки BMC/IPM, если для сервера они не указаны.Отсылка пакетовWOL будет происходить на те физические сетевые контроллеры хоста, к которым подключен vMotion-интерфейсVMkernel, одним из включенных серверов ESXi по команде vCenter.

Необходимо выполнение нескольких условий:

- Интерфейсы VMkernel с разрешенным vMotion должны быть на каждом сервере;
- Они должны находится в одной сети, и не иметь маршрутизаторов посередине;
- Должна быть поддержка WOL на том физическом сетевом контроллере, через который будет работать этот vMotion-интерфейс.

Существует три варианта настройки DPM:

- Off – выключен;
- Manual – включен с ограничениями - показывать рекомендации по выключению серверов;
- Automatic – автоматическое включение и выключение серверов, миграции виртуальных машин связанных с этими процессами. Для них можно указывать приоритеты выполнения.

8. Обеспечение избыточности

Простои, плановые они или внеплановые, приносят с собой значительные издержки. Решения для обеспечения более высокого уровня доступности традиционно являются очень дорогостоящими, трудноосуществимыми и тяжело управляемыми. VMware vSphere делает эти средства более простыми и менее дорогими, чтобы обеспечить более высокий уровень доступности для важных приложений. С vSphere, организации могут легко увеличить базовый уровень доступности, предоставляемый для всех приложений, а также обеспечить более высокий уровень доступности более легко и экономически эффективно. vSphere позволяет сократить запланированные и незапланированные простои. Революционные возможности VMwarevMotion позволяют выполнять плановое техническое обслуживание с нулевым временем простоя приложений. VMware HA, являющееся одной из возможностей vSphere, сокращает время незапланированных простоев за счет использования нескольких сконфигурированных в кластер хостов VMware ESX и VMware ESXi, чтобы обеспечить быстрое восстановление после сбоев, а также экономически эффективную высокую доступность для приложений, работающих в виртуальных машинах.

8.1. Принципы проектирования для обеспечения высокой доступности:

Ключом к архитектуре с высокой степенью доступности вычислительной среды является устранение единых точек отказа. Потенциально, сбои, происходящие в любой точке среды, могут повлиять на аппаратное и программное обеспечение. Создание избыточности в уязвимых точках позволит снизить время простоев, вызванных отказами, в том числе аварий на следующих физических слоях:

- Серверные компоненты, такие как сетевые адаптеры и хост-адаптеры шины (HBA),
- Серверы, включая лезвия и стойки лезвий,
- Сетевые компоненты,
- Массивы хранения данных и сетевые хранилища.

8.2. Выбор хоста

В целом vSphere начинается с правильного выбора хоста, включая такие элементы, как резервные блоки питания, ECC-память(англ. error-correcting code memory, память с коррекцией ошибок), удалённый мониторинг, уведомления, и так далее. Следует также уделять внимание удалению единых точек отказа в среде расположения хостов, в том числе распределению хостов между несколькими стойками лезвий, чтобы исключить возможность воздействия их сбоя на весь кластер. При развёртывании класте-

ра VMware HA лучше всего создавать кластер из одинакового серверного оборудования.

Использование одинакового оборудования обеспечивает ряд ключевых преимуществ:

- Упрощение конфигурирования и управления серверами с использованием профилей хостов.
- Увеличение возможности обрабатывать сбои сервера и уменьшение фрагментации ресурсов. Использование резко отличающегося оборудования приведет к несбалансированному кластеру, как описано в разделе "Контроль доступа". По умолчанию VMware HA готовится к худшему сценарию, что самый большой хост в кластере даст сбой. Поэтому для того, чтобы справиться с данным случаем, следует зарезервировать как можно больше ресурсов на всех узлах, что делает их использование практически невозможным.

Выбор размера кластера также оказывает значительное влияние на общую доступность и уровень возможного укрепления структуры. Более мелким кластерам требуется больший процент доступных ресурсов для того, чтобы облегчить их способность обрабатывать сбои. Более крупные кластеры могут получиться более сложными для управления с точки зрения сети и хранения. Идеальный размер кластера для большинства сред составляет от 6 до 10 хостов.

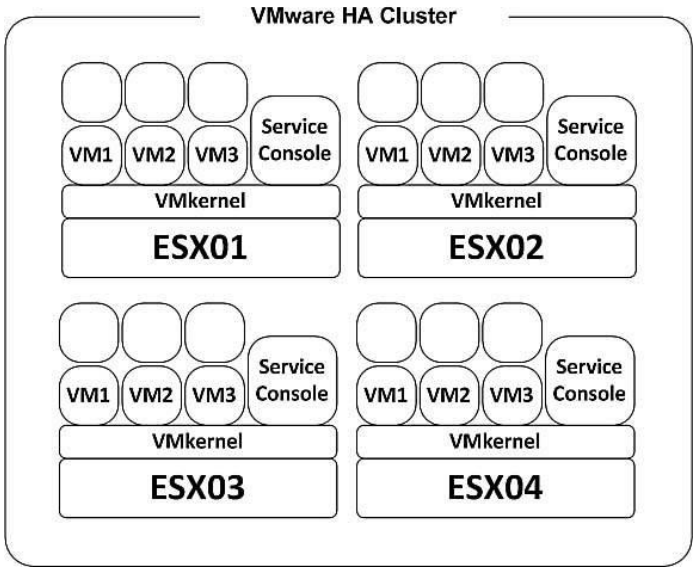

Рис. 25 Общая структура НАкластера

Создание среды с высоким уровнем доступности и основанном на кластеризации, представляет собой сложное и затратное дело. В среде VMware для включения системы аварийного переключения HA или FT потребуются сеть хранения данных, кластер из двух узлов и дополнительная сеть VMware vMotion для переноса виртуальной машины с одного узла на дру-

гой. Необходимо настроить целый ряд параметров, для того, чтобы определить, как кластер HA или FT будет вести себя в случае отказа одного из хостов, и обеспечить избыточность ресурсов для перехода виртуальной машины с одного хоста на другой.

vSphere HA сокращает время простоя за счет автоматического перезапуска виртуальных машин, но не защищает сервер от снижения производительности. При этом данные, не записанные на диск, теряются при отключении электропитания. Повторный запуск приложения зависит от его типа и не может быть мгновенным.

В контексте кластера высокой доступности (КВД) отказоустойчивость обеспечивается за счёт автоматического определения отказа оборудования и последующего запуска сервиса на исправном узле кластера.

В КВД не выполняется синхронизация запущенных на узлах процессов и не всегда выполняется синхронизация локальных дисков машин. Поэтому, использующиеся узлами носители должны размещаться на отдельном независимом хранилище, например, на сетевом хранилище данных (СХД). Делается это потому, что в случае отказа узла, с ним пропадёт соединение, а значит, не будет возможности получить доступ к информации на её накопителе. В этом случае предполагаем, что и СХД тоже должно обладать свойствами отказоустойчивости, иначе КВД не получится.

Кластер высокой доступности делится на два подкластера:

- Кластер хранилища. Там находятся диски, которые используются узлами вычислительного подкластера,
- Вычислительный. К нему относятся узлы, на которых непосредственно запущены виртуальные машины.

Выбор расположения хоста:

VMware HA использует концепцию первичных и вторичных хостов. Первичные хосты отвечают за принятие решений, обеспечивающих отказоустойчивость для кластера; вторичные хосты выполняют эти решения. Первые пять хостов ESXi, которые объединяют кластер VMware HA, обозначены в качестве первичных хостов. Все последующие хосты обозначены как вторичные хосты. Любой хост, который присоединяется к кластеру VMware HA, должен взаимодействовать с существующим первичным хостом для завершения его конфигурации, кроме ситуации, когда в кластер добавляется первый хост. По крайней мере, один первичный хост должен быть функциональным для корректной работы VMware HA. Если все первичные хосты недоступны, то никакие другие хосты не могут быть успешно настроены для VMware HA, и нет перехода на другие ресурсы.

Один из первичных хостов также обозначается в качестве активного первичного хоста. В его обязанности входит:

- Выбор расположения для перезапуска виртуальных машин,
- Отслеживание неудачных попыток перезапуска,

- Определение момента, когда необходимо продолжать пытаться перезапустить виртуальную машину.

Есть три сценария, которые могут вызвать недоступность первичного хоста в кластере VMware HA: вход в режим технического обслуживания; выключение питания; и ошибки.

- Вход в режим технического обслуживания:

Если первичный хост входит в режим обслуживания, вторичный хост, если таковой имеется, будет повышен до нового первичного хоста.

- Отказ и выключение питания:

Если первичный хост отключился или дал сбой, то общее число первичных хостов уменьшается на единицу. Ни один вторичный хост не будет повышен, чтобы стать новым первичным хостом в данном случае. Таким образом, VMware HA будет иметь возможность обрабатывать не более четырех последовательных неудач первичных хостов.

Если все основные хосты прекратят функционировать, кластер не сможет обеспечить отказоустойчивость VMware HA. Для того, чтобы предотвратить потерю всех первичных хостов из-за отказа одного физического компонента, настоятельно рекомендуется, чтобы кластер был размещен таким образом, чтобы не более четырех хостов размещались в одной серверной стойке или блэйд шасси.

8.3. Сетевые вопросы проектирования:

Рекомендации по дизайну сети можно разделить на два типа: повышение отказоустойчивости "на стороне клиента" сети, для обеспечения доступа из внешних систем для рабочих нагрузок, работающих в vSphere; и повышение отказоустойчивости соединений, используемых самой VMware HA.

Основное сетевое руководство:

Следующие предложения являются общими рекомендациями по настройке сети для улучшения доступности:

- Настройка коммутаторов. Если сетевые коммутаторы, которые соединяют физические серверы, поддерживают PortFast или эквивалентные настройки, включите их. Этот параметр предотвращает неправильное определение хостом того, что сеть изолирована, во время выполнения алгоритмов SpanningTree при загрузке системы.
- Отключение мониторинга хоста (с помощью VMware vCenterServer, снимите флажок "“EnableHostMonitoring" в диалоговом окне настройки кластера, VMware HA->HostMonitoringStatus) при выполнении любых работ по обслуживанию сетей, которые могут отключить все пути heartbeat между узлами сети и привести к изоляции ответов.

- Использование DNS для разрешения имён вместо приводящего к ошибкам метода ручного редактирования локального /etc/hosts / файла на ESXi хостах. Если вы делаете редактирование /etc/hosts /, вы должны включить как длинные, так и короткие имена. VMwareHA модифицирует файл /etc/hosts, если содержащаяся в нем информация является неверной или если нужно что-либо добавить.
- Используйте последовательные имена портов виртуальных локальных сетей для виртуальных сетевых машин на всех ESXi хостах в кластере. Имена портов используются виртуальными машинами для определения совместимости сети. VMware HA проверит, совместим ли хост, перед тем как инициировать отказоустойчивость. Если нет хостов с соответствующими доступными групповыми именами портов, то отказоустойчивость гарантировать невозможно.
- Использование документированной схемы именования настоятельно рекомендуется.
- Использование объединения двух сетевых адаптеров, подключенных к разным физическим коммутаторам, может повысить надежность сети управления. Поскольку серверы соединены друг с другом с помощью двух сетевых адаптеров и с помощью двух отдельных коммутаторов, то они имеют два независимых пути для передачи и приёма heartbeats, а следовательно кластер является более эластичным. (рис. 26).

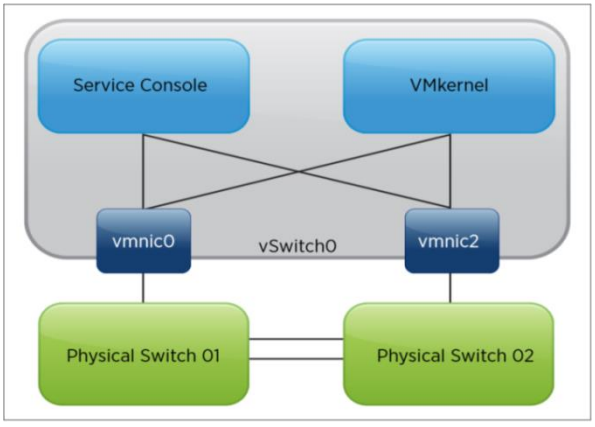

Рис. 26 NICteaming

8.4. Дизайн систем хранения данных:

Для поддержания постоянной связи междуESXi хостом и его устройствами хранения, ESXi поддерживает многоканальность. Многоканальность является методом, который позволяет использовать более одного физического пути, передающего данные между хостом и внешним запоминающим устройством.

В случае выхода из строя какого-либо элемента в сети SAN, такого как адаптер, коммутатор или кабель, ESXi может переключиться на другой фи-

зический путь, который не использует неисправный компонент. Этот процесс переключения пути, чтобы избежать вышедших из строя компонентов, известен как путь перехода на другой ресурс.

В дополнение к пути перехода на другой ресурс, многоканальность обеспечивает балансировку нагрузки. Балансировка нагрузки является процессом распределения нагрузок ввода / вывода по нескольким физическим путям. Балансировка нагрузки уменьшает или устраняет потенциальные узкие места.

Для конфигураций FibreChannel SAN, многоканальные настройки очень специфичны для выбора HBA (Хост-адаптера шины), коммутаторов и массивов компонентов.

Для конфигурацииiSCSI,ESXi поддерживают создание второго iSCSI инициатора для обеспечения многоканальных конфигураций.

VMware настоятельно рекомендует несколько путей к ресурсам хранения для обеспечения максимальной устойчивости.

8.5. VMware vSphere FaultTolerance

Бесперебойное предоставление клиенту услуг возможно только в случае присутствия в любой момент времени точной копии физического или виртуального сервера, на котором запущен сервис. Если произойдет сбой, и мы попытаемся создавать копию уже после отказа оборудования, то на это уйдет значительное в рамках беспрерывного обслуживания время, а значит, будет перебой в предоставлении сервиса. К тому же, после выхода из строя виртуальной машины невозможно будет получить содержимое оперативной памяти с неё, что означает, что находившаяся там информация будет потеряна.

CA (бесперебойное предоставление) реализуется двумя способами: аппаратным и программным.

Аппаратный способ представляет собой “дублированный” сервер: все компоненты раздвоены, а вычисления выполняются одновременно и независимо. За синхронность отвечает узел, который в числе прочего сверяет результаты с серверов. В случае рассинхронизации выполняется поиск причины и попытка коррекции ошибки. Если ошибка не исправляется, то несогласованный модуль отключается.

Программный способ.

Самый популярный инструмент для развёртывания кластера непрерывной доступности - vSphere от VMware. Именно технология “FaultTolerance” обеспечивает“ContinuousAvailability”.

В случае отказа сервера VMwarevSphere,FaultTolerance (vSphereFT) продолжает предоставлять непрерывную доступность для приложений с максимумом виртуальных CPU равным 4. Делает он это, создавая “тенево-

го” клона виртуальной машины, который всегда находится в актуальном для основной машины состоянии. В случае ошибки оборудования, vSphereFT автоматически запускает перемещение виртуальной машины на рабочий узел, обеспечивая нулевое время простоя и предотвращая потерю данных. Как и vSphereHA, оно защищает от ошибок оборудования, но полностью устраняет время простоя с помощью мгновенного переключения и восстановления. После перемещения, vSphereFT автоматически создаёт новую, дополнительную виртуальную машину для продолжения предоставления защиты приложения.

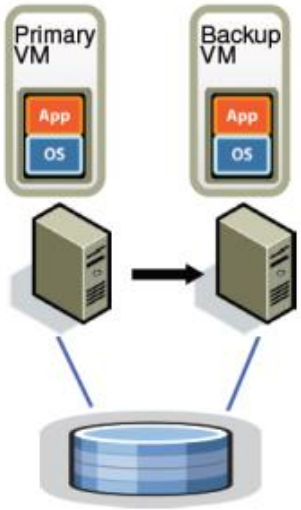

Рис. 27 Процедура работы FT

Технология vSphereFTимеет следующие достоинства:

- Защита критически важных высокопроизводительных приложений независимо от операционной системы
- Предоставление постоянной доступности с нулевым временем простоя и нулевой потерей данных в случае отказа инфраструктуры.
- Обеспечивает полностью автоматический отклик на ошибки в работе.

При виртуализации vCenterServer, такие технологии как vSphereFT могут помочь защитить сервер менеджмента vCenter от отказов оборудования. В сравнении с vSphereHA, vSphereFT может предоставлять мгновенную защиту, но при этом должны быть соблюдены следующие ограничения:

- Система vCenterServer ограничена четырьмя vCPU (виртуальными процессорами)
- vSphereFT защищает от отказа оборудования, но не отказа приложений
- vSphereFT не может уменьшить время простоя для отключений, связанных с установкой обновлений.
- у vSphereFT есть требования к ресурсам, которые могут вызвать дополнительные расходы.

Так как vSphereFT подходит только для рабочих станций с максимум четырьмя vCPU и 64 гигабайтами памяти, то оно может использоваться только на ”малых” vCenterServer решениях.

8.6. Контроль доступа:

VMware vCenter Сервер использует VMware HA управление доступом, чтобы обеспечить наличие достаточных ресурсов в кластере, зарезервированных для виртуального восстановления машины в случае выхода хоста из строя. Управление доступом предотвратит посягательства на ресурсы, зарезервированные для виртуальных машин, перезапущенных из-за сбоя.

Этот механизм настоятельно рекомендуется для того, чтобы гарантировать доступность виртуальных машин. Начиная с vSphere 4.0, VMware HA предлагает три варианта конфигурации для выбора стратегии управления допуском:

- HostFailuresClusterTolerates политики (по умолчанию): VMware HA гарантирует, что на определённом количестве хостов может произойти сбой, и достаточные ресурсы остаются в кластере после отказа всех виртуальных машин из этих хостов.
- Процент ресурсов кластера, зарезервированных в качестве отказоустойчивых резервных мощностей: VMwareHA гарантирует, что определенный процент от совокупных ресурсов кластера зарезервирован для перехода на другой ресурс. Эта политика рекомендуется для ситуаций, когда виртуальные машины должны быть размещены с существенно различающимися ресурсами процессора и памяти в одном кластере, или там, где имеются хосты с различными параметрами процессора и объема памяти.
- Указание отказоустойчивого узла: VMware HA обозначает конкретный хост в качестве отказоустойчивого хоста. Когда хост выходит из строя, VMware HA пытается перезапустить свои виртуальные машины на указанном отказоустойчивом хосте.

Передовые практики, рекомендации (Best-practices recommendation) от сотрудников VMware для управления допуском выглядят следующим образом:

- Выберите "процент зарезервированных ресурсов кластера " для управления допуском. Эта политика обеспечивает максимальную гибкость с точки зрения хоста и масштабирования виртуальной машины. В большинстве случаев применяется простой расчет 1 / N, где N = общее количество хостов в кластере, позволяющее получить адекватную масштабируемость.
- Убедитесь, что все узлы кластера имеют равные размеры. "Несбалансированные" узлы приводят к избыточным мощностям, резервируемым для обработки выхода из строя наибольшего возможного узла.

- Попытайтесь сохранить требования к масштабируемости виртуальных машин для всех настроенных виртуальных машин в системе. Host Failures Cluster Tolerates политика (Допуск отказа хостов кластера) использует понятие размеров слотов для расчета количества мощности, необходимой для резервирования для каждой виртуальной машины. Размер слота основан на величине зарезервированной памяти и центрального процессора, необходимой для любой виртуальной машины. Различные требования к ресурсам процессора и памяти виртуальных машин приводят к вычислению размера слота по умолчанию в максимально возможном размере для всех виртуальных машин, ограничивая гибкость и консолидацию.

8.7. Средства кластеризации на уровне гостевой ОС:

Для организации данного типа защиты потребуется основная и резервная ВМ, которые лучше разместить на разных хостах vCenter. С помощью кластера WindowsServerFailoverClustering(WSFC) организуется кворумный диск, общий для виртуальных машин, на котором размещены данные vCenter. В случае сбоя происходит переключение на резервную ВМ, которая продолжает работу с общим диском. Этот метод полностью поддерживается, начиная с vCenter 6.0.

Несколько приложений используют кластеризацию, включая приложения без определения их расположения на диске, такие как веб-сервера и приложения с встроенными функциями восстановления, например сервера баз данных. Типичная настройка кластеризации включает диски, которые распределены между виртуальными машинами. В кластере виртуальных машин между физическими хостами, распределённый диск должен поддерживать FibreChannel (FC) SAN, FCoE, или iSCSI. Между узлами необходима частная сеть “heartbeats”.

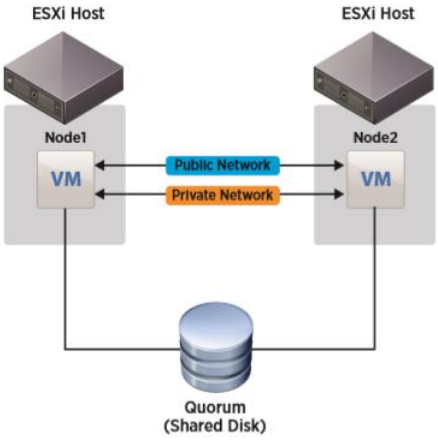

Рис. 28 Установка гостевой кластеризации.

На рисунке изображены две виртуальные машины с запущенной программой кластеризации на уровне операционной системы с контролем и исправлением. Виртуальные машины соединены частной сетью

«heartbeats» и публичной сетью при поддержке распределённого хранилища.

Данное решение использует основную и запасную виртуальные машины для защиты конкретного приложения. Фреймворк кластера контролирует состояние ресурсов приложения. Если настроенное приложение или связанные с ним сервисы становятся недоступны, то сервисы кластера перемещаются на запасной узел. Запланированные простои, связанные с обновлением операционной системы или приложения, защищены данным решением, так как приложения могут быть перенесены на запасные виртуальные машины во время обновления операционной системы, что позволит сократить время простоя до минимума.

8.8. Система копирования VMwarevSphere

VMware vSphereReplication может копировать виртуальные машины в пределах одного сайта (участка) или нескольких сайтов (участков), чтобы добавить ещё один слой защиты. Для осуществления восстановления виртуальной машины, vSphereReplication требуются включённые vCenterServer и веб-клиент vSphere. Даже с данными требованиями vSphereReplication может использоваться для защиты vCenterServer. В качестве примера возьмём несколько систем vCenterServer – одна из них, которая всё ещё доступна, может использоваться для восстановления отказавших систем vCenterServer – у неё может быть такое же физическое расположение или другое.

В данном случае виртуальная машина, на которой запущен vCenterServer, будет скопирована на удалённую сторону, обозначенную как сайт (участок) восстановления. vCenterServer может быть восстановлен на скопированной стороне и использован так долго, сколько будет доступен его IP адрес и подсеть.

8.9. Защита данных VMwarevSphere

VMware vSphereDataProtection это решение для восстановления и резервного копирования включённое во все версии vSphere 6.0. Оно представлено как виртуальное устройство и основано на ведущей технологии EMCAvamar. VMware vSphereDataProtection не имеет агента, а использует снапшоты виртуальной машины для резервного копирования и восстановления всей виртуальной машины, отдельных файлов диска виртуальной машины (VMDK) и отдельных файлов внутри виртуальной машины.

VMware vSphereDataProtection контролируется с использованием веб-клиента vSphere. Если сервер vCenter и соответствующий ему веб клиент vCenterServer станет недоступным, то EmergencyRestore (аварийное восстановление) может восстановить виртуальные машины, включая те, на которых запущены компоненты vCenterServer. EmergencyRestore включает восстановление виртуальной машины без необходимости в сервере vCenter и веб клиента vSphere. Это делает данное решение полезным для резервного

копирования компонентов vCenterServer, когда они запущены на одной или нескольких виртуальных машинах.

VMware vSphere Data Protection использует Windows Volume Copy Service (VSS) встроенный в VMware Tools.При выполнении резервной копии виртуальной машины с Windows, приложения для которых был установлен VSSwriter – такие как MicrosoftSQLServer и файловая система Windows – “замораживаются” сразу перед тем, как создаётся снапшот для резервного копирования. Результатом такого метода получаются бэкапы уровня приложения и уровня файлов.

Рекомендации по защите vCenter Server помощьюvSphere Data Protection:

1. Запустите все компоненты vCenterServer на одной или нескольких виртуальных машинах

2. Убедитесь, что DNS правильно настроен для всех виртуальных машин vCenterServer, хостов vSphere и виртуальных приложений vSphereDataProtection. Решение должно позволять использовать доменные имена (FQDN), короткие имена и зарезервированные lookups для каждой виртуальной машины.

3. Разверните vSphereDataProtection на том же кластере, что и vCenterServer

4. Создайте бекап всех систем, запущенных на виртуальной машине для всех виртуальных машин, которые содержат и поддерживают компоненты vCenterServer. Создание бэкапов только для виртуальных машин с vCenterServer, делает ручную процедуру создания бэкапов проще, в дополнение к запланированным бэкапам, которые происходят перед обновлением виртуальных машин и компонентов vCenterServer.

5. Сделайте расписание бэкапов с ежедневным созданием резервного копирования и с политикой сохранения их как минимум на протяжении 10 дней.

6. Создайте расписание бэкапов для vCenterServer, рассчитанное на то, что нагрузка в момент резервного копирования будет меньше всего.

7. Настройте аварийную сигнализацию vCenterServer для оповещения администраторов, когда защищаемая виртуальная машина работает на снапшоте.

8. Настройте аварийную сигнализацию vCenterServer для оповещения администраторов, когда защищаемой виртуальной машине требуется обновление.

9. Настройте оповещение по электронной почте для предоставления информации о состоянии приложения vSphereDataProtection и его бэкапов

Практическая часть

Обзор уязвимостей VMware ESXi.

В период с 2012 по 2016 год по данным Национальной базы данных угроз в VMware ESXi была обнаружена 31 уязвимость.

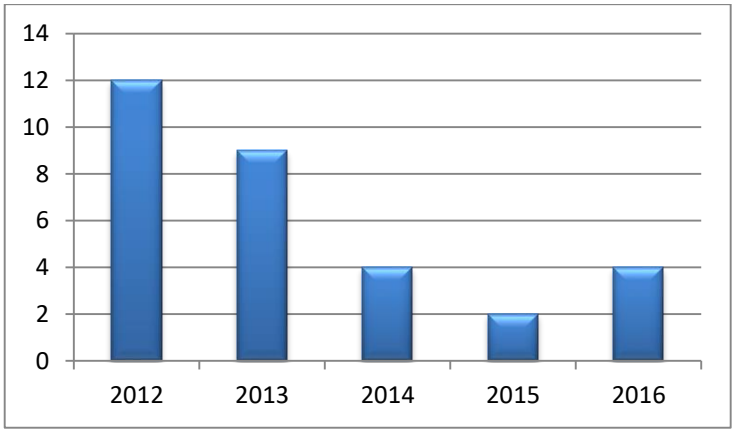

Рис. 29 Количество уязвимостей по годам.

Как видно, наиболее уязвимыми были версии, используемые в 2012 году: ESXi3.5, 4.0, 4.1 и 5.0. Разработчики гипервизора работали над устранением обнаруженных уязвимостей, что привело к тому, что в 2013 все эти уязвимости были исправлены. Так же уязвимости 2012 года отличает высокий уровень CVSS-оценки (CVSS - общая система подсчета уязвимости), что означает значительный вред при использовании данных уязвимостей злоумышленниками. Средняя оценка обнаруженных за данный год уязвимостей 8. Как видно по графику, представленному на рис. 30 большинство этих угроз относится к типу "Отказ в обслуживании (DoS-атаки)".

Рис. 30 Распределение угроз по их типу за 2012 год.

Самая серьезная уязвимость, обнаруженная в 2012 году позволяют удаленному злоумышленнику выполнить произвольный код на хост или вызвать отказ в обслуживании и повреждение памяти на хосте через со-

зданный файл контрольных точек. В следствии этой уязвимости происходит общее раскрытие информации, в том числе всех системных файлов, система компрометируется, из-за полной потери защиты. Так же злоумышленник может сделать ресурс полностью недоступным. Существовала данная уязвимость из-за недостаточной проверки данных при загрузке виртуальной машины из файлов контрольной точки.

В 2013 году число найденных уязвимостей снизилось на четверть, а средняя оценка снизилась до 7,2. Большинство из них были исправлены в этом же году. Как и в предыдущем году, большая часть относилась к DoS-атакам.

Именно в этом году была обнаружена уязвимость, имеющая наивысшую CVSS-оценку (10.0) - CVE-2013-1405. К уязвимым продуктам относятся ESXi версий 3.5, 4.0 и 4.1. из-за ошибки в реализации протокола аутентификации у злоумышленника была возможность отправить специально сформированный пакет авторизации, вызывающий переполнение буфера и отказ в обслуживании. Помимо того, что данная уязвимость приводит к раскрытию системных файлов, происходит полная потеря защиты системы и злоумышленник имеет возможность сделать ресурс полностью недоступным, она обладает низкой сложностью доступа: не требуется специализированный допуск, как и наличие особых знаний и навыков. Это делает уязвимость еще более опасной.

В период с 2014 по 2015 год количество обнаруженных уязвимостей снизилось до шести, что позволяет сделать вывод о серьезной работе над безопасностью VMwareESXi. На это же указывает снизившаяся до 4,6 средняя оценка всех уязвимостей за эти два года.

Наиболее серьезная уязвимость из найденных работает на версиях 5.0, 5.5 и 6.0 и позволяет пользователю хоста ОС получить привилегии или вызвать отказ в обслуживании путем изменения файла конфигурации. В этой уязвимости нет влияния на конфиденциальность системы, существует пониженная производительность или перебои в доступности ресурсов, без его полного отключения, и у злоумышленника нет полного контроля над изменением системных файлов или информации, но для данной угрозы не требуется аутентификация и особенные знания, навыки и ресурсы.

Из четырех уязвимостей, обнаруженных в 2016 году, две имели совершенно новые типы: межсайтовый скриптинг (XSS) и расщепление HTTP-запроса. Первый тип атаки позволяет удаленному аутентифицированному пользователю вводить произвольный веб-скрипт с помощью созданной ВМ, как результат злоумышленник может изменять некоторые системные файлы и информацию. Оценка данной уязвимости 3,5. Вторая уязвимость имеет оценку выше - 4,3. С помощью CRLF-инъекции удаленный злоумышленник может внедрить произвольные заголовки HTTP и проводить ответные HTTP-атаки расщепления через неуказанные векторы. В отличии о предыдущей уязвимости в данной злоумышленнику не нужна

аутентификация в системе. Данное несовершенство присутствует в продуктах VMwarevCenterServer 6.0 и ESXi 6.0

Наиболее серьезная уязвимость, найденная в последнее время так же относится к атаке типа "Отказ в обслуживании". Она позволяет через VMware Tools HGF(так называемый Shared Folders) гостевой ОС получить привилегии или вызвать отказ в обслуживании с помощью разрушения памяти ядра гостевой ОС. При этом происходит существенное раскрытие информации, возможна частичная модификация файлов и информации, снижается производительность и возникают перебои в доступности ресурсов.

Информация по данным уязвимостям была взята из Национальной базы данных уязвимостей правительства США, которая включает в себя базы списков проверки безопасности, уязвимостей ПО и конфигураций, списки продуктов и метрики для оценки защищенности.

Для максимального обеспечения безопасности в Центрах обработки данных, основанных на ПО компании WMware, необходимо детально и пошагово проработать механизм создания и настройки виртуальных систем. Для этого был разработан комплекс лабораторных работ, позволяющих обеспечивать безопасность систем, даже при наличии базового уровня знаний у студентов (Приложение 1).

При разработке лабораторных работ были использованы материалы VMware Education Services. В данных методических материалах отражены этапы от установки ESXi до настройки и обеспечения безопасности виртуальной системы.

Заключение

В последнее время именно безопасность является одним из ключевых факторов при принятии решения об использовании технологий виртуализации. В условиях необходимости защиты конфиденциальной информации виртуальные машины требуют повышенного внимания, хотя они и, напротив, могут использоваться для обеспечения безопасности (например, для изоляции критически важных систем друг от друга). В то же время, виртуализация сама по себе, как еще не опробованная технология, таит в себе множество опасностей. Вредоносное ПО, использующее виртуализацию, может в ближайшем будущем являться угрозой не только для организаций, но и для конечных пользователей. Поэтому при использовании виртуализации необходимо грамотно спланировать стратегию защиты виртуальной инфраструктуры. Простая развертываемость виртуальных систем требует постоянного контроля, поскольку «забытые» и необновляемые системы могут являться точкой проникновения злоумышленников во внутреннюю сеть компании. К тому же, нельзя забывать об инсайдерских угрозах - необходимо правильно разграничивать права доступа персонала к информационным ресурсам, содержащих виртуальные системы. В больших масштабах нужно использовать специализированное ПО для контроля за ИТ-инфраструктурой и средства обнаружения вторжений.

Большое количество уязвимостей, найденных за последнее время в платформах виртуализации, говорит о том, что средства защиты информации совершенствуются, но при этом появляются новые угрозы, поэтому изучение способов защиты информации одна из приоритетных задач обучения. Исходя из данной задачи, был разработан комплекс лабораторных работ по предмету "Защита информации в Центрах обработки данных".

Список использованных источников и литературы

1. Кусек К., Ван Ной В., Дэниел А. Администрирование VMware vSphere 5. - СПБ: Питер, 2013.
2. Михеев М.О. Администрирование VMware vSphere 5. - М.: ДМК Пресс, 2012.
3. Matthew Portnoy. Virtualization Essentials. - Sybex, 2012.
4. Nick Marshall and Scott Low. Mastering VMware vSphere 6. - Sybex, 2015.
5. https://habrahabr.ru
6. https://pubs.vmware.com/vsphere-60/index.jsp?topic=%2Fcom.vmware.vsphere.doc%2FGUID-1B959D6B-41CA-4E23-A7DB-E9165D5A0E80.html
7. https://ru.scribd.com/document/271906081/VMware-vSphere-Install-Configure-Manage-V6-Student-Lab-Manual-pdf
8. https://www.cvedetails.com/product/22134/Vmware-Esxi.html?vendor_id=252
9. http://www.delphiplus.org/sistemy-khraneniya-dannykh-v-windows/812-protokol-iscsi.html
10. https://www.vmware.com/ru.html