

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры)

09.04.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Информационные системы и технологии корпоративного управления

(направленность (профиль)/специализация)

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему «Методы и архитектурные решения виртуализации серверной инфраструктуры»

Студент

М.Г. Колоколов

(И.О. Фамилия)

(личная подпись)

Научный
руководитель

А.В. Очеповский

(И.О. Фамилия)

(личная подпись)

Руководитель программы д.т.н., доцент, С.В. Мкртычев

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

« » 20 г.

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский

« » 20 г.

(личная подпись)

Тольятти 2019

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
Глава 1 ПОНЯТИЕ И ОСОБЕННОСТИ АРХИТЕКТУРНЫХ РЕШЕНИЙ ВИРТУАЛИЗАЦИИ СЕРВЕРНОЙ ИНФРАСТРУКТУРЫ	10
1.1 Современные методы организации серверной инфраструктуры.....	10
1.2 Понятие и значение методов виртуализации	18
1.3 Исторические аспекты развития технологий серверной виртуализации.	26
1.4 Практическое применение технологий виртуализации серверной инфраструктуры	28
Выводы по 1 главе.....	35
Глава 2 АНАЛИЗ СОВРЕМЕННЫХ РЕШЕНИЙ ВИРТУАЛИЗАЦИИ СЕРВЕРНОЙ ИНФРАСТРУКТУРЫ	36
2.1 Программное и аппаратное обеспечение в современных решениях серверной виртуализации.....	36
2.2 Платформы виртуализации серверной инфраструктуры.....	44
2.3 Проблематика и специфика внедрения технологий виртуализации	52
Выводы по 2 главе.....	57
Глава 3 ИССЛЕДОВАНИЕ МЕТОДОВ И АРХИТЕКТУРНЫХ РЕШЕНИЙ ВИРТУАЛИЗАЦИИ СЕРВЕРНОЙ ИНФРАСТРУКТУРЫ	58
3.1 Выбор технологий для сравнительного анализа виртуализации серверной инфраструктуры	58
3.2. Средства управления и администрирования виртуальной средой	68
3.3. Сравнение показателей технологических решений серверной виртуализации	75
Выводы по 3 главе.....	94
Глава 4 АНАЛИЗ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ АРХИТЕКТУРНЫХ РЕШЕНИЙ ВИРТУАЛИЗАЦИИ СЕРВЕРНОЙ ИНФРАСТРУКТУРЫ	95

4.1 Обобщение результатов сравнительного исследования технологий виртуализации серверной инфраструктуры	95
4.2 Рекомендации по внедрению, использованию и поддержке технологий виртуализации серверной инфраструктуры	99
Выводы по 4 главе.....	101
ЗАКЛЮЧЕНИЕ	103
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	105

ВВЕДЕНИЕ

Эффективная и успешная работа каждого предприятия, как правило, обуславливается грамотным и рациональным формированием информационной инфраструктуры предприятия. Главные аспекты ее формирования определяются рядом условий, которые включают в себя масштаб, область и направление деятельности предприятия. Информационная структура отвечает за каждый рабочий процесс, который происходит на предприятии. Данная инфраструктура представляет собой главный фактор, напрямую оказывающий влияние на интенсивное развитие и качество функционирования бизнес-деятельности, а также на ее эффективность на предприятии.

В работе каждого предприятия важно получить взаимосвязь всевозможных факторов в бизнес-процессе именно во время реализации функций предприятия. Также наиважнейшим фактором выступают информационные технологии, которые способствуют росту эффективности и вместе с тем рассматриваются в роли существенно значимого нематериального актива предприятия. Оптимальный результат достигается лишь тогда, когда эти факторы объединяются и работают вместе в одной ИТ-инфраструктуре.

Термин «информационная инфраструктура» берет свое начало от термина «инфраструктура» и информационные технологии (ИТ). Инфраструктура, в свою очередь, представляет собой совокупность связанных между собой обслуживающих структур, которые составляют и обеспечивают так называемую базу для того, чтобы были приняты те или иные решения проблем.

Программно-аппаратный комплекс необходимый для того, чтобы обеспечить все возможные бизнес-процессы предприятия, по сути, и является ИТ-инфраструктурой. На сегодняшний день инфраструктура организации состоит из локальной сети, которая в обязательном порядке включает в себя

автоматизированные рабочие места, сетевое и серверное оснащение в виде оборудования, процессы взаимодействия информационных потоков.

Отметим и то, что важным элементом ИТ-инфраструктуры любого предприятия являются используемые вычислительные ресурсы, которые обеспечивают и организуют автоматизацию деятельности предприятия. При этом, следует учитывать, что современный мир предлагает настолько мощные компьютеры, где на один достаточно серьезный сервер, можно виртуальным образом сделать перенос функционирующих серверов любой организации. Несомненный плюс всего этого заключается в том, что будет сэкономлена существенно значимая сумма затрат на их содержание.

Поэтому сегодня так заметен сильный интерес к технологиям виртуализации, притом, в основном вычислительные мощности существующих сегодня компьютеров сильно превосходят потребности предприятия, что в результате способствует возникновению таких проблем, как:

- необоснованное увеличение затрат на инфраструктуру;
- не дающая результатов эксплуатация серверов;
- трудности архивации и резервного копирования информационных данных;
- трудности в управления ИТ-инфраструктурой.

Применение в работе ИТ-технологий, которые позволяют виртуальным образом делить вычислительные ресурсы среди программного обеспечения позволяет соответствующим и наилучшим образом использовать все вычислительные возможности рабочих станций и серверов.

Технологии виртуализации, можно сказать, выступают основным курсом развития комплекса аппаратно-программных средств на сегодняшний день. С помощью этой технологии можно установить немалое число операционных систем и приложений в одном месте, а также и дистанционно. Важно отметить то, что физические и географические ограничения совершенно не играют какой-либо

роли. Кроме того, значительно снижаются затраты на энергию и затраты в целом. Все это происходит за счет наиболее оптимального применения аппаратных ресурсов, виртуальная инфраструктура предоставляет высокую доступность ресурсов, наиболее продуктивную систему управления сервером, усиленную защиту и совершенствованную систему восстановления в критических состояниях.

Таким образом, особенности организации информационной инфраструктуры и место в этой структуре серверной виртуализации обусловили выбор темы исследования, определили цель, объект и предмет исследования, а проблема неэффективного использования серверных ресурсов определили **актуальность** данного исследования.

Цель исследования: анализ методов и архитектурных решений визуализации серверной инфраструктуры.

Объект работы: виртуализация серверной инфраструктуры.

Предмет работы: методы и архитектурные решения виртуализации серверной инфраструктуры.

В процессе исследования была выдвинута **рабочая гипотеза:** виртуализация серверной инфраструктуры позволяет более эффективно использовать программно-аппаратные ресурсы, обеспечивает более равномерное распределение аппаратных нагрузок, повышает надежность информационной структуры организации.

Задачи работы:

- изучить понятие и особенности технологий виртуализации серверной инфраструктуры;
- провести анализ современных решений виртуализации серверной инфраструктуры;
- исследовать методы и архитектурные решения виртуализации серверной инфраструктуры;

– проанализировать результаты исследования архитектурных решений виртуализации серверной инфраструктуры.

Теоретической и методологической основой магистерского исследования послужили работы отечественных и зарубежных ученых, которые изучали особенности информационной инфраструктуры: А. В. Богданов – новые возможности виртуализации; Т. М. Джонс – виртуализация встроенных систем; С. А. Сушков – проблемы внедрения технологий виртуализации; Д. Гачко – использование сервисных архитектур в бизнесе; А. И. Попов – виртуальные машины и сети; А. А. Сулейманов – возможности технологий серверной виртуализации; Д. Силаков – виртуализация на платформе x86; Г. И. Зверев – угрозы и методы обеспечения информационной безопасности виртуальных сред; А. Самойленко – понятия виртуализации и виртуальная машина; К. Несс – практические решения виртуализации; В. Зорин – технологии виртуализации и защищенность информационных систем.

Методологической основой исследования послужили: научный аппарат информатики, методы исследования программного обеспечения, и такие научные методы исследования, как: абстрактно-логический метод; анализ научной и учебной литературы; классификация; метод обобщения; описательный метод; проектный метод; синтез; системный анализ и подход; сравнительный анализ; тестирование; эмпирический метод.

Научная новизна. В результате проведенных исследований был проведен анализ методов серверной виртуализации, что позволило определить наиболее эффективные методы, которые можно использовать для внедрения в деятельность предприятий, что позволит оптимизировать их информационную инфраструктуру.

Теоретическая значимость диссертационного исследования заключается в развитии информационных технологий в направлении совершенствования методологического обеспечения процессов поддержки и обслуживания

информационной инфраструктуры предприятий при внедрении методов серверной виртуализации.

Практическая значимость диссертационного исследования заключается определении эффективных методов серверной виртуализации. Практическая значимость направлена на совершенствование информационной инфраструктуры предприятий за счет внедрения методов серверной виртуализации.

На защиту выносятся:

1. Концепции, особенности и направления развития архитектурных решений виртуализации серверной инфраструктуры.
2. Методы виртуализации серверной инфраструктуры.
3. Средства и методы исследования архитектурных решений виртуализации серверной инфраструктуры.
4. Результаты исследования архитектурных решений виртуализации серверной инфраструктуры.

Апробация. Экспериментальная оценка основывалась на базе оценки разных методов виртуализации серверной инфраструктуры. В результате апробации подтверждена гипотеза исследования в том, что виртуализация серверной инфраструктуры позволяет более эффективно использовать программно-аппаратные ресурсы, обеспечивает более равномерное распределение аппаратных нагрузок, повышает надежность информационной структуры организации.

Диссертация состоит из введения, четырех глав, заключения, списка литературы и приложений. Работа изложена на 114 страницах, содержит 24 рисунка и 13 таблиц.

В первой главе рассматриваются понятия и особенности методов виртуализации серверной инфраструктуры: современные методы организации серверной инфраструктуры; понятие и значение методов виртуализации;

исторические аспекты развития методов серверной виртуализации; практическое применение методов виртуализации серверной инфраструктуры.

Во второй главе проводится анализ современных решений виртуализации серверной инфраструктуры: программное и аппаратное обеспечение в современных решениях серверной виртуализации; платформы виртуализации серверной инфраструктуры; проблемы и специфика внедрения методов виртуализации.

В третьей главе проводится исследование методов и архитектурных решений виртуализации серверной инфраструктуры: выбор технологий для сравнительного анализа виртуализации серверной инфраструктуры; средства управления и администрирования виртуальной средой; сравнение показателей технологических решений серверной виртуализации.

В четвертой главе проводится анализ результатов исследования архитектурных решений виртуализации серверной инфраструктуры: обобщение результатов сравнительного исследования технологий виртуализации серверной инфраструктуры; рекомендации по внедрению, использованию и поддержке технологий виртуализации серверной инфраструктуры.

В заключении подводятся итоги выполненной работы.

ГЛАВА 1 ПОНЯТИЕ И ОСОБЕННОСТИ АРХИТЕКТУРНЫХ РЕШЕНИЙ ВИРТУАЛИЗАЦИИ СЕРВЕРНОЙ ИНФРАСТРУКТУРЫ

1.1 Современные методы организации серверной инфраструктуры

В настоящее время в корпоративных информационных системах появляется все больше задач, предъявляющих повышенные требования к качественным и количественным характеристикам вычислительных ресурсов. Возрастающие потребности в вычислительной мощности систем обработки данных современного предприятия сдерживаются недостатком площадей для размещения оборудования, ресурсов инженерной инфраструктуры, необходимых для обеспечения качественного энергопитания и параметров окружающей среды. Увеличение сложности систем предъявляет более высокие требования к их обслуживанию. Все указанные факторы определяют формирование новых требований к серверной инфраструктуре в организациях, что позволяет обеспечивать оптимизацию функционирования предприятий. Решением проблем оптимизации информационной инфраструктуры организации, связанных со сложностью, безопасностью, надёжностью и дороговизной компонентов информационной системы, может служить совершенствование организационной структуры организации (разграничение обязанностей участников, формирование новых требований к информационной инфраструктуре), обновление средств программно-аппаратного комплекса, модернизации серверной инфраструктуры и систем хранения данных. использование услуг аутсорсинга, внедрение технологий серверной виртуализации.

Использование технологий серверной виртуализации является одним из наиболее эффективных способов оптимизации информационно инфраструктуре любой организации, поскольку позволяет перенести ряд служб в виртуальное

пространство, при этом обеспечив более равномерное распределение нагрузки на аппаратные ресурсы, а также обеспечив процессы сохраняемости данных, которые в виртуальной среде осуществляются значительно быстрее и надежнее.

Более подробно рассмотрим основные понятия, которые относятся к технологиям организации серверной инфраструктуры. Так, Гачко Д. и Стародубцев А. дают следующие определения элементам серверных инфраструктур [35]:

- сервер – компьютерное оборудование, предназначенное для поддержки функционирования информационных сервисов;
- виртуализация – технология представления вычислительных ресурсов, абстрагированных от аппаратной части;
- гипервизор – программа для реализации виртуализации;
- система хранения данных – комплекс оборудования, предназначенного для консолидации дискового пространства в рамках одной системы.

Одним из основных подходов к реализации серверной инфраструктуры, помимо использования множества реальных серверов и других аппаратных средств, является технология виртуализации - предоставление вычислительных ресурсов, абстрагированное от их реальной аппаратной реализации, например, одновременное выполнение нескольких, изолированных друг от друга, операционных систем и приложений на одном компьютере. Совокупность компьютерных ресурсов, эмулирующую работу отдельных компонентов аппаратного или программного обеспечения, или даже целого компьютера, принято называть виртуальной машиной. Наличие нескольких виртуальных чего? на одном реальном компьютере обеспечивает возможность независимой работы на одном физическом сервере (узле) нескольких операционных систем и приложений.

Виртуализация может улучшить адаптивность, гибкость и масштабируемость ИТ-среды и существенно снизить расходы. Виртуализация позволяет более эффективно использовать вычислительные мощности и совместно использовать ресурсы различных аппаратных устройств при обслуживании многопользовательских клиентов. Кроме того, виртуализация ускоряет развертывание рабочих нагрузок, повышает их производительность и доступность, а также дает возможность автоматизировать многие процессы.

Можно выделить несколько вариантов использования технологии виртуализации [33, 36]:

- консолидация серверов - обеспечивает распределение аппаратных ресурсов между виртуальными машинами, обеспечивая таким образом более полную нагрузку на аппаратное обеспечение;

- создание приложений (с тестированием) - позволяет проводить тестирование программного обеспечения на разных виртуальных платформах и конфигурациях.

- использование персональных виртуальных рабочих станций - позволяет создавать виртуальные автоматизированные рабочие места для сотрудников компаний.

- создание единого рабочего пространства - обеспечивает единую программную инфраструктуру для множества виртуальных рабочих мест.

- предоставление ИТ-инфраструктуры как услуги - позволяет удобно контролировать и настраивать виртуальные машины, которые созданы на физическом сервере.

Существует две основные технологии создания серверных систем вычислений, основанные на виртуализации серверов, в соответствии с рисунком 1.1 [16].

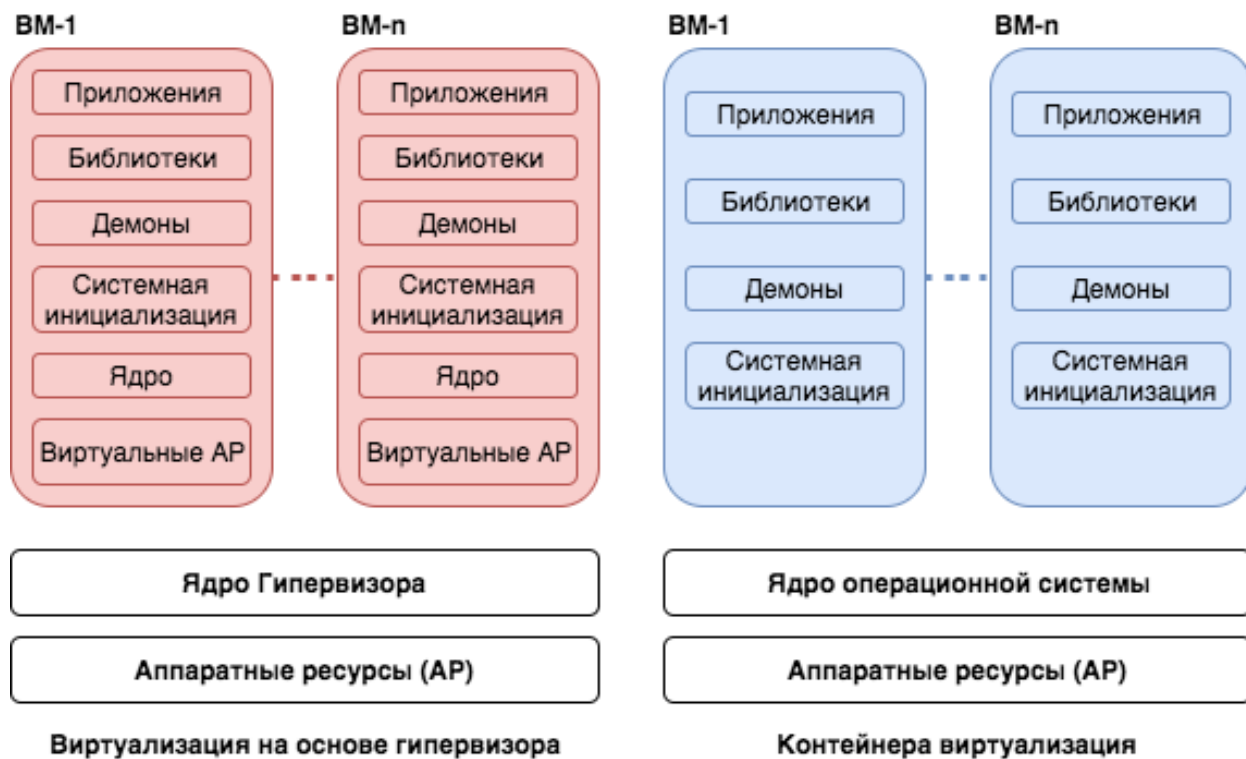


Рисунок 1.1 - Сравнение двух технологий виртуализации.

В первом подходе виртуализация осуществляется с помощью гипервизора - программной надстройки над основной ОС, которая отделяет виртуальные машины от сервера и по мере необходимости динамически выделяет вычислительные ресурсы каждой VM (Amazon, Azure, VMWare) [42]. Согласно определению облачных технологий, данному Национальным институтом стандартов и технологий США, использование данного подхода не является строгим требованием и существует второй способ, использующий изолированные контейнеры (OpenVZ, LXC (Linux Containers), Docker). В каждом из этих подходов есть как свои преимущества, так и недостатки. Подход с использованием виртуализации позволяет запускать в облаке операционные системы любых производителей, но теряя при этом в производительности от 8 до 12 процентов по сравнению с использованием физического сервера [14, 16]. Второй подход выгоднее с точки зрения вычислительной производительности системы и экономии дисковых ресурсов, так как контейнеры используют ядро основной

системы. При этом пользователи ограничены в выборе ОС только дистрибутивами семейства GNU/Linux [34], поскольку ОС Windows не поддерживают указанную технологию, что в большинстве случаев рассматривается как существенный недостаток контейнерной виртуализации. В тоже время, существенный выигрыш в производительности позволяет в этом случае использовать ресурсы аппаратного обеспечения виртуального сервера даже для высокопроизводительных вычислений. В последние годы и такие крупные игроки на рынке облачных услуг, как Amazon и Azure, помимо традиционной виртуализации на основе гипервизоров стали предоставлять услуги на основе контейнерных технологий [13]. Google использовали данную технологию изначально как основную.

Более подробно рассмотрим технологию «гипервизор». Гипервизор представляет собой специализированную операционную систему, которая функционирует исключительно на аппаратном обеспечении.

Далее уточним, что гипервизор (англ. - hypervisor) в большинстве случаев применяется в двух понятиях, которые важно отличать друг от друга. В первом случае он объединяет в себе все виды технологий, которые предоставляют поддержку работы компьютеров. По своей сути это вариант, который основан на том, что отсутствует хостовая ОС [3].

Гипервизор способствует созданию абстракции нижерасположенной аппаратной платформы, так, чтобы в дальнейшем она могла эксплуатироваться одним либо несколькими компьютерными программами, которые представляют собой имитацию оборудования для ОС. Важно отметить тот факт, что в данном случае, эти программы не знают, что перекрестно пользуются одним и тем же сетевым ресурсом и платформой. Таким образом, виртуальная машина является обычным хранилищем для операционной системы и приложений.

Достоинство данного подхода заключается в том, что виртуальная машина находится в изоляции от других виртуальных машин, которые запущены на этом

же гипервизоре, что в результате дает возможность иметь несколько операционных систем или ряд конфигураций одной операционной системы.

Гипервизор отвечает за изоляцию операционных систем (далее ОС), разделение ресурсов среди ОС, управление ресурсами, предоставление защиты и безопасности ОС [31].

Для операционных систем, которые функционируют под его управлением, виртуализацию и эмуляцию реально необходимого аппаратного обеспечения, позволяет управлять виртуальными ОС, а также выделяет и находит ресурсы для них, предоставляет возможности для самостоятельного их запуска, перезагрузки и остановки. [20].

По сути, гипервизор подразумевает под собой целый пласт ПО, на котором лежат полномочия за процесс работы виртуальных машин, а также он отделяет данную категорию продуктов от прочих элементов системы виртуализации.

Изучив типы гипервизоров, можно дать определение следующим видам виртуализации:

- виртуализация уровня операционной системы и различные виды эмуляции;
- паравиртуализация;
- полная виртуализация.

Также важно отметить то, что гипервизор, как правило, выступает в виде программы или аппаратной схемы, которые отвечают за единовременное, параллельное функционирование определенного числа ОС, на одном и том же хостовом компьютере.

Гипервизоры подразделяются на автономные (их основание это ОС) и на гибридные [2].

Еще одним способом организации серверной инфраструктуры, принципиально отличным от гипервизора, является кластер. Кластер представляет собой одну из видов параллельной или распределенной системы, состоящей из

взаимосвязанных компьютеров (серверов) и применяется в виде одного целого, стандартизированного вычислительного ресурса (функционирует в виде одной системы; пользователи также видят его как одну единую систему; управление им также в виде одной системы) [37].

Кластер включает в себя узлы, которые по своей сути представляют собой работоспособные компьютеры, объединенные между собой сетью для того, чтобы выполнять обмен данными. При этом данные могут быть разными и не однотипными, а система в них не обязательно будет гетерогенной, то есть система кластера объединяет в себе компьютеры совершенно разной структуры.

Кластер разделяется по территориальному признаку на сосредоточенный и распределённый. Формирование распределённых кластеров обуславливается развитием интернета. Во время формирования кластеров выделяются два способа [40]:

1. Система кластера включает в себя все доступные компьютеры, при этом эти компьютеры могут работать и сами по себе.

2. Система кластеров намерено объединяет промышленно выпускаемые компьютеры. При этом формируется серьезный вычислительный ресурс. Данный способ формирования кластера предоставляет возможности сделать менее затратной саму кластерную систему, поскольку в данном случае не нужно обеспечивать каждый отдельный узел монитором, клавиатурой и прочими необходимыми устройствами.

Во время применения второго подхода узлы кластера находятся в специальных стойках, а для управления им и мониторинга его состояния предоставляются один, два или более полностью работоспособных компьютеров. Данные устройства именуется хост-компьютерами.

Существует две модели функционирования кластера:

1. С разделяемыми дисками;
2. Без разделяемых ресурсов в первом случае диски – во втором ресурсы.

В модели с разделяемыми дисками каждый из них наделен физическим соединением с каждым сервером. Любой сервер обладает доступом к информации любого диска. Если с двух серверов делать запрос на чтение информации в одно и то же время, информация может читаться дважды либо запрашиваться у одного из серверов [17, 18].

Когда совершается в одно и то же время запрос на запись данных появляется конфликт. Для того, чтобы разрешить этот конфликт кластерная операционная система с разделяемыми дисками включает в себе обязательный компонент, представляющий из себя распределенный механизм блокировок.

Механизмы блокировок часто используются во всех СУБД, тем не менее, в подобной ситуации конфликты появляются среди нескольких приложений (в данный момент это база данных), которые осуществляются на разных серверах. В связи с этим механизм блокировок обязан быть одним целым для всей системы [15].

В модели, где отсутствуют разделяемые ресурсы диск может физически соединяться с сервером. В независимости от того, когда происходит процесс соединения каждому серверу доступна информация исключительно своей группы дисков. Также отсутствуют какие-либо конфликты в независимости от того, происходит ли чтение или запись.

Со стороны технической выполнения отличия моделей не настолько значительны: обе модели нуждаются в разделяемой дисковой системе. Их отличия обуславливаются только операционной системой [15, 37].

Модель, которая имеет разделяемые диски используется тогда, где основная необходимость кластера выражается в том, чтобы обеспечить высокую доступность и масштабируемость какого-либо определенного приложения (как правило, СУБД).

Именно такая модель, в частности, применена в СУБД Oracle Parallel Server. На разных серверах кластера, прежде всего, установлены отдельные элементы от

общей базы данных, а не копии какой-либо определенной базы данных. Если будет требоваться высокая производительность базы данных серьезных размеров, то тогда в систему будет добавлен дополнительный сервер со своей собственной дисковой системой. Надо сказать, что на самой дисковой системе добавленного сервера установлен элемент от общей базы данных. Тем не менее, пользователи этого не увидят, так как они видят лишь один сервер базы данных и одну базу данных. Чтобы прийти к подобному результату в модели, которая не содержит разделяемые ресурсы, нужно приложить много усилий [44].

Модель без разделяемых ресурсов используется в кластерах, где главное их предназначение заключается в предоставлении постоянной доступности спектра сетевых служб. Такая модель применяется в Microsoft Cluster Server (кодовое название Wolfpack). В обычном режиме на каждом сервере в основном происходит свой собственный набор приложений. Все приложения, которые осуществляются на одном сервере, не нуждаются в доступе к сведениям иного сервера [45, 46]. Такой подход предоставляет возможность использовать модель без разделяемых ресурсов. Модель без разделяемых ресурсов в данном случае обеспечивает большую производительность, так как каждый сервер работает только со своими дисками.

1.2 Понятие и значение методов виртуализации

Виртуализация вычислительной инфраструктуры – подход, позволяющий сформировать и использовать управляемую, высоконадежную, защищенную и наиболее продуктивную информационную инфраструктуру, которая может гибко и быстрым образом настраиваться на все возможные перемены в бизнесе. Виртуализация отвечает за процесс, который изолирует вычислительные ресурсы друг от друга, и который позволяет снизить зависимость между ними, что в результате приводит к упрощению управления переменными в системе посредством

их локализации в каком-либо определенном слое изолированных через виртуализацию ресурсов [42].

Под виртуализацией понимают технологию, которая позволяет, используя более современные, мощные и менее энергоемкие серверы объединять приложения с платформ и аппаратных средств предыдущих поколений. С каждым годом сферы использования этой технологии расширяются. Примером может служить виртуализация хранения, когда создаются базы данных, состоящие из нескольких уровней. Или виртуализация клиентских мест, когда пользователь может получить рабочие материалы, используя любой терминал, в том числе территориально удаленный [18].

Существует тесная взаимосвязь между аппаратным обеспечением, операционной системой и программами. Их взаимозависимость в физической среде весьма выраженная, в то время как в виртуальной среде эта зависимость минимальна и существует логическая изолированность. Разберем на примере принципа «машинной виртуализации». Это ситуация, когда на одном компьютере, благодаря установке специального программного комплекса (монитора виртуальных машин) разделяется оперативная память, процессор и устройства ввода-вывода между виртуальными машинами. И создаются условия, при которых одна машина работает как несколько. Все данные на них разделены, установлены различные программы, операционные системы [9, 27].

Виртуальные машины обеспечивают широкий спектр возможностей. С информацией на ней можно осуществлять любые операции, создавать резервные копии, машину можно останавливать и перезапускать, получать данные о состоянии жесткого диска и оперативной памяти. Существуют гибкие возможности переноса виртуальной машины с одного компьютера на другой. Эти возможности сохраняются даже если нет совпадения аппаратных конфигураций систем, на которые переносится виртуальная машина. Это достигается благодаря универсальности хост-систем всех виртуальных машин. Виртуализация

приложений и представлений (службы терминалов) также являются примерами виртуализации [33].

Большие возможности предоставляет серверная виртуализация. На хост-серверах создаются виртуальные машины. Это позволяет перераспределить нагрузки между серверами [35]. Преимущества, которые предоставляет этот метод:

1. Сохранение или увеличение функциональности и надежности серверов. При этом не нужно поддерживать целый парк серверов. Администрирование становится гораздо проще. Предоставляется возможность экономии на аренде помещений, снижаются затраты на охлаждение и электропитание.

2. Установка виртуальных серверов предоставляет возможность использовать хост-сервер в полной мере, так как зачастую физические серверы являются загруженными недостаточно. Это делает работу более эффективной.

3. ИТ-сервисы становятся более доступными. Это достигается благодаря перераспределению нагрузок между хост-серверами. Не возникает ситуаций, при которых один сервер недогружен, а другой перегружен. Нагрузка равномерная. Это устраняет запланированные и незапланированные простои системы.

4. Виртуальные машины имеют решения, которые базируются на оборудовании и операционных системах предыдущего поколения. Производители уже не поддерживают это оборудование. Но его можно эффективно использовать, применяя серверную виртуализацию.

Виртуализация серверов является новейшим и главным технологическим трендом в центрах обработки данных. Несмотря на ограничения, касающиеся виртуализации серверов, она дает огромный положительный результат в отношении расходов предприятий на электроэнергию и охлаждение, а также емкости и мощности центров обработки данных. Она позволяет продлить жизнь стареющим центрам обработки данных и даже дает некоторым крупным

компаниям возможность закрыть некоторые из имеющихся центров обработки данных [21].

Технологии второго поколения виртуализации, позволяющие нескольким виртуальным серверам совместно использовать одну лицензию операционной, создают возможности для дальнейшей консолидации. Она позволяет строить узлы, где в одной физической стойке работает по 50-100 виртуальных серверов, обслуживающих приложения, создающие высокую вычислительную нагрузку и плохо подходящие для использования с технологиями виртуализации первого поколения [19, 27].

Виртуализация позволяет освободить приложения от ограничений единственного физического сервера. Она позволяет запускать несколько приложений на одном сервере и дает каждому приложению возможность использовать ресурсы всей корпоративной сети.

Одним из серьезных преимуществ виртуализации является возможность динамического перемещения приложений с одного физического сервера на другой для обеспечения доступности ресурсов и непрерывности обслуживания. Виртуализация также предполагает обязательное использование сетей хранения данных (SAN) и сетевых хранилищ. Это делает сети основным элементом инфраструктуры и архитектуры ИТ [8].

С точки зрения безопасности, сосредоточение нескольких приложений на одном устройстве создает единую точку отказа, как на самом физическом сервере, так и на его сетевом подключении. В случае сбоя виртуального сервера, падения скорости или отказа его сетевого подключения, это повлияет на все приложения данной стойки [1].

Последствия этого при планировании сети проявляются в том, что на нескольких крупных серверах намного больше трафика, будет сосредоточено вместо его распределения по большому числу небольших компьютеров, размещенных в центре обработки данных. Кроме того, виртуализация лучше всего

работает с сетевыми, а не физически подключенными накопителями. Это требует очень быстрого и надежного сетевого соединения между серверами, и накопителями, размещенными на сети хранения данных (SAN). Отказ от применения физически подключаемых накопителей, происходящий в рамках перехода к виртуализации, приводит к значительному росту сетевого трафика. Все это увеличивает потребность в мощной системе управления сетью [9, 30].

Жизнеспособность среды с высокой степенью виртуализации зависит от эффективности и надежности сети передачи данных. Отказ физического сервера, соединений, коммутатора или маршрутизатора может оказаться очень дорогостоящим, если приведет к отключению рабочих мест, зависимых от работоспособности ИТ-инфраструктуры [25].

Создание серверной виртуализации имеет ряд отличительных особенностей. Более эффективно используются ресурсы, происходит интеграция ресурсов, информационные системы становятся более доступными и появляются возможности масштабирования, снижается стоимость обслуживания, оно становится проще, дешевле и эффективнее.

Виртуальными машинами несложно управлять, между ними легко перераспределять ресурсы. Это позволяет владельцам центров обработки данных оказывать услуги по предоставлению вычислительных ресурсов и ресурсов хранения. Предприниматели могут не иметь собственного ЦОД. Они пользуются внешними сервисами при обработке данных. Это приводит к экономии, ведь нужно платить лишь за потребленную вычислительную мощность [31].

Технологии виртуализации серверных ресурсов появились на рынке как средства, позволяющие использовать одну физическую вычислительную платформу в качестве основы для организации нескольких виртуальных платформ (виртуальных серверов). Для адекватной работы виртуальных серверов не имеет значение тип операционной системы, которая будет им управлять. На физическом

сервере рационально распределяются ресурсы, они используются приблизительно на 30% [30, 35]. Один сервер при классической схеме, обслуживает одну задачу.

Имеются преимущества и при размещении на виртуальных серверах приложений. Эти преимущества реализуются не только за счет рационального перераспределения ресурсов, но и за счет того, что более приоритетные задачи выполняются в первую очередь, а менее приоритетные позже.

Построение виртуальной вычислительной платформы, как было указано выше, позволяет консолидировать ресурсы, повышая тем самым степень контроля ИТ-сервисов, улучшая характеристики управляемости и, как следствие, уменьшая количество сбоев в работе систем за счёт более оперативного реагирования на первые признаки некорректной работы [30].

Бизнес и информатизация постоянно растут и развиваются. И при таком темпе роста физические платформы не справляются с объемами обработки данных. А виртуализация позволяет масштабировать ресурсы. Их проще добавлять к физическим платформам. В этом плане наиболее эффективными являются блейд-серверы. Это серверы-лезвия, на базе которых создается виртуальная серверная платформа. ИТ-инфраструктура в целом становится совершеннее. Меньше затрачивается физического пространства на оборудование, в том числе сетевое, уменьшаются энергозатраты. Это делает блейд-серверы наиболее используемыми в данной сфере [9].

Централизованный мониторинг и управление серверной платформой с помощью виртуализации делают серверы более доступными. Кроме того, создаются специальные возможности по дальнейшему увеличению доступности серверов. Преимуществом является и то, что переносить виртуальные серверы с платформы на платформу становится проще. Не нужно останавливать приложения, они могут работать в автоматическом режиме по определенным заранее критериям. Эти системы защищены от сбоев программного и аппаратного оборудования. Это выгодно отличает их от кластерных систем, при том, что

принципы работы у них аналогичные. Для предотвращения сбоев используется технология локальной защиты. Кроме того, для дополнительной защиты используются географическое кластерное разнесение. Для этого виртуальный сервер переносят на большое расстояние. При этом он дальше работает на новом сайте. Можно проводить плановые работы с аппаратным и программным оборудованием без остановки работы ИТ-сервиса [7].

Консолидация ресурсов и повышение эффективности их использования снижает требования к помещениям и инженерной инфраструктуре. Важным дополнением к этим технологиям являются дополнительные средства, позволяющие снизить затраты на инженерную инфраструктуру. Бывают периоды времени с невысокой нагрузкой на серверы, ресурсы используются мало. В эти периоды происходит перегруппировка серверов. Они распределяются так, чтобы занимать наименьшее количество физического оборудования. То оборудование, которое не используется, выходит из эксплуатации и даже отключается. И наоборот, когда повышается потребность в ресурсах, увеличивается число используемого оборудования до полного закрытия потребностей [4].

При построении ИТ-инфраструктуры компании, создаются отдельные платформы отдельно для каждого жизненного цикла приложения. Поэтому для стадий разработки, тестирования или продуктивного использования создаются и работают разные платформы. Особенностью является то, что нужна архитектурная схожесть платформ. Это позволит уменьшить количество ошибок и сбоев. И виртуальные платформы подходят для этого наилучшим образом. Они сходны по архитектуре, ресурсы между ними легко перераспределяются [3].

Консолидация всех серверных платформ предприятия в рамках одной виртуальной среды дает возможность строить систему мониторинга и управления на базе средств, обеспечивающих единый механизм взаимодействия с вычислительной средой. В дополнение к функционалу классических средств управления, средства управления виртуальным окружением позволяют

перераспределить ресурсы в соответствии с текущими потребностями систем, автоматизировать процессы, оптимизировать ресурсы и гарантировать высокую доступность виртуальной ИТ-инфраструктуры. Это обеспечивает ИТ-средам уникальную простоту эксплуатации, эффективности и надежности. Как правило, централизованные средства управления и мониторинга виртуальной среды включают обширный набор программных интерфейсов веб-служб, обеспечивающих интеграцию с продуктами для управления инфраструктурой других производителей, а также разработку специализированных решений. Положительными результатами использования виртуализации являются:

- снижение сервисных расходов, которые обычно направлены на поддержание программного и аппаратного оборудования.
- нивелирование ошибок, обусловленных человеческим фактором.
- улучшение качества работы, за счет снижения совокупных затрат [14].

Другими популярными методами виртуализации являются виртуализация систем хранения данных, передачи данных, создание виртуальных рабочих мест. Для этого существует концепция централизованной обработки данных - Server Base Computing.

Это помогает создать наиболее удобную для использования систему, которая реагирует на любые изменения потребностей бизнеса, изменяется в зависимости от требований, надежно и бесперебойно работает и доступна для пользователя [18].

Уровень операционных систем и уровень аппаратного обеспечения отделены друг от друга благодаря специально разработанным механизмам. Благодаря этому осуществляется гибкое распределение вычислительных ресурсов и ресурсов хранения данных между операционными системами, позволяют организовывать взаимодействие с вычислительными центрами по сервисной модели.

Это качественно новый подход к организации обработки пользовательских данных при котором ресурсы центра обработки данных предоставляются пользователям как сервис. Такой подход называется - «облачные вычисления» (cloud computing). В этом случае пользователь не вкладывает средства в организацию центра обработки данных как такового, не заботится о поддержании инфраструктуры центра обработки данных, работоспособности программного и аппаратного обеспечения, пользователь оплачивает лишь услугу по обработке своих данных с заданными эксплуатационными показателями. Организация обработки данных путём облачных вычислений позволяет контролировать издержки в максимальной степени [32].

Допускается также интегрировать классический подход и облачные вычисления в единую информационную инфраструктуру предприятия и применять тот или иной подход к обеспечению различных ИТ-сервисов в зависимости от требований бизнеса [25, 35]. Единые механизмы виртуализации ИТ-инфраструктуры классического центра обработки данных и ИТ-инфраструктуры облака, а также разработанные интерфейсы взаимодействия со смежными системами позволяют гибко мигрировать ИТ-сервисы между внешним облаком и центром обработки данных. Это создает необходимое равновесие: затраты на содержание ИТ-инфраструктуры фирмы против абсолютного контроля за теми сервисами, которые играют ключевую роль в работе компании.

1.3 Исторические аспекты развития технологий серверной виртуализации

Виртуализация - далеко не новая технология в настоящее время. Первые попытки применить ее осуществлялись в 1960-х гг. И разработчиком этих концепций являлась компания IBM. Они пробовали использовать эту технологию на комплексах мэйнфрейм. Это было необходимо для работы «больших ЭВМ». Эти машины редко встречались, были дорогими и нуждались в технологиях,

позволяющих обеспечить слаженную работу всего оборудования ЭВМ. Как результат - машины стали выполнять больше задач одновременно, совместно могли функционировать сразу несколько приложений. Это позволило оправдать затраты на такие дорогие машины.[3].

Ресурсы, которые могли использовать эти машины, были недостаточными. Кроме того, гибкость процессов также была минимальна. Эта ситуация сохранялась и в архитектуре x86.

Мэйнфреймы активно использовались вплоть до конца 80-х и начала 90-х. Затем появились более дешевые и меньшие по размеру серверы x86, что снизило популярность системы виртуализации. Новый виток развития она получила в 1999 г. Это стало возможным благодаря внедрению разработок компании VMware. Основываясь на архитектуре x86, ей удалось создать программное обеспечение VMware Workstation, с функцией виртуализации. Инновацией стало то, что те ресурсы, которые не использовались в данный момент, временно переставали быть активными. В работе систему возникало меньше проблем. Следующей их разработкой стало ESX Server, который работает без операционной системы хоста [21].

Происходит переосмысление использования данной технологии. Ее по-прежнему используют при тестировании и разработки программ. Но теперь она становится центральным звеном в центрах обработки данных. Такие центры более гибкие, при их работе возникает меньше отказов, при их работе централизованно создаются резервные копии. Виртуализация позволяет компаниям стать более гибкими, лучше реагировать на меняющиеся условия рынка, их система ИТ становится более безопасной и эффективной [9, 11].

Такая эволюция была необходима. Ведь для оправдания затрат на использование дорогих и сложных ресурсов необходима была реальная выгода. Все пользователи должны были удовлетворить свои потребности. Кроме того, все пользователи одной системы должны были иметь возможность одновременного ее

использования. Это и привело к созданию виртуальной машины. Каждый пользователь получал логически обособленную среду, создаваемую благодаря технологии разделения вычислительных мощностей. Каждый пользователь отделен от другого и является единоличным пользователем. Это привело к повышению эффективности мейнфрейма, повышению материальной выгоды от его использования [27].

В 1980-х гг. появилась и закрепились архитектура «клиент-сервер», которая стала вытеснять виртуализацию. Этому способствовало распространение процессорной архитектуры x86. Эта платформа, не предназначенная для виртуализации, позволила выпускать более дешевые и доступные компьютеры и серверы [21].

Дальнейшее развитие вычислительных машин, улучшение их производительности и технологической сложности, снова вернуло проблему нерационального использования ресурсов на первый план. И технология виртуализации представляла в этой сфере большой интерес. Главной технологической трудностью явилось то, что архитектура x86 не совсем для этого подходила. Это преодолевалось различными методами, но результатом стало появление процессоров, которые поддерживали технологию аппаратной виртуализации. Они стали использоваться в клиентских компьютерах и серверах [25].

1.4 Практическое применение технологий виртуализации серверной инфраструктуры

Технология виртуализации позволяет наиболее рациональным способом использовать ресурсы. Это достигается тем, что нивелируется модель "одно приложение на сервер", а все стандартные ресурсы объединяются в единую систему. Для работы нужно меньше серверов. Сокращается количество

необходимого оборудования. Однотипные задачи (инициация, настройка, отслеживание и техническое обслуживание) выполняются быстрее, часто автоматически. Это облегчает работу ИТ-администраторов, снижает расход времени на решение этих задач [33]. На одном сервере можно разместить несколько несовместимых платформ, например linux и windows серверов, но только при условии наличия виртуализированной системы.

При возникновении критической ситуации виртуализированную систему проще восстановить. У нее надежная система резервного копирования, а виртуальные среды можно переносить целиком. Для планового обслуживания не нужно останавливать работу системы. Это уменьшает периоды бездействий. Всю машину можно скопировать на носитель, ведь жестким диском у нее является файл определенного формата и расположения. Его можно скопировать, заархивировать, сделать резервную копию. Виртуальная структура управляется централизованно и затраты времени на модерацию серверов минимальны. В такой системе нагрузка распределена равномерно, виртуальные машин могут свободно мигрировать [4].

Инфраструктура современного предприятия включает в себя различного вида информационные ресурсы. Наиболее сложная в реализации задача - обеспечение корректного взаимодействия сотрудников с ресурсами, включая быстрое развертывание, ввод в эксплуатацию, масштабирование и миграцию информационных систем. Вместе с тем данная задача приобретает особую важность с учетом множества граничных условий, связанных со стоимостью аппаратного и программного обеспечения, сложностью организации сетевой связи, ограничений на физическое размещение оборудования и т. д. Дополнительно, специфика работы организации предполагает возможность разделения прав доступа при эксплуатации информационных ресурсов, возможность получения индивидуального доступа к ресурсам, их создания, модификации и т. п. С учетом недостаточного уровня профессиональной

подготовки новых работников поддержка информационной системы предприятия в консистентном состоянии становится нетривиальной задачей для сетевых и системных администраторов. Таким образом, можно конкретизировать задачи, стоящие перед руководством организации в области технологического совершенствования информационной среды [9, 30]:

- быстрое развертывание и ввод в эксплуатацию новых программных систем, быстрое свертывание, вывод из эксплуатации и замена программных систем;
- быстрая миграция программных систем с одного физического узла на другой;
- возможность масштабирования программных систем, модификация количества аппаратных и программных ресурсов, предоставляемых в их пользование;
- быстрое развертывание и ввод в эксплуатацию личных кабинетов сотрудников с интеграцией в инфраструктуру компании;
- централизованный контроль за состоянием информационной системы;
- централизованный контроль за содержанием и содержанием системы;
- минимизация расходов на приобретение и ввод в эксплуатацию аппаратного и программного обеспечения;
- минимизация расходов на выполнение всех вышеперечисленных работ.

Внедрение технологий виртуализации является одним из механизмов, который позволит обеспечить решение поставленной совокупности задач. Технологии виртуализации позволяют обеспечить быстрый ввод в эксплуатацию информационных ресурсов, их масштабирование, миграцию между физическими узлами, изоляцию окружения, консолидацию ресурсов. С их помощью могут быть быстро развернуты, а в последствии оперативно свернуты эксплуатационные и экспериментальные площадки, без необходимости ввода в строй новых

аппаратных ресурсов и с минимальным влиянием на остальные части информационной системы компании [1, 30].

Система виртуализации на предприятии позволяет обеспечить логическую изоляцию друг от друга множества наборов процессов и ресурсов в рамках одного или нескольких физических ресурсов. В прикладном понимании это означает возможность одновременного запуска нескольких программных систем, полностью изолированных друг от друга, на одной аппаратной системе [11].

Технология виртуальных машин на предприятии обеспечивает эмуляцию аппаратных ресурсов в рамках программной виртуализации или предоставление изолированного доступа к аппаратным ресурсам в рамках аппаратной виртуализации для полностью изолированных друг от друга операционных систем. Хостовая машина, обеспечивающая виртуализацию гостевых окружений, реализует данную задачу с помощью гипервизора, расположенного в ПЗУ или ППЗУ, либо с помощью дополнительного ПО для операционной системы, запущенной на хостовой машине. Данная технология позволяет запускать на одном физическом узле множество виртуальных машин с различными операционными системами в полноценном, с точки зрения гостевой системы, аппаратном окружении [8].

Среди достоинств, специфичных именно для виртуальной машины на предприятии, следует выделить:

- полный контроль администратора гостевой машины над ее ПО;
- отсутствие ограничений на тип ОС и другого ПО;
- высокая производительность гостевых машин в случае использования аппаратной виртуализации;
- высокий уровень стандартизации образов виртуальных машин, что позволяет использовать гостевые машины на гипервизорах различных производителей.

Среди недостатков использования виртуальных машин в рамках деятельности предприятия следует обратить внимание на:

- низкую производительность гостевых машин в случае использования программной виртуализации;
- относительно высокую стоимость физических узлов с поддержкой аппаратной виртуализации;
- низкий уровень контроля администратора хостовых машин над процессами, происходящими в гостевых машинах и их содержимым;
- относительно низкая скорость развертывания новых гостевых машин;
- высокие затраты внешней памяти на создание архивных копий.

На сегодняшний день предприятие может использовать достаточно большой набор программных продуктов, обеспечивающих поддержку виртуальных машин с программной или аппаратной виртуализацией, наиболее известными из которых являются: KVM (аппаратная), Qemu (программная и аппаратная), Hyper-V (аппаратная), VirtualBox (программная и аппаратная), VMware (программная и аппаратная), Xen (аппаратная и паравиртуализация) и многие другие [31, 34].

Вторая из рассматриваемых технологий виртуализации операционных систем (ТВОС), позволяет обеспечить запуск на одном и том же ядре ОС в рамках хостовой машины множество полностью изолированных друг от друга гостевых машин. Изоляция обеспечивается только на уровне гостевых машин, с точки зрения администратора хостовой машины все процессы гостевых машин исполняются в адресном пространстве хостовой и полностью доступны администратору. Пространство внешней памяти (жесткого диска) для гостевых машин обеспечивается в отдельном каталоге файловой системы хостовой машины, что означает перманентную доступность файлов гостевых машин администратору хостовой машины. Во многих реализациях данной технологии существует возможность передавать отдельные устройства в эксклюзивное пользование от хостовой машины гостевой.

Среди достоинств, специфичных для данной технологии, следует выделить [32, 35]:

- полный контроль администратора хостовой машины над ресурсами и процессами всех гостевых машин;
- максимальный уровень производительности среди всех технологий виртуализации;
- отсутствие требований к наличию аппаратной виртуализации и, как следствие, удешевление аппаратного обеспечения хостовой машины;
- высокая скорость и простота миграции гостевых машин с одной хостовой машины на другую;
- высокая скорость ввода и вывода из эксплуатации гостевых машин;
- минимальные затраты внешней памяти на архивацию, простота реализации инкрементной архивации.
- Среди недостатков следует обратить внимание на:
 - невозможность запуска в гостевой машине операционной системы или ядра ОС, отличного от ОС/ядра хостовой системы;
 - ограниченность доступа к аппаратным устройствам;
 - несовместимость контента гостевых машин в различных технологиях виртуализации уровня ОС.

ТВОС специфична для каждой конкретной операционной системы. На сегодняшний день существуют и активно развиваются следующие реализации данной технологии: OpenVZ (ОС Linux), Virtuozzo (ОС Linux, ОС Windows), Solaris Containers (ОС Solaris), FreeBSD Jail (ОС FreeBSD) и некоторые другие [33, 35].

Ниже приведены несколько наиболее распространенных сценариев использования серверной виртуализации, которые можно использовать в рамках функционирования информационных технологий предприятия [27, 31, 35]:

– Консолидация. С помощью консолидации предприятия могут быстро и просто снизить операционные затраты на ИТ: серверная виртуализация помогает уменьшить количество подлежащих техническому обслуживанию физических систем и таким образом упростить ИТ-инфраструктуру. Компания может извлечь значительную экономическую выгоду из консолидации как большого центра обработки данных, так и небольших серверных ферм, используемых, например, в филиалах.

– Тестирование и разработка. Серверная виртуализация значительно облегчает процесс тестирования программного обеспечения. Администраторы могут быстро развернуть новые операционные системы, приложения и целые решения в виртуальной среде для целей тестирования. Причем это не потребует приобретения дополнительного оборудования или использования рабочей инфраструктуры предприятия с риском ее повреждения. Разработчики также получают возможность провести всестороннее тестирование создаваемых программных продуктов с возможностью вернуться к любому сохраненному ранее состоянию среды тестирования.

– Непрерывность ИТ-сервисов. При использовании серверной виртуализации влияние запланированных и незапланированных простоев физических серверов значительно меньше влияет на доступность ИТ-сервисов.

– В случае сбоя физического сервера можно оперативно ввести функционирующие на нем виртуальные машины в работу на другом сервере, что сводит к минимуму обусловленные сбоями перерывы в работе ИТ.

– Динамический центр обработки данных. Виртуализация сервера является важным шагом в обеспечении адаптивности ИТ-инфраструктуры к меняющимся потребностям бизнеса. Благодаря возможности переноса виртуальных машин между хост-серверами, может осуществляться динамическая балансировка загрузки существующих в центре обработки данных вычислительных ресурсов. Инструменты для сквозного управления

инфраструктурой позволяют контролировать состояние систем не только по простым параметрам загрузки физических серверов, таким как использованная оперативная память и процессорное время, но и по специфическим, важным для конкретных полезных нагрузок и бизнес-приложений характеристикам. Таким образом достигается способность адаптации инфраструктуры, исходя из текущих потребностей бизнеса.

Выводы по 1 главе

В результате выполнения первой главы работы были рассмотрены современные способы организации серверной инфраструктуры, определено понятие и значение технологии виртуализации, изучены исторические аспекты развития технологий серверной виртуализации и особенности практического применения технологий виртуализации серверной инфраструктуры. Виртуализация – это технология, которая дает возможность объединить приложения на различных платформах и аппаратных средствах предыдущих поколений с использованием меньшего числа современных, более мощных серверов с низким энергопотреблением. Виртуализация серверов является новейшим и главным технологическим трендом в центрах обработки данных. Несмотря на ограничения, касающиеся виртуализации серверов, она дает огромный положительный результат в отношении расходов предприятий на электроэнергию и охлаждение, а также емкости и мощности центров обработки данных.

Глава 2 АНАЛИЗ СОВРЕМЕННЫХ РЕШЕНИЙ ВИРТУАЛИЗАЦИИ СЕРВЕРНОЙ ИНФРАСТРУКТУРЫ

2.1 Программное и аппаратное обеспечение в современных решениях серверной виртуализации

Серверная виртуализация представляет собой полную эмуляцию оборудования на уровне программного обеспечения. Виртуальное окружение для виртуальной машины представляется операционной системе как аппаратное, виртуальная машина не может определить подключен эмулятор оборудования, или реальное оборудование. В этом состоят особенности программного и аппаратного обеспечения виртуальных машин [6].

В 60-е годы 20 века инженеры IBM изготовили и выпустили первые виртуальные машины. Они работали на платформах System/360 и System /370. В этих машинах процессы строго разграничивались и осуществлялись каждый на своем уровне в соответствии с разрешенным или запрещенным перечнем инструкций и привилегий. Программы, которые работали на более высоких уровнях осуществляли контроль над программами с более низким уровнем привилегий. И если такие программы пытались выполнить задачу не своего уровня, что это действие исключалось [1, 4].

Классическая схема виртуализации называлась «Trap and Emulate», что переводится лови и эмулируй. Именно ее разработали инженеры IBM. В ней существовал гипервизор или монитор виртуальных машин. Эта программа имела наивысший уровень привилегий. А основная операционная система открывалась на одном из низших уровней [6].

Работа организовывалась следующим образом. Гостевая ОС обрабатывает непривилегированные инструкции безо всяких ограничений, ее обработкой занимается аппаратура ЭВМ. Если же она пытается выполнить

привилегированную инструкцию, то генерируется программное исключение, которое захватывает гипервизор. Он эмулирует выполнение инструкции для данной вычислительной машины [3, 6].

Гипервизор помогает эффективно взаимодействовать нескольким виртуальным машинам внутри одного физического устройства, поделить гостевым системам физические устройства. Он анализирует задачи, которые выполняет виртуальная машина и либо предоставляет ей доступ к оборудованию, либо эмулирует доступ к оборудованию. Те есть является как бы диспетчером, связующим звеном в работе нескольких VM.

В период появления первых средств виртуализации главная функция виртуальной машины заключалась в персонализации рабочего стола пользователя. На современном этапе платформы виртуализации применяются для решения совершенно иных задач, от организации виртуальных рабочих мест до тестирования программных продуктов. Серверную виртуализацию целесообразно использовать в том случае, когда реализуется приложение, для работы которого необходимо наличие нескольких платформ, или несовместимых конфигураций. Трафик и загруженность этого приложения могут быть и невелики [9].

Конечно, главными пользователями виртуальных машин являются разработчики приложений, системные интеграторы и администраторы сетей. Но эта технология доказала свою значимость и для бизнес-пользователей. Виртуальные машины могут использоваться при запуске приложений, которые работают вне рабочей станции пользователя.

С их помощью можно тестировать операционные системы или сетевые службы. Проводить демонстрацию продуктов, размещенных на разных платформах, осуществлять их взаимодействие внутри одного аппаратного оборудования. Осуществлять техническую поддержку продуктов с нескольких платформ. Они позволяют подготовить учебные классы, скопировав на компьютеры настроенное ПО.

Представленными вариантами возможности применения виртуальных машин не ограничены [8].

Каждая виртуальная машина включает определенную архитектуру, которая включает ряд взаимодействующих основных компонентов:

- приложение виртуальной машины – приложение с графическим интерфейсом, которое выполняется под управлением основной операционной системы;

- драйвер виртуальной машины – шлюз между приложением и монитором виртуальной машины;

- монитор виртуальной машины – управляющий элемент виртуальной машины.

Приложение виртуальной машины имеет многопоточную технологию и может поддерживать 3 потока:

- поток виртуализации для обмена информационными сообщениями посредством монитора;

- графический поток отображения буфера видео гостевой системы;

- поток пользовательского интерфейса, который обеспечивает диалог пользователя и системы.

Можно установить на один компьютер несколько хостинговых ОС и разделить нагрузку между ними. Но предпочтительно использовать виртуальные машины так как:

- можно работать с несколькими системами в одно и то же время, эти системы могут взаимодействовать между собой при желании;

- можно легко оценить текущее состояние системы, провести оценку и при необходимости, быстро вернуться в первоначальное состояние;

- можно легко создавать резервные копии;

- количество виртуальных машин на одном компьютере не ограничено.

Они могут иметь различные операционные системы;

– при переключении между операционными системами не нужно перезагружать компьютер и др.

Виртуальные машины имеют несомненные плюсы, но не лишены недостатков [4]:

– аппаратные ресурсы должны быть достаточно мощными, чтобы обеспечить работу сразу нескольких операционных систем;

– работа операционной системы внутри виртуальной машины замедляется.

Эти недостатки пытаются нивелировать. Виртуальные машины постепенно улучшаются. И производительность гостевых операционных систем также увеличивается. Она постепенно приближается физическим операционным системам. Вскоре эти показатели сравняются. Существуют методики определения в фактической или виртуальной машине запущена программа. Эти методы чаще всего демонстрируют производители виртуальных машин. Знают о них, и производители вредоносного ПО и уже включают в вирусные программы эти методы. Не все оборудование и интерфейсы поддерживают виртуализацию. Но производители платформ виртуализации постоянно расширяют свои возможности.

Виртуальные машины постоянно совершенствуются и в скором времени эти проблемы будут полностью решены. А их достоинства несомненно преобладают над недостатками. Именно этим объясняется их популярности, огромные темпы развития, а также возникновение новых сфер использования.

Следует учитывать одну особенность классической схемы виртуализации. Все инструкции процессора, влияющие на ресурсы компьютера, обязаны иметь привилегированный статус. Аппаратура не должна содержать инструкций, меняющих свое поведение при изменении состояния этих ресурсов. Все инструкции должны вести себя одинаково на разных уровнях привилегий, а если их состояния меняются, то должно включаться аппаратное исключение.

Существуют критерии Попека–Голдберга, названные в честь разработчиков систем виртуализации IBM. Они более четко определяют требования к платформам, необходимые для создания и функционирования гипервизора.

Эти простые требования часто не выполняются разработчиками реальных процессоров. Они не планируют использовать свое аппаратное средство для деятельности виртуальных машин.

Для персональных компьютеров длительно пользовались процессорами с архитектурой x86. Но виртуализация с каждым годом становится все более востребованной. Это заставило основных разработчиков процессоров Intel и AMD модернизировать свою архитектуру, сделать ее более удобной для создания гипервизоров [6].

Intel (VT-x) и AMD (AMD-V) ввели корневой (root) и некорневой (nonroot) режимы на аппаратном уровне. Это разграничивает привилегии и делает их более удобными для виртуализации. Разница для гостевых ОС не заметна, они работают также как на процессоре с поддержкой виртуализации.

Гипервизор размещается на корневом режиме. Гипервизор размещается на корневом режиме. Пока все инструкции выполняются по правилам, процессор работает в некорневом режиме, если управляющая структура замечает другую ситуацию, то процессом начинает управлять гипервизор. Он выполняет эмуляцию инструкций. Управление снова начинает работать в некорневом режиме.

Аппаратная поддержка виртуализации осуществлялась, на первых парах, благодаря изменениям свойств процессора. Производился перехват инструкций. В последующем виртуализация распространилась и на оперативную память, и дальше - виртуальные машины получили возможность управлять всеми устройствами компьютера. Гипервизоры могут контролировать все его системы.

Intel (VT-x) и AMD (AMD-V) получили названия «аппаратная виртуализация». Это определение в равной мере используют и у нас, и за рубежом. Но оно создает неправильное впечатление будто внутри процессора уже

встроен гипервизор. Это не так. Данные процессоры лучше приспособлены для виртуализации, но гипервизоров в них нет. При желании систему виртуализации можно установить. И она будет работать в соответствии с классической схемой «Trap and Emulate».

Изменения в аппаратном обеспечении освободили производителей гостевых систем от необходимости их изменений.

Аппаратная поддержка делает ненужным громоздкий и сложный код бинарных трансляторов. Этот код используется для оптимизаций и его главная цель - обеспечить высокую производительность.

Те решения, что предложили Intel и AMD имеют огромные перспективы. Их можно использовать не только для поддержки виртуализации. Благодаря им компьютер может запускать программы с более высоким уровнем привилегий, чем на установленной операционной системе. Они обладают способностями останавливать работу системы, изменять адресное пространство и вновь возвращать управление операционной системе, при возникновении определенных ситуаций [7].

Необходимы дополнительные исследования в этой области. Особенно для обеспечения информационной безопасности. Работая в корневом режиме ПО, уходит из-под контроля операционной системы и антивирусных программ. Для создателей вредоносного ПО это открывает новые возможности. Поэтому необходимы новые разработки и методы, защищающие программное обеспечение при работе в корневом режиме.

Исследования в этой области активно идут уже несколько лет, и есть достаточно интересные практические наработки. Более подробно проблемы защиты будут рассмотрены в п. 2.3.

Виртуализация физических ресурсов ИТ-инфраструктуры позволяет предоставлять обобщенные ресурсы на нескольких серверах (узлах гипервизоров) для создания на их основе виртуальных машин. Данный подход позволяет

одновременное выполнение нескольких ОС и приложений на одном сервере что позволяет использовать единый физический сервер, как для серверных нужд, так и для нужд виртуализации рабочих мест пользователей. Подобная консолидация серверных ресурсов позволяет устранить избыточное выделение ресурсов, повысить эффективность использования серверов, сократить расходы на оборудование и эксплуатацию. Использование виртуализированной инфраструктуры позволяет сократить время на создание нового типового сервера или рабочего места сотрудника.

Подсистема виртуализации состоит из следующих компонент:

- программного обеспечения управления виртуализации;
- программного гипервизора.

Функции подсистемы виртуализации разделены на следующие группы: – административные функции:

- операции по администрированию, управлению, масштабированию, планированию и мониторингу виртуализированных ресурсов ИТ-инфраструктуры, осуществляемые техническими специалистами Заказчика в рамках эксплуатации изделия;

- пользовательские функции – ориентированные на выполнение рутинных задач по работе с виртуализированной средой конечными пользователями.

В зависимости от роли пользователя в системе, в соответствии с утверждённой ролевой моделью, доступ к административным и пользовательским функциям осуществляется через специальный интерфейс пользователя [8].

Административные функции интерфейса администратора подсистемы виртуализации включают:

- создание виртуальной инфраструктуры;
- управление виртуальной инфраструктурой;
- управление и учёт пользователей и администраторов;

- создание отчётов по работе подсистемы и её компонентов;
- авторизация в системе каталогов;
- проведение пользователем административных мероприятий по отношению к виртуальным машинам, предоставленным ему в личное пользование, включая возможность включения/выключения, перехода в спящий режим;
- осуществление доступа пользователя по специальным протоколам доступа и проведение пользователем всех необходимых действий с виртуальной машиной, её ОС и прикладным ПО в рамках полномочий пользователя.

Кластер в виртуальной инфраструктуре изделия — это группа независимых узлов (серверов), совместно выполняющих общий набор приложений, которая воспринимается запущенными приложениями как единая система. Эти связи позволяют узлам кластера использовать методы передачи нагрузки при сбое и балансировки нагрузки, которые неприменимы к одиночному серверу.

Кластер виртуализации обеспечивает следующий ключевой функционал:

- Балансировка нагрузки на аппаратные ресурсы изделия в соответствии с заданными политиками использования ресурсов, включая возможность миграции пользовательских виртуальных машин между узлами кластера.
- Возможность консолидации всех запущенных виртуальных машин на одном узле кластера при невысокой загрузке системы с целью экономии электроэнергии и перевода незадействованных узлов в экономичный режим.
- Обеспечение отказоустойчивости. В случае выхода из строя аппаратных компонентов одного из узлов кластера, что приводит к его неработоспособности, все пользовательские виртуальные машины, работавшие на данном узле, автоматически перезапускаются на других рабочих узлах кластера в соответствии с утверждёнными политиками миграции и обеспечения отказоустойчивости.

Для работы с виртуализованными программными средами (виртуальными машинами) возможно использование ряда протоколов доступа. Протокол определяет набор протокольных сообщений для подключения, управления и получения входящей информации от удаленных компьютерных устройств (например, клавиатура, видеокарта, мышь) по сети, а также отправка им сообщений. Контролируемое устройство может располагаться на любой стороне: как на стороне сервера, так и/или на стороне клиента. Кроме того, протокол определяет набор вызовов для поддержки миграции удаленного сервера с одного сетевого адреса на другой.

Для хранения служебных баз данных системы виртуализации могут включать СУБД (PostgreSQL/MySQL/MSSQL), драйвера доступа к базам данных ODBC, ADO др. Базы данных обеспечивают хранение и оперирование данными пользователей систем виртуализации.

2.2 Платформы виртуализации серверной инфраструктуры

Виртуализация по данным аналитической компании Gartner занимает первую позицию среди ключевых трендов в ИТ-области. Существует два типа виртуализации: виртуализация ресурсов и виртуализация платформ. Технологии виртуализации платформ в настоящее время активно развиваются и прогрессируют, имеют множество различных видов реализации (полная виртуализация, нативная виртуализация, паравиртуализация, виртуализация уровня операционной системы, виртуализация уровня приложений). Перспективность виртуализации платформ определяется рядом ее достоинств, в число которых входят такие ее возможности, как: создание требуемых аппаратных конфигураций с требуемыми параметрами; создание представлений устройств, несуществующих в вычислительной системе; проведение безопасных экспериментов со старыми и новыми операционными системами на одном

физическом компьютере в целях проверки на совместимость; безопасная работа с изолированными сомнительными и подозрительными приложениями и компонентами; создание виртуальной сети из нескольких систем на одном физическом сервере; проведение безопасных экспериментов и обучения в ИТ-сфере; обеспечение высокой мобильности вне зависимости от платформ приложений, рабочих столов и т.д.; экономия аппаратного обеспечения при виртуализации серверов; лучшая управляемость виртуальных машин по сравнению с реальными.

На сегодняшний день существует несколько систем виртуализации. Рассмотрим наиболее функциональные из них: VMware ESXi Server, Citrix XenServer, Microsoft Hyper-V, Linux KVM [9].

Первопроходцем в области эффективной виртуализации на платформе x86 стала компания VMware, реализовавшая динамическую трансляцию бинарных инструкций виртуальной машины. В предложенном компанией подходе гипервизор анализирует все инструкции, которые пытается выполнить гостевая ОС. Привилегированные инструкции, а также «нехорошие» непривилегированные команды заменяются «на лету» на обращения к гипервизору. «Хорошие» непривилегированные инструкции передаются на выполнение непосредственно процессору [6].

Продукты, основанные на бинарной трансляции, выполняют эмуляцию всех необходимых устройств для каждой VM. Проблема эмуляции сложных устройств (видеокарта и др.) решается установкой в гостевую ОС дополнений (таких как VMware Tools или VirtualBox Guest Additions), включающих в себя специализированные драйверы. Эти драйверы знают о том, что ОС работает внутри VM, а не на реальной машине, и вместо попытки обращения к реальным устройствам обращаются напрямую к гипервизору. Такой подход упрощает схему работы (гипервизору не надо перехватывать и анализировать инструкции), повышает производительность и дает возможность предоставить виртуальной

машине нечто большее, чем видеокарта, поддерживающая режим VESA в разрешении 800x600. Впрочем, эти драйверы уже фактически относятся к другой технологии – паравиртуализации. Паравиртуализация основана на внесении модификаций в код гостевых ОС. Цель таких модификаций – замена инструкций, которые необходимо эмулировать, на прямые обращения к гипервизору, так что последнему не надо ничего перехватывать – ОС сама отдает ему управление в случае появления «неправильной» инструкции. Таким образом работают указанные выше специализированные драйверы в средах, использующих бинарную трансляцию, поэтому корректно их называть паравиртуализированными драйверами. Точно такие же драйверы используются и в системах с паравиртуализированным ядром.

VMware ESXi Server — это программный продукт для виртуализации уровня предприятия, предлагаемый компанией VMware. ESXi является гипервизором и ставится непосредственно «на железо», то есть при установке не требуют наличия на машине установленной операционной системы. ESXi позволяет разделить физический сервер на логические разделы, называемые виртуальными машинами. Включает в себя средства управления виртуальными ресурсами предъявляет определённый набор требований к аппаратному обеспечению: например, наличие поддержки виртуализации со стороны архитектуры процессора является обязательным [1].

Citrix XenServer – это кроссплатформенный гипервизор. Является разработкой Кембриджского университета и распространяется на условиях лицензии GPL. С самого начала поддерживались две ветки: с открытым исходным кодом и коммерческая версия. В 2007 году Citrix поглотила XenSource и уже в 2009 году объявила, что коммерческие версии XenServer станут полностью свободными, что подразумевает под собой его полную бесплатность и открытость исходных кодов. На данный момент для скачивания на сайте компании Citrix

доступна полноценная версия гипервизора. Гипервизор является платформой для создания ВМ, которые создаются и администрируются программой XenCenter [2].

Microsoft Hyper-V — система аппаратной виртуализации для x64-систем на основе гипервизора. Бета-версия Hyper-V была включена в x64-версии Windows Server 2008, а финальная версия была выпущена в 2008 году. Ранее была известна как виртуализация Windows Server (Windows Server Virtualization). Имеет широкий набор возможностей для управления ВМ. Отличительной чертой Hyper-V является его взаимосвязанность с ролью Active Directory (AD), которая, как и гипервизор, является компонентом Windows Server 2012. Hyper-V, в отличие от аналогов, не может использоваться отдельно от Windows Server [3].

Гипервизор Hyper-V в серверной операционной системе Windows Server позволяет создать универсальную платформу, повышающую эффективность работы виртуальных машин, с удобной возможностью добавления виртуальных машин и простым управлением ими. Расширенные функции, такие как создание виртуальных подсетей и встроенные средства автоматизации, позволяют размещать и перемещать виртуальные машины в любое время и в любое место [10]. Среди возможностей можно выделить следующие:

1. Масштабирование с выходом за рамки виртуальных локальных сетей. Вместо использования виртуальных локальных сетей (VLAN), которые отличаются сложностью и требуют значительных затрат времени, теперь можно воспользоваться функцией виртуализации сетей Hyper-V Network Virtualization. Эта функция позволяет изолировать сетевой трафик в разделяемой инфраструктуре и перемещать виртуальные машины в виртуальной инфраструктуре - и даже перемещать их в публичное облако - при этом сохраняя настройки их виртуальных сетей.

2. Миграция виртуальных машин и систем хранения данных без прекращения их работы. Улучшенная система миграции позволяет перемещать несколько виртуальных машин одновременно, даже за пределы среды кластера,

без прекращения их работы. Кроме того, миграция систем хранения данных не приводит к простоям, так как новое средство миграции хранилищ обеспечивает перемещение виртуальных жестких дисков, подключенных к работающим виртуальным машинам, без прекращения их работы.

3. Автоматизация рутинных задач. Полная поддержка автоматизации с помощью Windows PowerShell позволяет повысить производительность труда и тратить меньше времени на техническое обслуживание. Администратор может создать инструменты, работающие из командной строки, или автоматизированные сценарии для настройки, конфигурирования, отслеживания и выполнения прочих задач.

4. Гибкость для СХД. Hyper-V поддерживает работу с хранилищем, которое подключено напрямую к серверу (DAS), общие хранилища с доступом по протоколу SMB, подключенные через адаптер (HBA) и сети хранения данных (SAN). Как результат, становится проще управлять потребностями в хранении и сократить затраты на хранение данных.

Гипервизор Hyper-V, встроенный в Windows Server, полезен вне зависимости от того уровня развития инфраструктуры предприятия. При этом возможна виртуализация любой рабочей нагрузки. Hyper-V способен поддерживать наиболее объемные и критически важные нагрузки, предоставляя доступ к большому числу различных функций. Гипервизор Hyper-V обеспечивает решение следующих задач:

1. Масштабирование по мере необходимости. Гипервизор Hyper-V в Windows Server значительно расширяет возможности масштабирования ресурсов узла и рабочих нагрузок — до 320 логических процессоров на аппаратном уровне, до 4 ТБ физической памяти и до 1 ТБ памяти для каждой из виртуальных машин, до 64 виртуальных процессоров и до 4 000 виртуальных машин в одном кластере.

2. Улучшенная поддержка динамической памяти. Расширенная поддержка динамической памяти позволяет повысить степень консолидации и

увеличить надежность, что, в свою очередь, способствует снижению затрат, в особенности для сред со множеством недозагруженных или неактивных виртуальных машин.

3. Точное измерение потребляемых ресурсов. Средство измерения ресурсов Resource Metering в Windows Server позволяет получать более точные сведения об уровне использования ресурсов конкретными серверами. Эти данные можно использовать для любых целей, начиная с планирования мощностей и заканчивая перераспределением рабочих нагрузок. Использование расширенных возможностей хранилищ данных. Windows Server разработана с расчетом на улучшение поддержки текущих и будущих рабочих нагрузок, а также позволяет не сомневаться, что развивающиеся потребности пользователя будут удовлетворены. Этому способствует такой новый функционал, как поддержка нового формата виртуальных жестких дисков большой емкости VHDX, возможность передачи задач работы с данными на аппаратный уровень (Offloaded Data Transfer), Data Center Bridging, а также виртуальный интерфейс Fibre Channel.

4. Гарантированная пропускная способность. Служба QoS, управляющая распределением полосы пропускания, позволяет гарантировать минимальную пропускную способность виртуальной машине, службе или сетевой системе хранения данных.

Поскольку клиенты размещают в виртуализированных средах все более важные рабочие процессы, возрастает потребность в интегрированных решениях повышения доступности, как для крупных компаний, так и для небольших предприятий. Гипервизор Hyper-V в Windows Server по-прежнему обеспечивает создание лучших в своем классе высокодоступных решений, а его дополнительные функции гарантируют удовлетворение всего спектра потребностей любых предприятий, независимо от размера их бюджетов. Поддерживаемые возможности:

1. Повышение гибкости кластеризованных сред. Использование дополнительных преимуществ, которые предоставляют повышенную плотность, одновременную миграцию и виртуальный интерфейс Fibre Channel.

2. Оптимизация операций резервного копирования. Добавочное резервное копирование виртуальных жестких дисков позволяет ускорить и упростить процедуры резервного копирования, а также сэкономить полосу пропускания вычислительной сети и место на дисках.

3. Обеспечение непрерывности бизнеса. Функция Hyper-V Replica обеспечивает асинхронную репликацию виртуальных машин в различные местоположения, в том числе и географически удаленные, что позволяет добиться непрерывности бизнеса и обеспечить восстановление после сбоев.

4. Повышение устойчивости к незапланированным простоям. Windows Server обеспечивает защиту от множества незапланированных сценариев сбоев за счет поддержки широкого спектра функций, включая прозрачный механизм восстановления после сбоя для файл-серверов SMB, надежное восстановление после сбоев и объединение сетевых адаптеров.

KVM (Kernel-based Virtual Machine) — это программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86, которая поддерживает аппаратную виртуализацию на базе Intel VT (Virtualization Technology) либо AMD SVM (Secure Virtual Machine). Программное обеспечение KVM состоит из загружаемого модуля ядра, называемого `kvm.ko`, предоставляющего базовый сервис виртуализации, процессорно-специфического загружаемого модуля `kvm-amd.ko` либо `kvm-intel.ko`, и компонентов пользовательского режима (модифицированного QEMU). Все компоненты ПО KVM являются ПО с открытым исходным кодом.

Физические сервера, на которых запускаются ВМ, могут быть объединены в кластер, и в случае отказа одного из серверов автоматически «переезжать» на другой. Полной отказоустойчивости добиться не всегда возможно. К примеру, в

MS Hyper-V такое перемещение будет выглядеть так же, и иметь такие же последствия, как внезапное обесточивание сервера. Если же перемещение виртуальных машин происходит в штатном режиме, то оно может пройти незаметно для пользователей. Такие технологии у разных разработчиков называются по-разному: у Microsoft она называется «Live Migration», у VMware – «Vmotion». Эти технологии позволяют проводить работы, связанные с выключением ВМ прямо в рабочее время и не отключая пользователей от работы. Кроме того, если структура сети построена соответствующим образом, запущенные виртуальные машины могут автоматически перемещаться на менее нагруженные сервера. В инфраструктуре сети, построенной на основе технологий от Microsoft для этого используются «System Center Virtual Machine Manager» и «Operations Manager». В инфраструктуре, построенной на технологиях от VMware и Citrix, используются «Vsphere Center» и «XenCenter», соответственно.

Рассмотренные платформы виртуализации имеют одну общую главную функцию – непосредственно создание ВМ. Они имеют набор сходных функций, однако имеются и различия. Сравнительная характеристика систем виртуализации приведена в таблице 2.1.

Таблица 2.1 – Сравнение систем виртуализации

	Vmware ESXI	Citrix XenServer	Microsoft HyperV	Linux KVM
Встроенное резервное копирование ВМ	Да	Да	Нет	Нет
Живая миграция виртуальных машин	Да	Да	Да	Да
Непрерывная доступность ВМ	Да	Нет	Да	Нет
Миграция ВМ между системами хранения данных	Да	Нет	Да	Нет
Автоматическое распределение нагрузки между хостами	Да	Да	Нет	Нет
Автоматическое распределение нагрузки между СХД	Да	Нет	Нет	Нет
Распределение ресурсов процессора и памяти между ВМ	Да	Да	Да	Да
Распределение ресурсов сетевых адаптеров между ВМ	Да	Нет	Нет	Нет

Резюмируя описанное, можно выделить ряд достоинств виртуализации: снижение требований к «железу» на стороне клиентов, повышение безопасности, значительное упрощение администрирования и поддержки. К недостаткам можно отнести следующее: повышение требований к серверам, как по производительности, так и по надежности, относительная дороговизна лицензий и сервера.

2.3 Проблематика и специфика внедрения технологий виртуализации

Несмотря на достоинства наличие недостатков технологий виртуализации сдерживает их широкое признание и применение. К общепризнанным недостаткам относятся:

- невозможность создания представлений устройств, не учтенных вендорами в системах виртуализации;
- высокие требования к аппаратному обеспечению; высокая стоимость корпоративных платформ виртуализации;
- более низкое быстродействие виртуальных машин по сравнению с реальными;
- появление новых малоизученных и малоисследованных рисков и угроз безопасности информации;
- появление новых проблем, которые всегда связаны с внедрением новых технологий (например, отсутствие специалистов, необходимость изучения технологии и управления ею и т. д.).

Многие компании с настороженностью относятся к виртуализации. Это связано с несколькими факторами, среди которых существенную роль играют специфические риски и угрозы безопасности информации при виртуализации платформ, которые мало исследованы [1, 2]. Исследование рисков и угроз

безопасности информации при использовании виртуализации является актуальной проблемой.

Существует достаточное количество специфических рисков виртуализации платформ, среди которых и внутренние угрозы безопасности:

- возможность сетевых атак между виртуальными машинами (виртуальными серверами), расположенными на одном физическом сервере;

- риск компрометации гипервизора (монитора) виртуальных машин (при его наличии) и/или хостовой ОС (при ее наличии);

- нарушение основного принципа виртуализации – изоляции процессов (виртуальных машин);

- отсутствие видимости сетевого трафика между виртуальными машинами. Если в физических локальных сетях передача любого трафика между рабочими станциями возможна лишь через специальные узлы сети (маршрутизаторы, хабы, свитчи, коммутаторы), то в виртуальной среде присутствует трафик, которым рабочие станции и сервера обмениваются напрямую между собой. Внешние сетевые системы безопасности не применимы для защиты систем виртуализации, так как они «не видят» сетевого трафика, передаваемого между виртуальными машинами внутри платформы виртуализации;

- бреши в защите в момент включения виртуальных машин. Виртуальные сервера и рабочие станции гораздо чаще физических вводятся и выводятся из эксплуатации, часто хранятся выключенными в виде файлов, не редко доступ к ним предоставляется по требованию. Разумеется, что внутри выключенных виртуальных машин не устанавливаются обновления прикладного ПО и сигнатур антивируса, т. к. выключенных виртуальных машин для «традиционных» средств защиты, например антивируса -просто не существует;

- конфликт ресурсов, «антивирусный шторм». В наши дни антивирус на каждом сервере и рабочей станции уже практически не снижает

производительности самой системы. В среде виртуализации ситуация несколько иная, т. к. на одной аппаратной платформе может быть развернуто множество виртуальных серверов и рабочих станций. При одновременном сканировании хотя бы части виртуальных ресурсов производительность платформы значительно снижается. Традиционный антивирус хранит экземпляр антивирусной базы в каждой виртуальной машине и не учитывает, что защищаемые узлы размещены на одной аппаратной платформе (физическом сервере).

Кроме указанных рисков, виртуальные машины могут быть подвержены множеству угроз, среди которых:

- угроза воздействия на программы с высокими привилегиями;
- угроза выхода процесса за пределы виртуальной машины;
- угроза злоупотребления возможностями, предоставленными потребителям облачных услуг;
- угроза изменения компонентов системы;
- угроза искажения вводимой и выводимой на периферийные устройства информации;
- угроза нарушения изоляции пользовательских данных внутри виртуальной машины;
- угроза нарушения процедуры аутентификации субъектов виртуального информационного взаимодействия;
- угроза нарушения технологии обработки информации путём несанкционированного внесения изменений в образы виртуальных машин;
- угроза невозможности миграции образов виртуальных машин из-за несовместимости аппаратного и программного обеспечения;
- угроза неконтролируемого роста числа виртуальных машин;
- угроза неконтролируемого роста числа зарезервированных вычислительных ресурсов;

- угроза несанкционированного доступа к активному и (или) пассивному виртуальному и (или) физическому сетевому оборудованию из физической и (или) виртуальной сети;
- угроза несанкционированного доступа к виртуальным каналам передачи;
- угроза несанкционированного доступа к гипервизору из виртуальной машины и (или) физической сети;
- угроза несанкционированного доступа к данным за пределами зарезервированного адресного пространства, в том числе выделенного под виртуальное аппаратное обеспечение;
- угроза несанкционированного доступа к защищаемым виртуальным машинам из виртуальной и (или) физической сети;
- угроза несанкционированного доступа к защищаемым виртуальным машинам со стороны других виртуальных машин;
- угроза несанкционированного доступа к защищаемым виртуальным устройствам из виртуальной и (или) физической сети;
- угроза несанкционированного доступа к системе хранения данных из виртуальной и (или) физической сети;
- угроза несанкционированного доступа к хранимой в виртуальном пространстве защищаемой информации;
- угроза ошибки обновления гипервизора;
- угроза перехвата управления гипервизором;
- угроза перехвата управления средой виртуализации.

Имеются различные направления работ для минимизации новых рисков и угроз технологий виртуализации платформ. Например, хорошие системы виртуализации содержат виртуальные коммутаторы и брандмауэры, которые обеспечивают защиту от сетевых атак между виртуальными машинами,

расположенными на одном физическом сервере. Гарантию изоляции процессов и данных виртуальных машин друг от друга для гипервизора виртуализации дают основные вендоры систем виртуализации – компании Microsoft, IBM, Citrix и VMware. Но следует понять, что это все работает в случае отсутствия компрометации гипервизора виртуальных машин. Поэтому никаких гарантий исключения рисков и угроз виртуализации платформ нет. К тому же, в настоящее время сами технологии виртуализации все активнее начинают применяться для нарушения конфиденциальности, целостности и доступности информации (в шпионских и других целях).

Потенциальные внутренние уязвимости виртуализации платформ могут быть выявлены лишь тестированием на проникновение, для реализации которого может использоваться такое удобное и доступное средство каким является специализированная ОС Kali Linux [3]. Бесплатная ОС Kali Linux создана на основе Debian дистрибутива для проведения тестирования на проникновение и аудита безопасности, имеет видоизмененное специальное ядро. Это специальное ядро защищено от инъекций, что позволяет безопасно проводить аудит беспроводных сетей. Kali Linux поддерживает большое количество беспроводных устройств, является совместимым с USB и другими беспроводными устройствами, включает более 300 инструментов для проведения тестирования на проникновение и многоязычную поддержку. Существуют рабочие инсталляции для x86 и ARM-систем.

В заключение можно отметить, что технологии виртуализации дают новые возможности, но и предъявляют к ИТ-специалистам более высокие требования как в плане уровня профессионализма, так и в плане уровня ответственности.

Выводы по 2 главе

При выполнении 2 главы работы был проведен анализ современных решений виртуализации серверной инфраструктуры, изучено программное и аппаратное обеспечение в современных решениях серверной виртуализации, рассмотрены платформы виртуализации серверной инфраструктуры, определена проблематика и специфика внедрения технологий виртуализации.

Глава 3 ИССЛЕДОВАНИЕ МЕТОДОВ И АРХИТЕКТУРНЫХ РЕШЕНИЙ ВИРТУАЛИЗАЦИИ СЕРВЕРНОЙ ИНФРАСТРУКТУРЫ

3.1 Выбор технологий для сравнительного анализа виртуализации серверной инфраструктуры

Для анализа серверов существуют ряд средств, которые позволяют оценить производительность виртуальной системы. Ниже представлены технологии и средства, с помощью которых можно оценить производительность не только виртуальных, но и реальных систем. Перед их рассмотрением опишем термины, которые будут описаны в описании.

Утилита — компьютерная программа для выполнения специализированных задач, связанных с работой оборудования и операционной системы устройства. Это небольшие программы, которые помогают производить анализ каких-либо данных, настройку или другими способами облегчать работу пользователя с системой.

Пользователь видит утилиты в качестве ярлыков на рабочих столах, файлов скриптов, вызывает их из командной строки или двойным щелчком. Внутри утилит содержится программный код, который осуществляет вычислительные действия.

Утилиты могут входить в состав операционных систем, идти в комплекте со специализированным оборудованием или распространяться отдельно (в таком случае их нужно скачать либо установить из репозитория).

Производительность (вычислительная мощность) — это скорость выполнения определённых операций на компьютере, которую можно представить (измерить) количественной характеристикой (процентами, секундами, флопсами).

Оценка реальной вычислительной мощности производится путём прохождения специальных тестов, предназначенных для проведения некоторых операций и измерения времени их выполнения.

В случае анализа серверов всю работу по тестированию и оценке данных (насколько это возможно) будут производить утилиты и выводить данные на устройства выводы.

Для верной интерпретации полученных фактов необходимо также понимать взаимосвязи между различными аппаратными компонентами, в сумме они влияют на производительность сервера (вне зависимости от целевой ОС).

Для упрощения понимания результатов необходимо рассматривать отдельно несколько параметров работы системы. В сумме их показатели влияют на общую производительность системы. Измерив всего один показатель, нельзя сказать, что работа происходит плохо, долго или некорректно, т. е. значение имеет комплексный показатель.

Сейчас большинство серверов работают на основе операционной системы Linux, поэтому рассмотрим её более подробно. Вторая популярная система Windows тоже будет рассмотрена.

Важно помнить, что приведённые утилиты имеют свои аналоги в обеих системах. Поэтому будут рассмотрены основные принципы работы наиболее популярных команд и программ. Дополнительные данные можно найти в открытом доступе на ресурсах производителей, участников сообществ поддержки и пользователей систем.

Анализ системы начинается использованием команды `top`. Данная утилита широко используется для анализа программ в режиме реального времени. Набрав в командной строке `top`, в консоли отображается динамическая выдача процессов, которые в данный момент выполняются, спят или ожидают своей очереди. Интересными с точки зрения анализа сервера являются данные, которые отображены вверху окна в соответствии с рисунком 3.1.

```

top - 19:18:29 up 12 days, 16:55,  8 users,  load average: 1,80, 1,77, 1,98
Tasks: 289 total,  4 running, 284 sleeping,  0 stopped, 1 zombie
%Cpu(s): 33,6 us,  2,7 sy,  0,0 ni, 63,7 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem:  12211328 total, 11368356 used,  842972 free, 172500 buffers
KiB Swap:  6105648 total, 1405972 used, 4699676 free. 3704548 cached Mem

  PID USER      PR  NI  VIRT    RES  SHR  S  %CPU  %MEM     TIME+ COMMAND
 28124 admin7    20   0 3065672 1,508g 242044 R 132,6 12,9   3084:46 mysqld
 21572 admin7    20   0 3294116 1,518g 221012 S   8,0 13,0  1718:40 nginx

```

Рисунок 3.1 – Вывод команды top

Показатели, на которые стоит обратить внимание, помечены красным цветом, их показатели – зелёным. Значения выделены вручную для наглядности.

Более наглядный вывод обеспечивает использование htop — это аналог рассматриваемой системной утилиты top.

Более подробно рассмотрим выводимые показатели и их значения. Показатели загрузки системы: load average и %CPU, us, id, wa.

1. load average. Состоит из трёх чисел и демонстрирует усреднённую загрузку сервера за 1, 5 и 15 минут. Чем ниже значения, тем лучше. Простое правило: значения не должны быть больше количества процессоров.

2. %CPU. Какие процессы сколько процессорных ресурсов потребляют:

- us. Загрузка пользовательскими процессами. Если сервер постоянно не загружен ресурсоёмкими операциями типа конвертации видео, то этот показатель не должен превышать 10-20%.

- id. Процент времени бездействия процессора должен быть высоким, в норме - от 80.

- wa. Ожидание операций ввода/вывода, чем ниже, тем лучше (иначе процессор слишком долго ждёт ответы от диска или сети).

Существует целый набор консольных утилит для измерения и анализа производительности системы - sysstat:

- `iostat` - показывает статистику использования процессора и потоков ввода/вывода для дисков;
- `mpstat` - выводит информацию об отдельных параметрах и общей статистике по процессору;
- `isag` - построение графика активности системы в интерактивном режиме;
- `pidstat` - мониторинг отдельных задач, управление которыми осуществляется ядром Linux.

Последнюю утилиту стоит рассмотреть подробнее. Для её использования можно применить информацию, полученную с помощью предыдущей программы - `top`.

`Pidstat` — это утилита, которая предназначена для сбора и вывода статистики использования ресурсов процессами. Команда сообщает об использовании процессорного времени: `pidstat -p 611,1102 10 1`.

PID необходимого процесса можно посмотреть в результатах команды `top`: первый столбец сообщается `process id` (указано PID 611 и 1102). В результате можно узнать количество выделяемых ресурсов процессам с определённым идентификационным номером в системе.

Далее указать время в секундах, в течение которого будет осуществляться проверка (в данном случае это 10 секунд). Пользователь задавать время на своё усмотрение в зависимости от задач, которые предстоит решить. В завершение указываем число отчётов, которые необходимо видеть по итогу.

При помощи флага `-d` можно получить статистику ввода/вывода (остальные показатели остались неизменными): `pidstat -p 1102 -d 10 1`

При помощи флага `-r` можно получить статистику использования оперативной памяти: `pidstat -p 1102 -r 10 1`.

Для подробного отчёта об использовании всей оперативной памяти на устройстве подойдёт: `free -h` в соответствии с рисунком 3.2.

	total	used	free	shared	buff/cache	available
Память:	7,3G	5,0G	1,0G	124M	1,2G	1,8G
Подкачка:	7,5G	1,3G	6,2G			

	total	used	free	shared	buff/cache	available
Mem:		488M		23M	370M	94M 441M
Swap:		0B	0B	0B		

Рисунок 3.2 - Использование оперативной памяти

Столбцы:

- total - Общее количество памяти
- used - Реально использующая и зарезервированная системой память
- free - Свободная память (total - used)
- shared - Разделяемая память (быстрое средство обмена данными между процессами)
- buff/cache - Буферы в памяти (страницы памяти, зарезервированные системой для выделения их процессам, когда они потребуют этого, также известна как heap-memory)
- available - Доступная для использования память

Строки:

- mem (память) - Показывает, сколько физической оперативной памяти сейчас свободно. Стабильно низкий объём свободной памяти сам по себе не говорит о каких-то проблемах, но за ним стоит начать следить, чтобы убедиться, что памяти будет хватать даже при пиковых нагрузках.
- swap (подкачка) - Показывает использование файла подкачки или виртуальной оперативной памяти. Если значение больше нуля, значит, часть данных не помещается в оперативную память и вытесняется на диск, а так как

дисковые операции чтения и записи гораздо медленнее аналогичных для памяти, то падает производительность всей системы.

Далее кратко рассмотрим утилиту PS. PS (от англ. process status) — программа, выводящая отчёт о работающих процессах.

Можно использовать следующую готовую команду, чтобы узнать топ ресурсоемких к памяти процессов: `ps aux --sort=-rssize | head -20`

Отсортировать и по виртуальному размеру - что процесс просит у ядра, но не получает: `ps aux --sort=-vsz | head -20`.

Использование процессорного времени. Утилита `vmstat` - используется для определения производительности системы в соответствии с рисунком 3.3.

```
procs -----memory----- --swap-- -----io----- -system-- ----
r  b swpd   free buff  cache si so bi bo   in cs us sy id wa st
0  0 7596 329216 13260 14506756 0 0 4625 26 42 145 4 1 81 13 0
```

Рисунок 3.3 - Использование процессорного времени

В этой утилите мы обращаем внимание на следующие показатели:

us (user) - Процессорное время, затраченное на пользовательские процессы (демоны и прикладного ПО).

sy (system) - Системное время, затраченное на пользовательские процессы (демоны и прикладного ПО).

in (interrupts) - Количество прерываний контекста

cs (context switches) - Количество переключений контекста

Утилита `nmon` - Программа называется Nigel's Monitor, или просто `nmon`. Она в реальном времени выводит сведения о различных показателях, характеризующих состояние сервера.

`Nmon` имеет текстовый интерфейс, поэтому для работы с ним достаточно подключиться к серверу по SSH в соответствии с рисунком 3.4.

```

nmon-14g [H for help] Hostname=ubuntu Refresh= 2secs 10:48.42
# # # # # ##### # #
## # ## ## # # ## #
# # # # ## # # # # #
# # # # # # # # # #
# ## # # # # # # # #
# # # # # ##### # #
# # # # # # # # # #

For help type H or ...
nmon -? - hint
nmon -h - full

To start the same way every time
set the NMON ksh variable

Use these keys to toggle statistics on/off:
c = CPU           l = CPU Long-term   - = Faster screen updates
m = Memory        j = Filesystems      + = Slower screen updates
d = Disks         n = Network         0 = Virtual Memory
r = Resource     N = NFS             v = Verbose hints
k = kernel       t = Top-processes   . = only busy disks/procs
h = more options          q = Quit

```

Рисунок 3.4 - Утилита nmon

В окне утилиты, показанном на рисунке ниже, надо указать, какие именно сведения интересуют, включая и отключая соответствующие информационные разделы.

Клавиша d на клавиатуре, nmon выведет данные обо всех дисках, которые подключены к серверу в соответствии с рисунком 3.5.

```

nmon-14g Hostname=ubuntu Refresh= 2secs 10:54.45
Disk I/O /proc/diskstats mostly in KB/s Warning:contains duplicates
DiskName Busy Read WriteKBIO 125 150 175 100|
sda 0% 0.0 0.01> |
sda1 0% 0.0 0.01> |
sda2 0% 0.0 0.01> |
sda5 0% 0.0 0.01> |
Totals Read-MB/s=0.0 Writes-MB/s=0.0 Transfers/sec=0.0

```

Рисунок 3.5 - Данные о дисках

Для получения данных со сведениями о сети и памяти предназначены клавиши n и m в соответствии с рисунком 3.6.


```

nmon-14g      Hostname=ubuntu      Refresh= 2secs  10:57.04
-----
Memory Stats
-----
RAM      High      Low      Swap      Page Size=4 KB
Total MB 3008.5   -0.0    -0.0     3070.0
Free MB  571.3   -0.0    -0.0     3070.0
Free Percent 19.0%  100.0%  100.0%   100.0%
-----
MB              MB              MB
Buffers= 139.4  Cached= 1858.9  Active= 1338.5
Dirty =   0.0  Swapcached= 0.0  Inactive = 839.8
Slab = 218.5  Writeback = 0.0  Mapped = 72.5
Commit_AS = 841.5  PageTables= 7.8
-----
Network I/O
-----
I/F Name Recv=KB/s Trans=KB/s packin packout insize outsize Peak->Recv Trans
enp0s3   0.0     0.0     0.0     0.0     0.0     0.0     8.4     0.4
lo       0.0     0.0     0.0     0.0     0.0     0.0     0.1     0.1
-----
Disk I/O  /proc/diskstats  mostly in KB/s  Warning:contains duplicates
-----
DiskName Busy  Read WriteKB10  125  150  175  100|
sda  0%  0.0  0.0|>
sda1 0%  0.0  0.0|>
sda2 0%  0.0  0.0|>
sda5 0%  0.0  0.0|>
Totals Read-MB/s=0.0  Writes-MB/s=0.0  Transfers/sec=0.0
-----

```

Рисунок 3.6 - Данные о сети и памяти

Echofish — система для сбора и анализа логов в реальном времени.

Fluentd — open-source система для сбора логов.

Flume — распределённая система сбора и агрегации логов.

Graylog2 — сервер для анализа событий и логов с возможностями оповещения.

Нека — система для работы с логами от Mozilla.

Kibana — инструмент для визуализации данных.

Logstash — инструмент для управления событиями и логами.

Octopussy — open-source решение для работы с логами.

Alerta — распределённая, масштабируемая и гибкая система мониторинга;

Cacti — инструмент с веб-интерфейсом, который собирает статистические данные за определённые временные интервалы и позволяет отобразить их в графическом виде.

Cabot — легко развёртываемая служба мониторинга и оповещений на собственном сервере.

Centreon — open-source инструмент для мониторинга производительности.

check_mk — набор расширений для Nagios.

Flapjack — система мониторинга маршрутизации уведомлений.

Monit — небольшой open-source инструмент для управления и мониторинга Unix-систем.

Munin — сетевой инструмент мониторинга ресурсов.

Naemon — сетевой инструмент для мониторинга, основанный на Nagios, но с улучшениями производительности и новой функциональностью.

Sasti — инструмент с веб-интерфейсом, который собирает статистические данные за определённые временные интервалы и позволяет отобразить их в графическом виде.

Cabot — легко развёртываемая служба мониторинга и оповещений на собственном сервере.

Centreon — open-source инструмент для мониторинга производительности.

check_mk — набор расширений для Nagios.

Flapjack — система мониторинга маршрутизации уведомлений.

Monit — небольшой open-source инструмент для управления и мониторинга Unix-систем.

Munin — сетевой инструмент мониторинга ресурсов.

Naemon — сетевой инструмент для мониторинга, основанный на Nagios, но с улучшениями производительности и новой функциональностью.

Для оценки базовой производительности сервера достаточно собрать информацию:

- Average Disk Queue — для жестких дисков
- % Processor Time — для процессора и процессов
- Committed Bytes — для оперативной памяти

Когда процессу нужен доступ к физическому ресурсу, операционная система ставит запрос в очередь. Если в очереди стабильно больше 2 элементов, значит, ресурс становится узким местом.

Для сравнительного анализа виртуализации серверной инфраструктуры выбраны 2 наиболее востребованные платформы виртуализации Hyper-V и VMware.

В качестве серверных технологий высокой доступности, выбраны следующие кластера:

- кластер на VMware ESXi 5.5.
- кластер на HyperV Windows Server 2012R255.

Кластера собраны на одинаковых серверах по 3 физических сервера на кластер HP ProLiant BL465c G7, находящихся в одной корзине Blade System c7000 Enclosure G2 со следующей конфигурацией:

- центральный процессор – 2 шт. AMD Opteron™ Processor 6167 (12 core) 2300 MHz;
- оперативная память (ОЗУ) – 16 GB;
- 2 HDD 72GB Raid 10.

В качестве среды хранения данных используем EMC VNX570, где каждому кластеру выделено по 2 дисковых раздела на 300 GB и 1GB.

Связь между дисковым массивом и корзиной организовано через фабрику на MDS9148 на скорости 4 Gbps.

Ниже представлена схема стенда для проведения сравнительных тестов кластеров в соответствии с рисунком 3.7.

В качестве критериев сравнения мы будем брать значимые, применимые ко всем сравниваемым объектам, систем виртуализации Microsoft Hyper-V и VMware vSphere параметры. Выберем для рассмотрения следующие пункты, приведенные в описании систем виртуализации:

- масштабируемость;
- интеграция с хранилищами данных;
- сетевое взаимодействие;
- доступность виртуальных машин;

- поддержка гостевых ОС и устройств;
- управление памятью и другими ресурсами хостов;
- развертывание и управление.

Так же мы проведем тесты систем на отказоустойчивость.

3.2. Средства управления и администрирования виртуальной средой

В процессе администрирования виртуальных сред у каждого системного администратора имеется свой набор инструментов, способов и методов. Для управления кластерами высокой доступности мы не брали специальные инструменты, а пользовались стандартными средствами предоставляемые самими системами (рис 3.8. – 3.17.)

Коммутатор Cisco Catalyst 6509-E Коммутатор Cisco Catalyst 6509-E

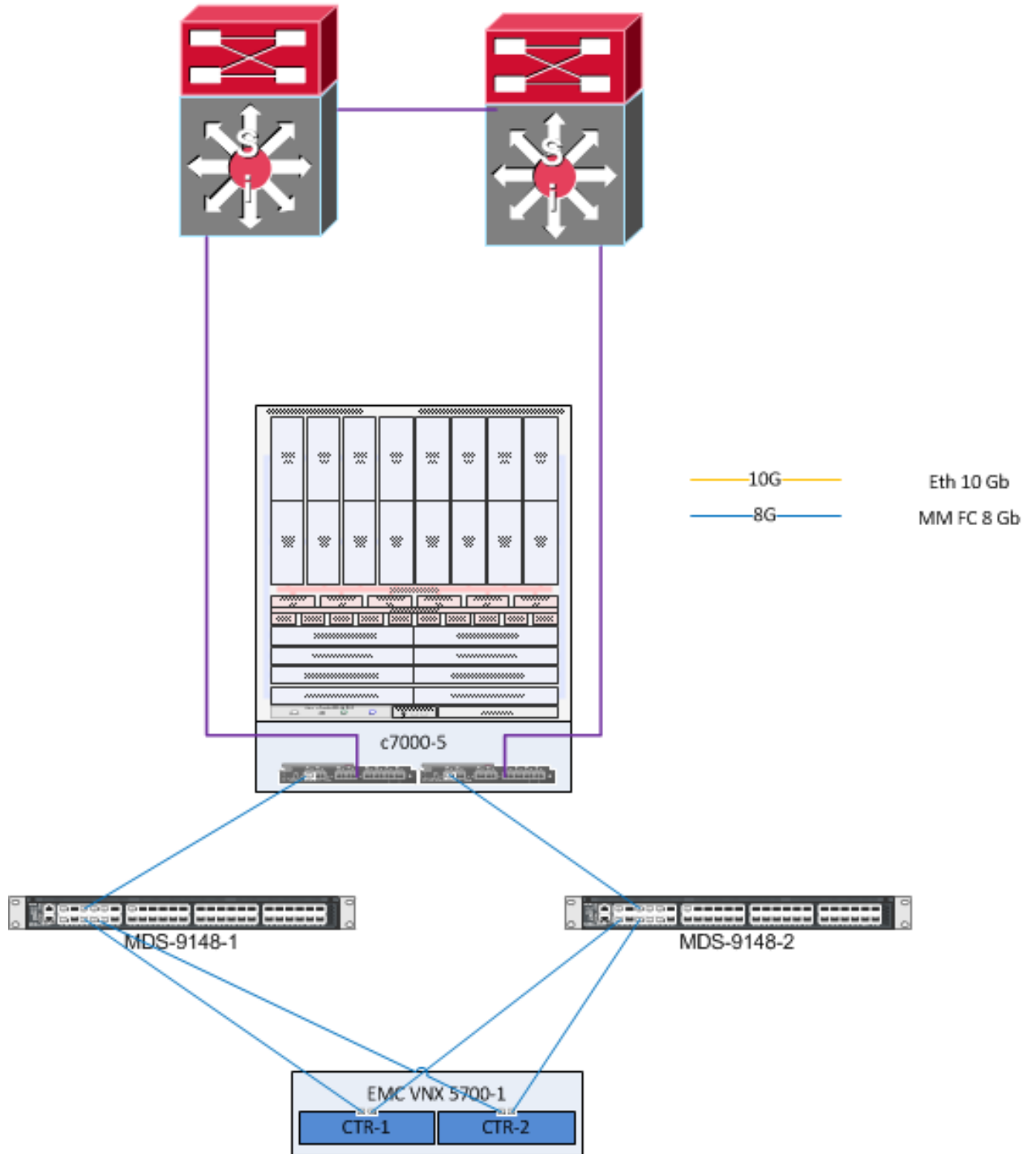


Рисунок 3.7 - Стенд для проведения тестов.

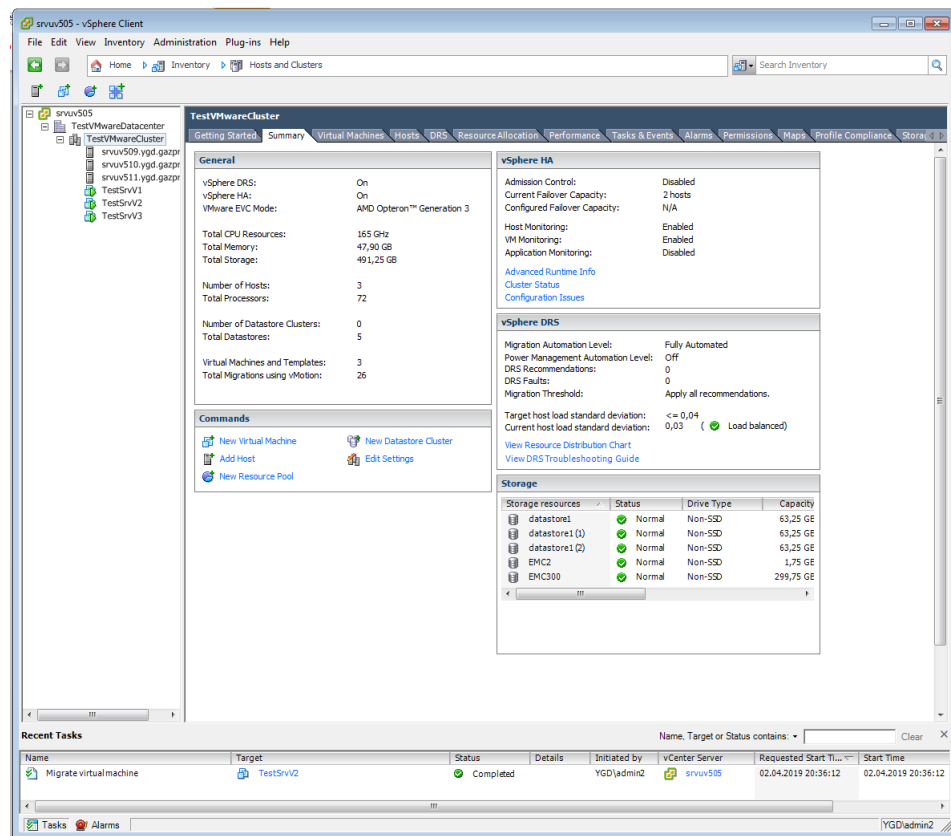


Рисунок 3.8 - Система управления VMware Cluster

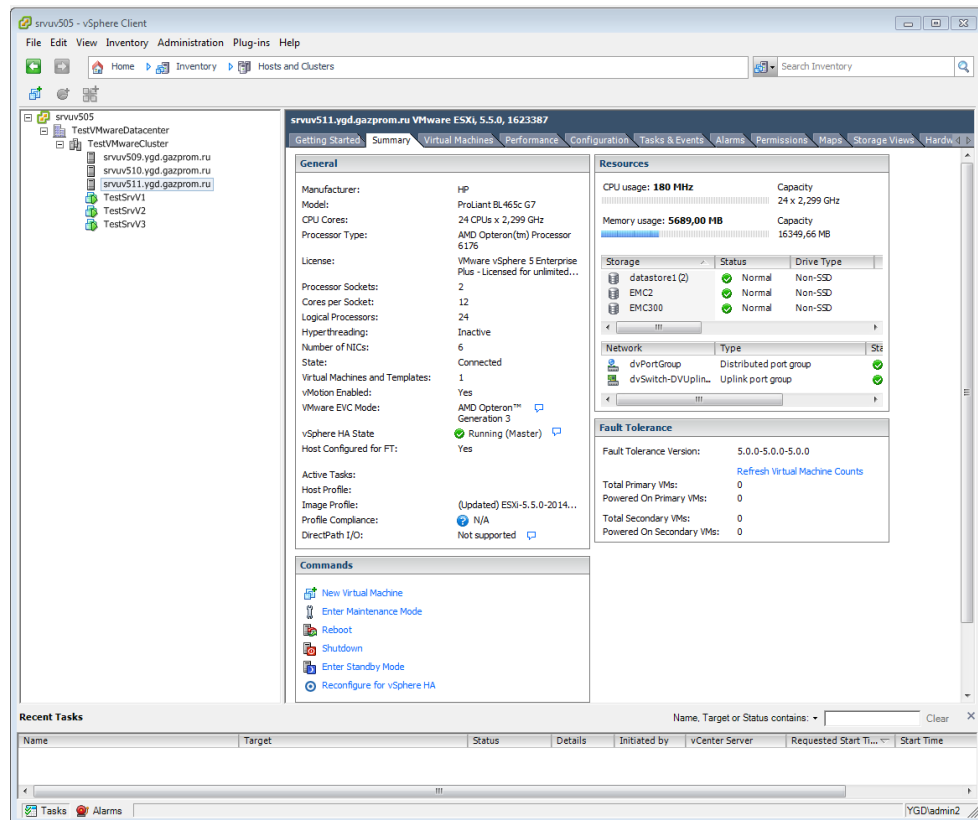


Рисунок 3.9 - Система управления VMware Host

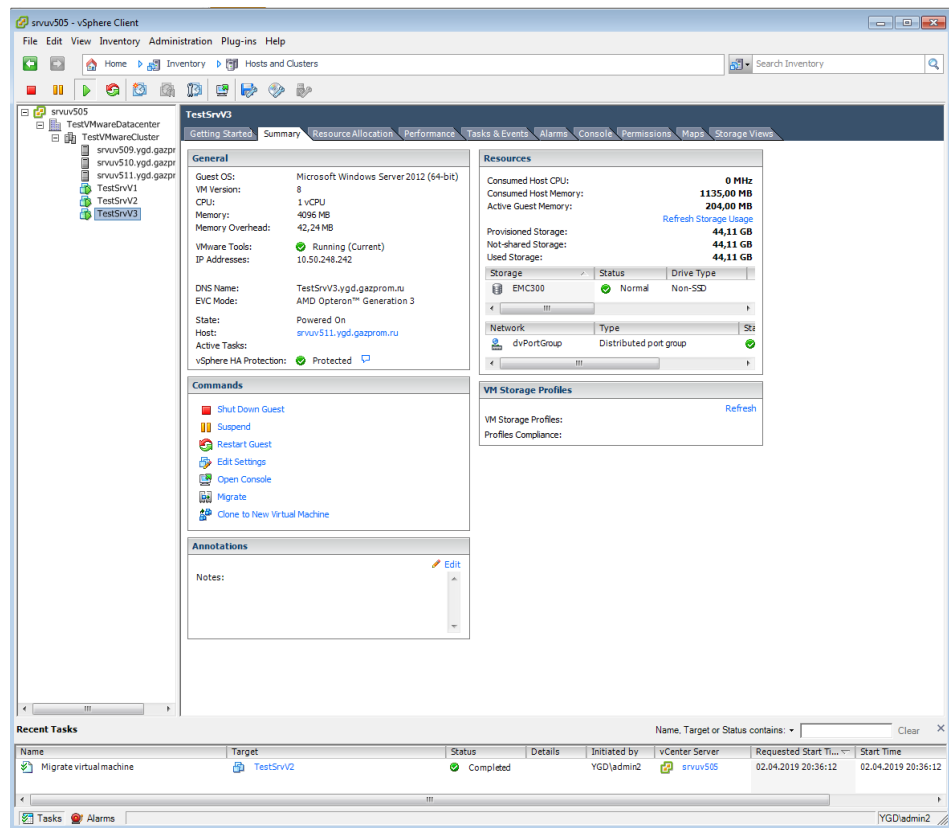


Рисунок 3.10 - Система управления VMware Virtual Machine

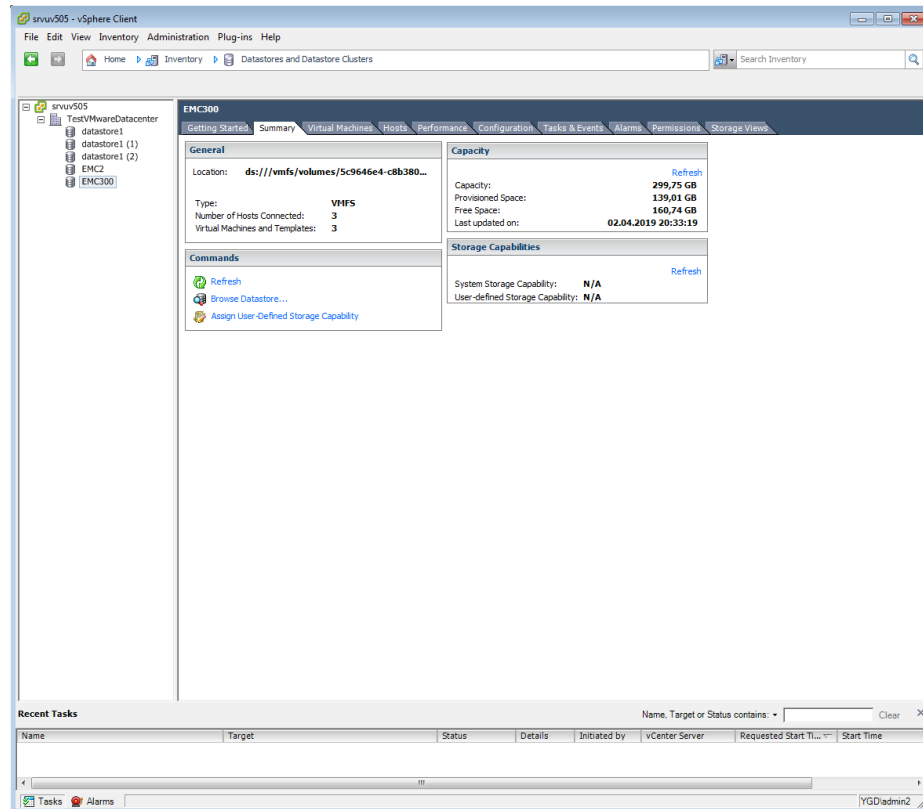


Рисунок 3.11 - Система управления VMware storage

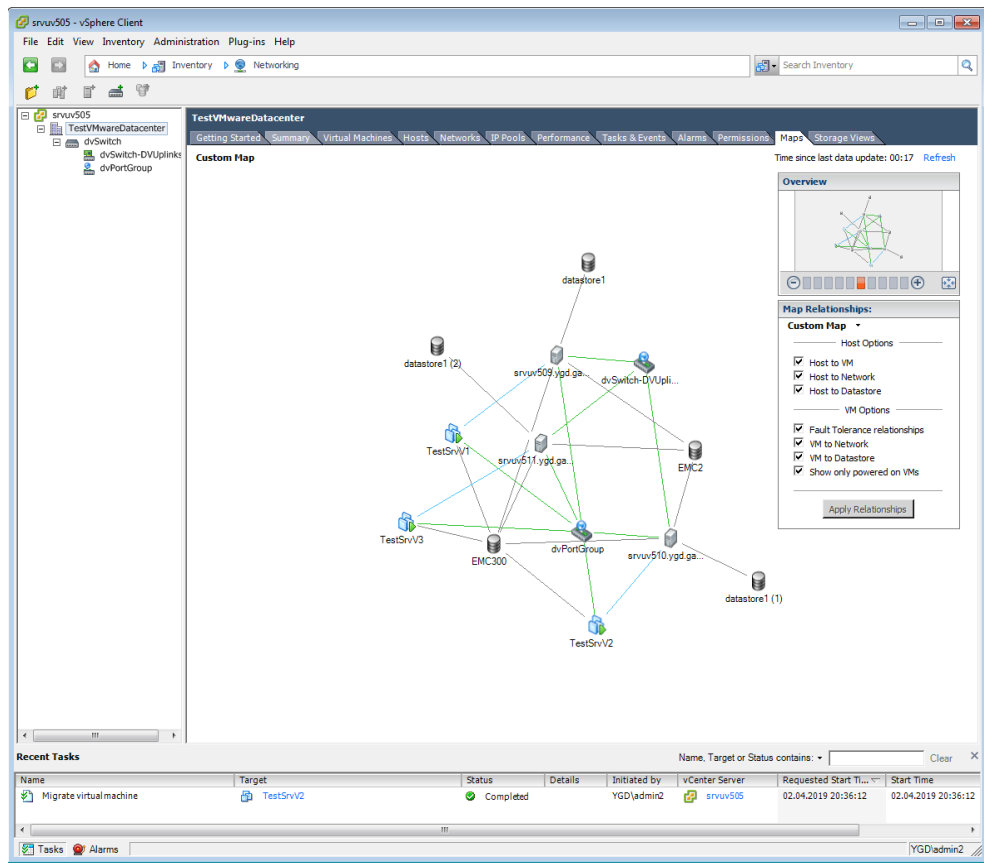


Рисунок 3.12 - Система управления VMware Networking

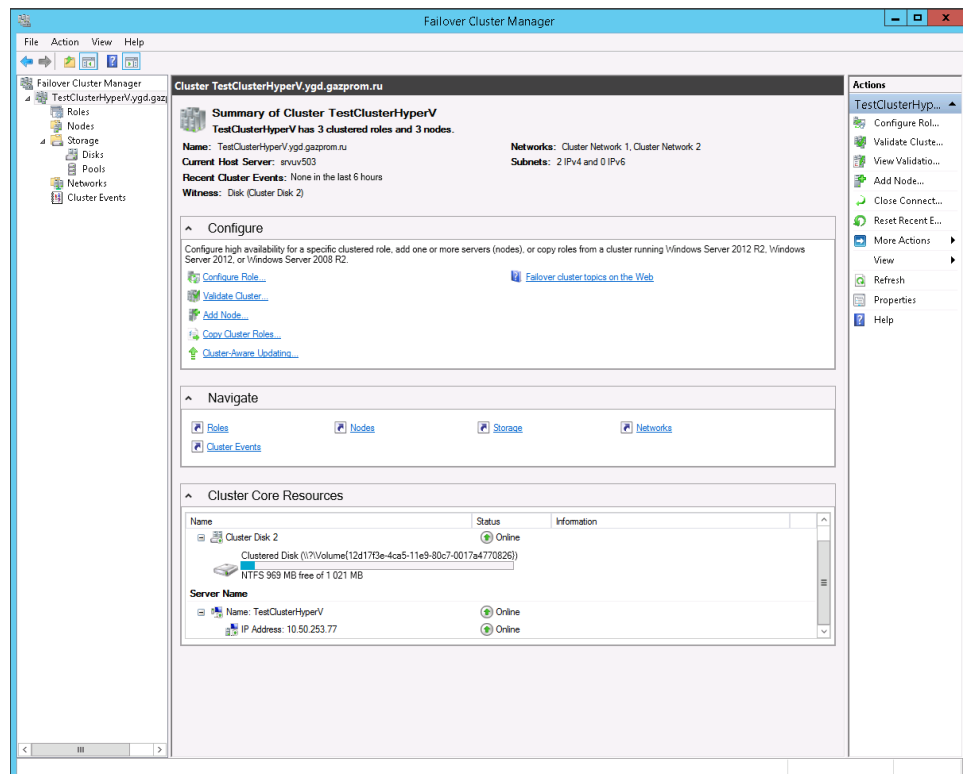


Рисунок 3.13 - Система управления Hyper-V Cluster

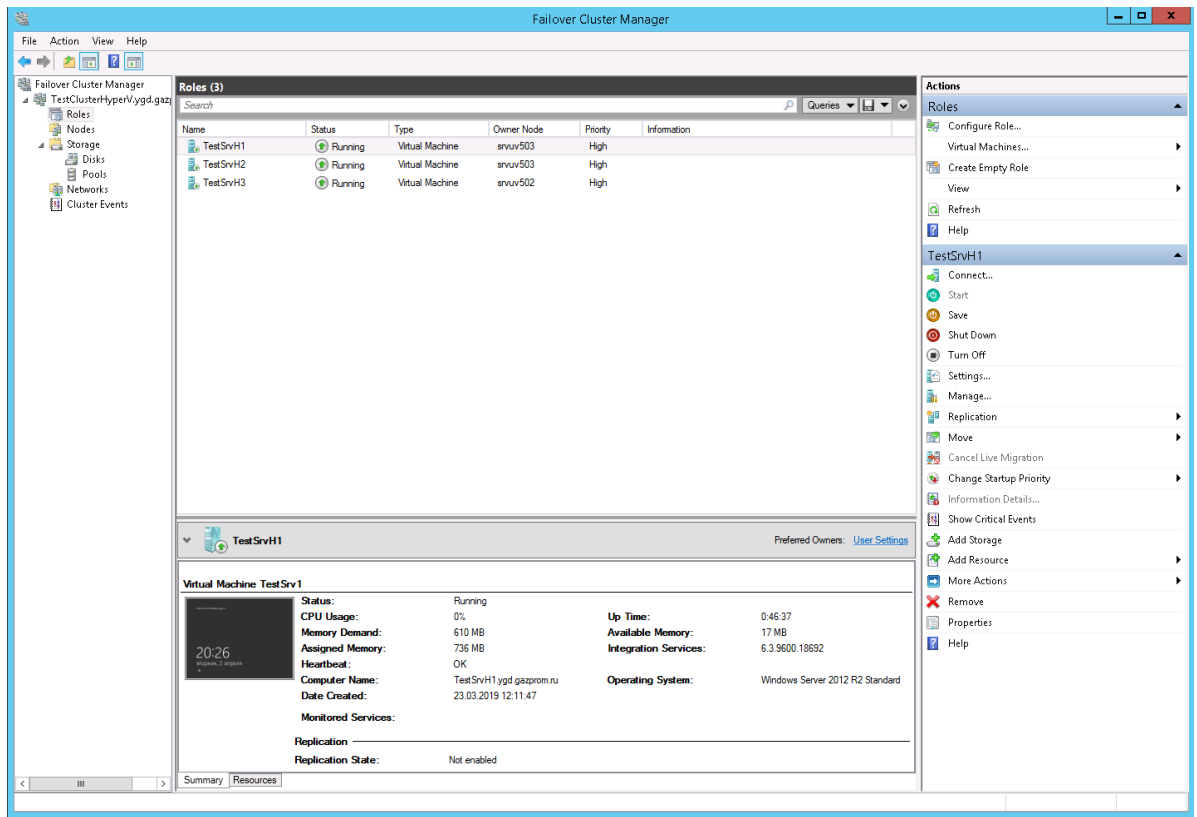


Рисунок 3.14 - Система управления Hyper-V Virtual Machine

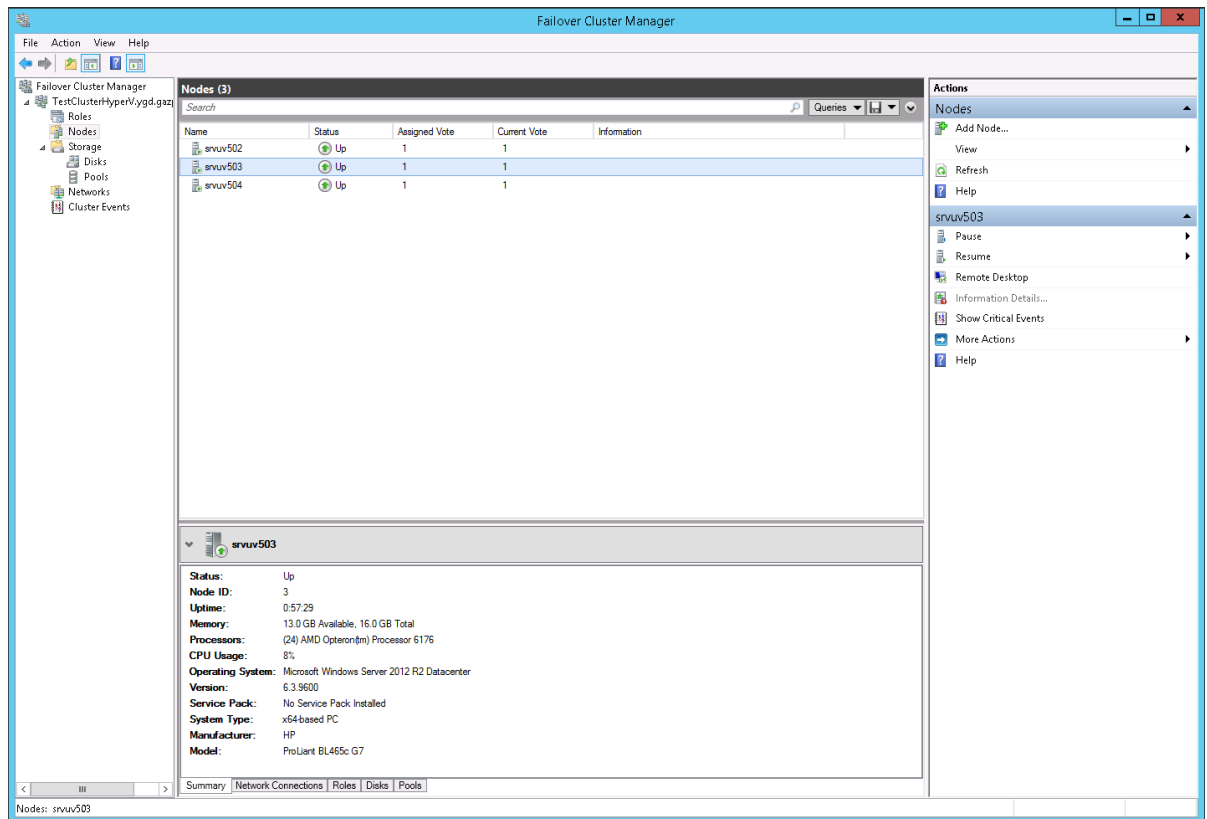


Рисунок 3.15 - Система управления Hyper-V Host

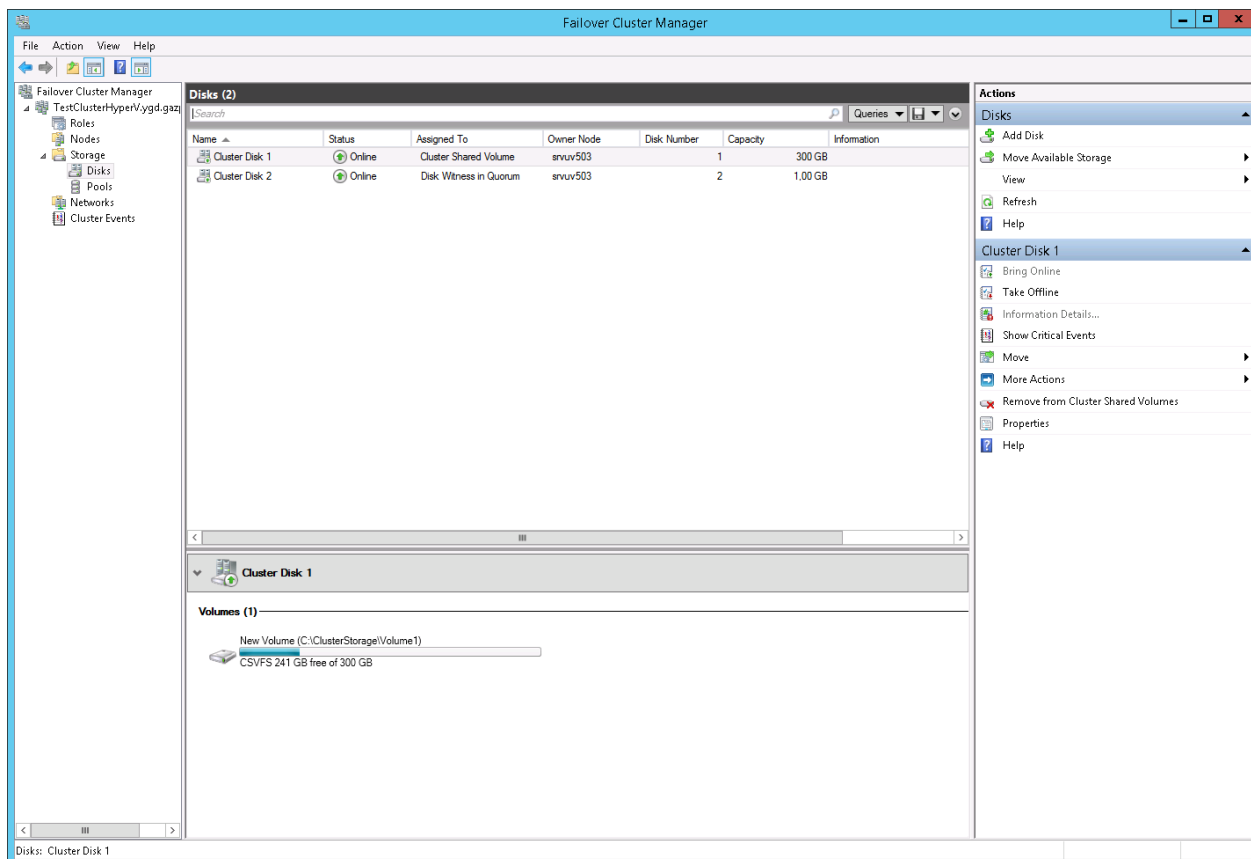


Рисунок 3.16 - Система управления Hyper-V Virtual Storage

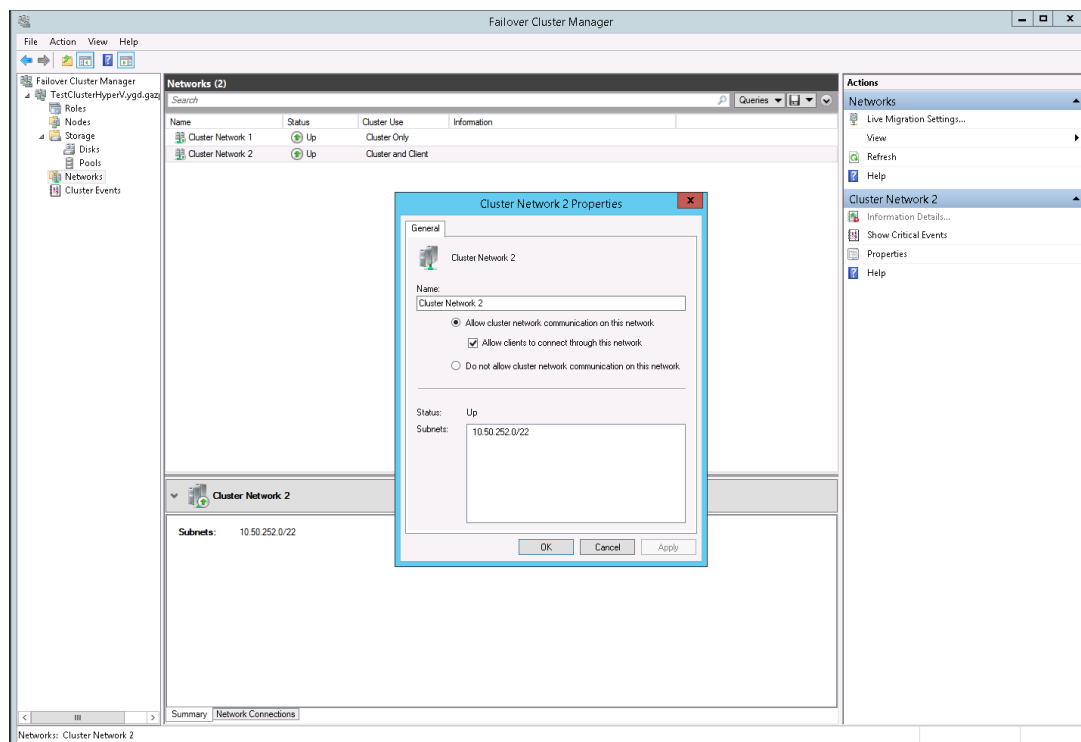


Рисунок 3.17 - Система управления Hyper-V Virtual Networking

3.3. Сравнение показателей технологических решений серверной виртуализации

После рассмотрения средств виртуализации и мониторинга изучены показатели функционирования виртуальных серверов. Данные анализа представлены ниже.

Анализ производительности виртуальных серверов осуществлялся путем размещения на них веб-приложений с разным объемом базы данных и генерации трафика. Было выбрано 4 конфигурации веб-приложения, которые представлены в таблице 3.1.

Таблица 3.1 – Конфигурация веб-приложений для виртуальных систем

№	Конфигурация	Редакция	Объем базы	Примечание
1	Управление торговлей	11.1.9.44	16,6 Гб	Рабочая
2	ERP Управление предприятием	2.0.9.38	676 Мб	Демонстрационная
3	Бухгалтерия предприятия	3.0.22.14	820 Мб	Рабочая
4	Управление производственным предприятием (УПП)	1.3.57.1	16,3 Гб	Рабочая

В качестве серверных технологий выбраны:

- VMware ESXi 5.1;
- физический сервер;
- Windows Server 2012R2 HyperV (1Gen);
- Windows Server 2012R2 HyperV (2Gen).

Ресурсы аппаратного сервера были выделены следующие:

- центральный процессор – Intel Core i5 3330 (3.0 Ghz);
- оперативная память (ОЗУ) – 16 GB DDD3;
- жесткий диск – SSD 240 Gb Intel.

Гипервизоры были развернуты на этой же аппаратной платформе с выделением 8 Гб оперативной памяти и 4-х виртуальных процессоров. Все базы кроме ERP 2.0 являются настоящими базами. В качестве СУБД использовался MSSQL 2012. Операционная система на физическом сервере и виртуальных машинах использовалась Windows Server 2012R2. Сервер приложений и СУБД устанавливались на один сервер и работали в режиме shared memory, данные отображены в таблицах 3.2 – 3.5.

Таблица 3.2 – Результаты замеров скорости операций конфигурации 1

Типовые операции	Hyper-V 2012. VM 1-st generation	Hyper-V 2012. VM 2-d generation	VMware ESXi	Аппаратный компьютер
Время запуска конфигурации, сек	58	61	20	18
Отчет по финансам – анализ движений денежных средств (за год) ,сек	9	5	2,5	5
Маркетинг – отчеты по маркетингу и планированию – АБС/XYZ анализ номенклатуры, сек	30	41	20	27
Продажи – Отчеты по продажам – Выручки и себестоимость продаж, сек	20	15	7,5	7
Закупки – отчеты по запасам и закупкам – Ведомости по остаткам на складе, сек	14	8	7	10
Склад – отчеты по складу – ведомость по товарам на складах, сек	24	3	1	2
Склад – перемещение товара, сек	9	15	6	3
Перепроведение квартала 3 квартал 2014г. 3381 документ, сек	3252	2987	1436	2003

Таблица 3.3 – Результаты замеров скорости операций конфигурации 2

Типовые операции	Hyper-V 2012. VM 1- st generation	Hyper-V 2012. VM 2- d generation	VMware ESXi	Аппаратный компьютер
Показатель теста Гилева 8.3	17,12	17,12	25	24,15
Время запуска конфигурации	64	65	36	40
Маркетинг и планирование – отчеты по маркетингу и планированию	1	1	2	1
Маркетинг и планирование – отчеты по маркетингу и планированию – Анализ зависимостей от клиентов (АБС)	1	1	1	1
Продажи-состояние обеспечения заказов	1	2	1	2
Закупки – отчеты по запасам и закупкам – остатки товаров, принятых на комиссию. (год)	2	6	1	2
Склад – отчеты по складу – ведомость по товарам на складе (год)	2	2	1	2
Склад – заказ на внутреннее потребление	1,5	1,5	1	2
Производство – Отчету по производству – состояние объектов эксплуатации	1	1	1	2
Зарплата – отчет по зарплате – расчетный листок по сотрудникам за год	21	22	16,5	22
Финансы – отчеты по финансам – Анализ движений денежных средств (год)	1,5	1,5	1	2
Финансы – финансовый результат – Закрытие результатов месяца	135	140	121	158
Бюджетирование – отчеты по бюджетированию – оборотная ведомость по статьям бюджета (год)	22	9	6	7
Международный финансовый учет – отчет по Международному финансовый учету – Анализ Субконто (год)	2	5	1	2

Таблица 3.4 – Результаты замеров скорости операций конфигурации 3

Типовые операции	Hyper-V 2012. VM 1-st generation	Hyper-V 2012. VM 2-d generation	VMware ESXi	Аппаратный компьютер
Время запуска конфигурации	8	19	9,4	11
Учет, налоги, отчетность – Отчет по проводкам (год)	3	8	3	5
Справочники и настройки учета – Обороты счета (год)	10	3	1	2
Справочники и настройки учета – Анализ счетов за год	2	2	1	2
Справочники и настройки учета – Оборотно-сальдовая ведомость (год, все показатели)	2	2	1	2

Таблица 3.5 – Результаты замеров скорости операций конфигурации УПП

УПП				
Типовые операции	Hyper-V 2012. VM 1-st generation	Hyper-V 2012. VM 2-d generation	VMware ESXi	Аппаратный компьютер
Время запуска конфигурации	44	30	20,9	30
Отчет по финансам – анализ движений денежных средств (за год)	3	2	0,5	1
Отчёты – продажи – Анализ (XYZ\АБС) (год)	76	92	73	80
Отчеты – Затраты – анализ распределения затрат (год)	27	31	16	22
Отчеты – Закупки – План-фактный анализ закупок (год)	6	8	5,3	10
Отчеты – Запасы – Товары на складах (год)	2	1	1	1
Заказ покупателя	1	1	1	1
Восстановление последовательности налогового учета УСН	5	4	1	1

Таким образом, виртуальные машины первого и второго поколения Hyper-V практически не отличаются друг от друга. Их производительность в ряде тестов отличалась, но нельзя с уверенностью сказать, какое поколение работает лучше, так как попеременно то одно, то другое поколение показывало результат лучше. Замеры производительности на VMware на виртуальной машине в большинстве случаев работала быстрее, чем на аппаратной платформе. Иногда показывая просто высокое превосходство – иногда до 40% меньше времени, чем на аппаратном компьютере. Отставание виртуальных машин на HyperV было более 108% и 126% для 2-го и 1-го поколения соответственно. Скорее всего, данный феномен происходит по причине лучшей работы с железом драйверов от VMware, против аналогичных от Microsoft. Также, возможно, ESXi формирует кэш для хранения данных, и таким образом, обрабатывает информацию быстрее.

Следующим этапом исследования было оценивание производительности виртуальных серверов облачных провайдеров.

У каждого облачного провайдера последовательно запускаются в разных зонах доступности (если зоны две, то 1 машина первой зоне и 2 во второй) три виртуальные машины с 4 CPU, 8 GB оперативной памяти и системным диском на 50 GB. Тип процессора/инстанса — новейший из доступных, если есть выбор.

Тип VM — shared с полным выделением ядер.

Тип дисков — сетевой SSD с возможностью перемонтирования на другую VM.

Опции гарантированного выделения IOPS или машины, оптимизированные под это, не использовались, если это не предусмотрено стандартными условиями использования и отказаться от этого нельзя.

Файловая система дефолтная — ext4.

Никакие ручные настройки системы не производились.

На каждой из машин запускалась серия тестов, итоговые значения по каждой машине были усреднены. Итоговая производительность платформы

выражается средним арифметическим усредненных значений тестов для каждой из виртуальных машин, но стандартное отклонение тоже есть в таблицах для интересующихся. Операционная система — Ubuntu 16.04 последнего доступного уровня патчей.

Для запуска тестов был создан скрипт, из которого видно все параметры запуска. Скрипт тестирования выглядит в соответствии с рисунком 3.18.

```
1  #!/usr/bin/env bash
2  TIME=60
3  # Workload 70% read 30% write
4  cat > fio-rand.fio << EOL
5  [global]
6  name=fio-rand-RW
7  filename=fio-rand-RW
8  rw=randrw
9  rwmixread=70
10 rwmixwrite=30
11 bs=4K
12 direct=1
13 numjobs=1
14 time_based=1
15 runtime=${TIME}
16 [file1]
17 size=2G
18 iodepth=16
19 EOL
20 echo "Run FIO"
21 for i in {1..3}; do
22     echo "$i iter:"
23     fio fio-rand.fio |grep -E "(read|write|bw|iops|READ|WRITE)" |grep -v "Disk"
24 done
25 echo "Run stress-ng."
26 for i in {1,2,4}; do
27     for z in {1..3}; do
28         echo -n "$z iter. Stress-NG for $i CPU: "
29         stress-ng --cpu $i --cpu-method matrixprod --metrics-brief -t $TIME 2>&1 |sed -n '6p'|awk '{print $5}'
30     done
31 done
32 for i in {1,2,4}; do
33     for z in {1..3}; do
34         echo -n "$z iter. Sysbench CPU for $i thread(s): "
35         sysbench --num-threads=$i --max-time=$TIME --test=cpu run 2>&1|grep "total time:"|awk '{print $3}'
36     done
37 done
38 for i in {1,2,4}; do
39     for z in {1..3}; do
40         echo -n "$z iter. Sysbench Memory for $i thread(s): "
41         sysbench --num-threads=$i --max-time=$TIME --test=memory run 2>&1| grep "Operations performed:"
42     done
43 done
```

Рисунок 3.18 – Скрипт тестирования

Данные по работе скрипта размещены в таблицах 3.6 – 3.10.

Таблица 3.6 - Яндекс.Облако

Test	Average	Avg Min	Avg Max	StDev
FIO READ IOPS	543,89	534,33	550,00	8,38
FIO WRITE IOPS	232,78	228,67	235,67	3,66
STRESS-NG 1 CPU	10526,11	10147,33	11117,33	518,72
STRESS-NG 2 CPU	20340,89	19842,00	21330,00	856,61
STRESS-NG 4 CPU	22039,67	18280,33	28703,67	5786,99
Sysbench CPU for 1	11,65	11,33	11,96	0,31
Sysbench CPU for 2	6,37	5,88	6,86	0,49
Sysbench CPU for 4	3,55	3,40	3,71	0,16
Sysbench Mem t 1	2137230,43	2081791,40	2195173,28	56732,39
Sysbench Mem t 2	2511195,10	2430456,68	2570853,73	72533,45
Sysbench Mem t 4	2968665,32	2841193,56	3089212,43	124154,35

Таблица 3.7 - Mail.RU Cloud (MCS)

Test	Average	Avg Min	Avg Max	StDev
1	2	3	4	5
FIO READ IOPS	487,00	538,00	534,00	28,36
FIO WRITE IOPS	209,00	231,00	229,00	12,17
STRESS-NG 1 CPU	7359,00	6567,00	7022,00	397,46
STRESS-NG 2 CPU	14144,00	14916,00	13137,00	892,08
STRESS-NG 4 CPU	21381,00	21199,00	21032,00	174,55
Sysbench CPU for 1	15,54	16,20	14,98	0,61
Sysbench CPU for 2	7,30	7,70	7,53	0,20
Sysbench CPU for 4	4,02	4,09	3,79	0,16
Sysbench Mem t 1	1117493,99	1161261,85	1423941,92	165744,17

Продолжение таблицы 3.7

1	2	3	4	5
Sysbench Mem t 2	1819474,62	1692128,17	1668347,81	81262,88
Sysbench Mem t 4	2357943,97	2379492,56	2312976,14	33938,38

Таблица 3.8 - Google Cloud

Test	Average	Avg Min	Avg Max	StDev
FIO READ IOPS	924,00	910,00	888,00	18,15
FIO WRITE IOPS	396,00	391,00	380,00	8,19
STRESS-NG 1 CPU	14237,00	14137,00	14094,00	73,37
STRESS-NG 2 CPU	28576,00	28419,00	28544,00	82,96
STRESS-NG 4 CPU	29996,00	29880,00	29449,00	288,22
Sysbench CPU for 1	12,63	12,66	12,67	0,02
Sysbench CPU for 2	6,52	6,41	6,38	0,08
Sysbench CPU for 4	3,35	3,56	3,56	0,12
Sysbench Mem t 1	2055240,49	2056617,63	2054720,94	980,13
Sysbench Mem t 2	1377683,73	1346931,63	1397680,79	25563,81
Sysbench Mem t 4	2279937,89	2275427,56	2278615,94	2318,63

Таблица 3.9 - Amazon Web Services (AWS)

Test	Average	Avg Min	Avg Max	StDev
1	2	3	4	5
FIO READ IOPS	1953,56	1937,33	1966,00	14,70
FIO WRITE IOPS	839,33	832,33	844,67	6,33
STRESS-NG 1 CPU	21677,11	21626,67	21753,00	66,90
STRESS-NG 2 CPU	42960,33	42348,00	43277,67	530,41

Продолжение таблицы 3.7

1	2	3	4	5
STRESS-NG 4 CPU	40601,11	40041,67	40882,67	484,50
Sysbench CPU for 1	8,77	8,77	8,77	0,00
Sysbench CPU for 2	4,40	4,39	4,40	0,00
Sysbench CPU for 4	2,52	2,52	2,52	0,00
Sysbench Mem t 1	3057058,93	3042532,67	3064915,47	12594,10
Sysbench Mem t 2	1991422,87	1814117,21	2163753,34	174871,16
Sysbench Mem t 4	2597272,55	2320876,43	2818418,98	253330,90

Таблица 3.10 - Azure

Test	Average	Avg Min	Avg Max	StDev
FIO READ IOPS	1094,56	1068,67	1134,00	34,71
FIO WRITE IOPS	469,11	458,33	485,33	14,30
STRESS-NG 1 CPU	10353,56	10096,00	10641,00	273,73
STRESS-NG 2 CPU	20819,33	20489,00	21347,00	461,79
STRESS-NG 4 CPU	40362,44	39158,00	41720,33	1288,04
Sysbench CPU for 1	10,10	9,80	10,42	0,31
Sysbench CPU for 2	5,19	5,08	5,35	0,14
Sysbench CPU for 4	2,71	2,66	2,77	0,05
Sysbench Mem t 1	2536842,28	2457812,89	2623752,76	83250,19
Sysbench Mem t 2	2409424,00	2352580,84	2496940,39	76912,65
Sysbench Mem t 4	2752661,55	2676802,64	2814777,95	70006,71

Проведены сравнения полученных результатов и представлены в соответствии с рисунками 3.19 – 3.22.

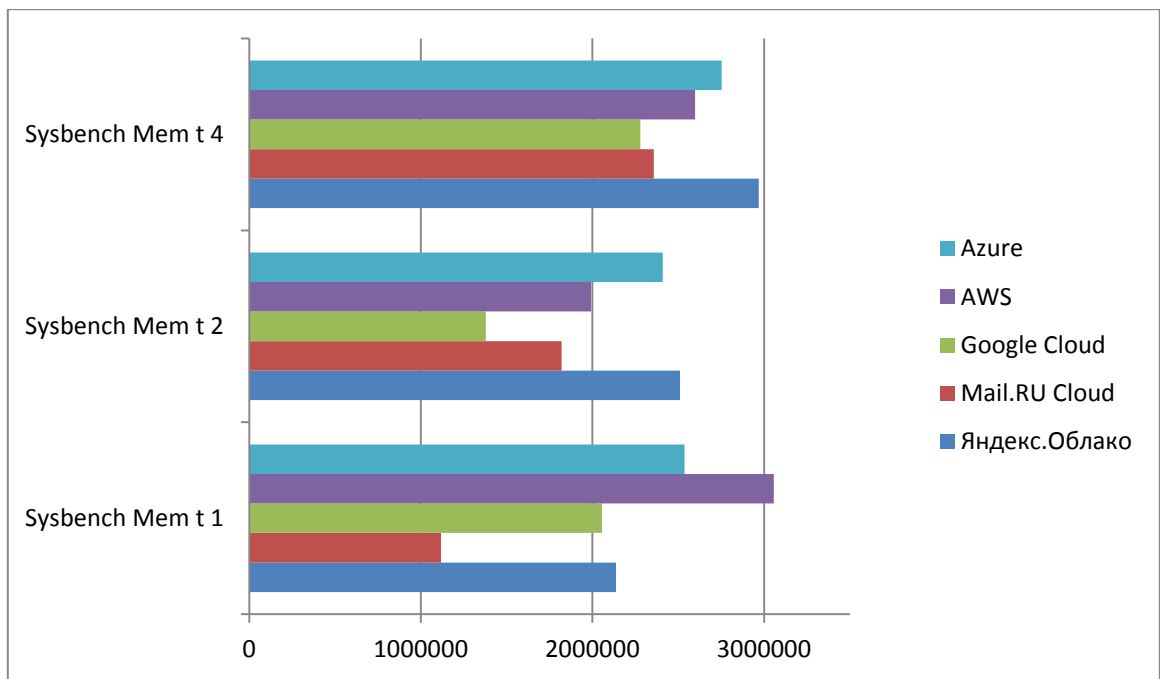


Рисунок 3.19 - Измерения производительности последовательных операций чтения/записи в оперативную память по времени

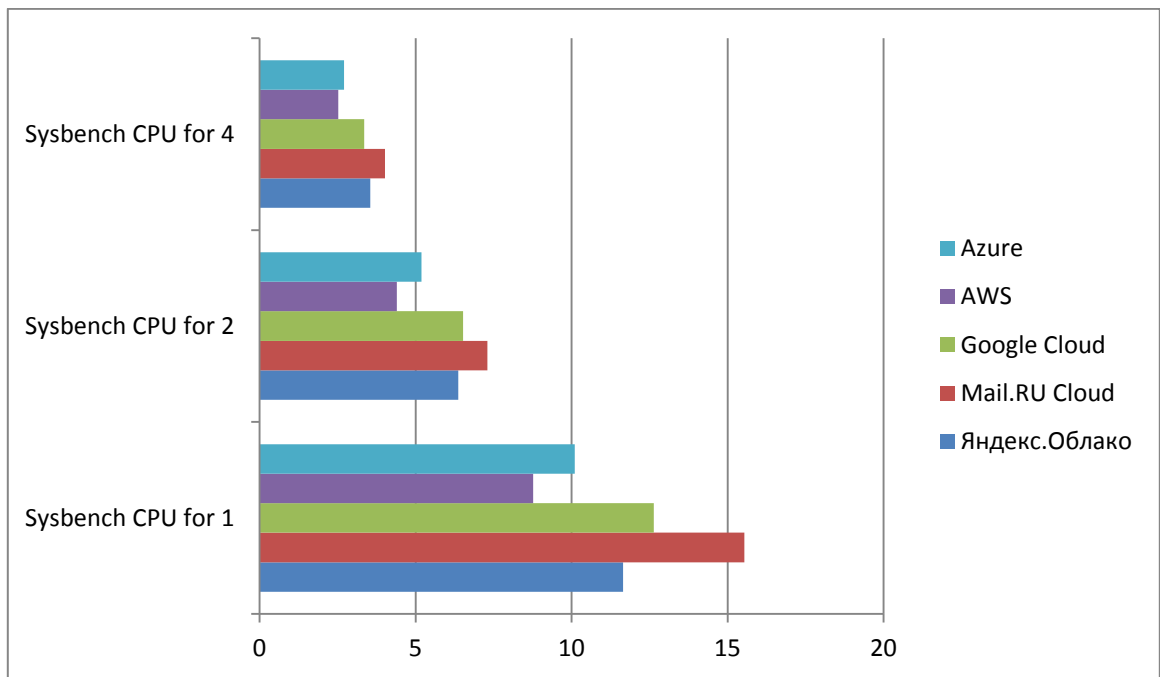


Рисунок 3.20 - Тест проверки производительности процессоров на время.

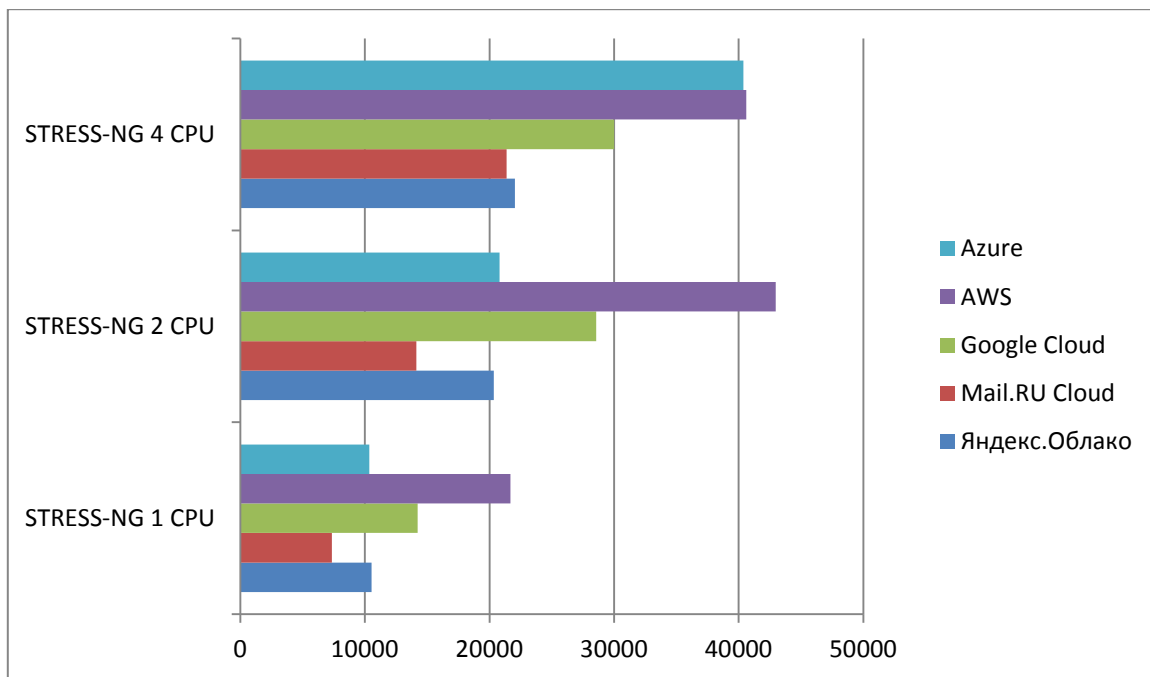


Рисунок 3.21 – Тесты на скорость выполнения операций.

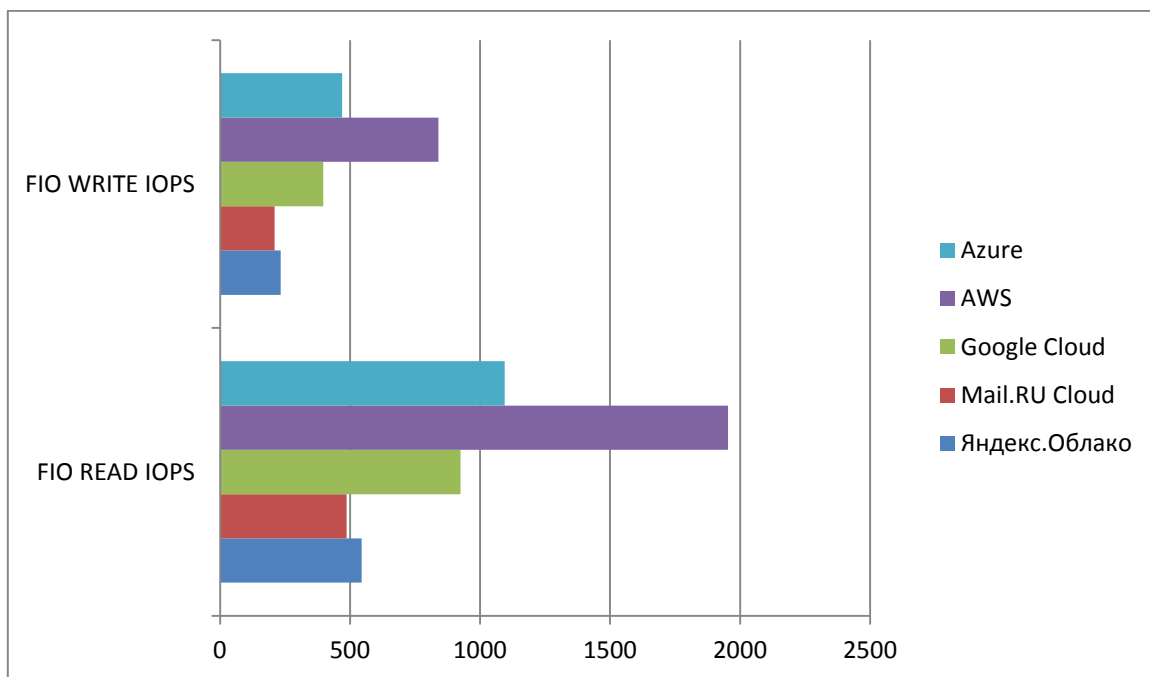


Рисунок 3.22 - Тестирование производительности дисковой подсистемы.

Из проведенных тестов мы видим что явным лидером из облачных сервисов является Amazon Web Services.

После рассмотрения средств виртуализации и мониторинга изучены показатели функционирования кластеров высокой доступности. Данные анализа представлены в таблице 3.11.

Таблица 3.11 – Сравнительный анализ кластеров виртуализации

Группа возможностей	Возможность	VMware vSphere 6.0 Enterprise Plus	Microsoft Hyper-V в Windows Server 2012 R2 Datacenter Edition	
1	2	3	4	
Возможности гипервизора	Версия гипервизора	VMware ESXi 5.5	Hyper-V 2012 R2	
	Максимальное число запущенных виртуальных машин	1024	1024	
	Максимальное число процессоров (CPU) на хост-сервер	480	320	
	Число ядер на процессор хоста	Не ограничено	Не ограничено	
	Максимальное число виртуальных процессоров (vCPU) на хост-сервер	4096	2048	
	Максимальный объем памяти (RAM) на хост-сервер	6 ТБ	4 ТБ	
	Техники Memory overcommitment (динамическое перераспределение памяти между машинами)		Memory ballooning	Dynamic Memory
	Техники дедупликации страниц памяти		Transparent page sharing	Нет

1	2	3	4
	Поддержка больших страниц памяти (Large Memory Pages)	Да	Да
Управление платформой	Централизованное управление	vCenter Server + vSphere Client + vSphere Web Client, а также виртуальный модуль vCenter Server Appliance (vCSA)	System Center Virtual Machine Manager (SC VMM)
	Интеграция с Active Directory	Да, как для vCenter, так и для ESXi-хостов через расширенный механизм SSO	Да (через SC VMM)
	Поддержка снимотов (VM Snapshot)	Да, снимоты могут быть сделаны и удалены для работающих виртуальных машин	Да, технология Checkpoint, включая функции live export
	Управление через браузер (тонкий клиент)	Да, полнофункциональный vSphere Web Client	Ограниченное, через Self Service Portal
	Обновления хост-серверов / гипервизора	Да, через VMware Update Manager (VUM), Auto Deploy и CLI	Да - Cluster Aware Updates, Fabric Updates, Management Servers

1	2	3	4
	Управление сторонними гипервизорами	Да, бесплатный аддон Multi-Hypervisor Manager	Да, управление VMware vCenter и Citrix XenCenter поддерживается в SC VMM
	Обновление (патчинг) виртуальных машин	Да, через VMware Update Manager (VUM) и vCenter Configuration Manager (vCM)	Да (WSUS, SCCM, VMST)
	Режим обслуживания (Maintenance Mode)	Да, горячая миграция VM в кластере DRS на другие хосты	Да
	Динамическое управление питанием	Да, функции Distributed Power Management в составе DRS	Да, функции Power Optimization в составе Dynamic Optimization
	API для решений резервного копирования	Да, vStorage API for Data Protection	Да, VSS API
	Шаблоны виртуальных машин (VM Templates)	Да + Multi-site content library	Да, включая шаблоны Gen2
	Профили настройки хостов (Host Profiles)	Да, расширенные функции host profiles и интеграция с Auto Deploy	Да, функции Physical Computer Profiles
	Решение по миграции физических серверов в виртуальные машины	Да, VMware vCenter Converter	Нет, больше не поддерживается

1	2	3	4
	Горячая миграция виртуальных машин	Да, vMotion между хостами, между датацентрами с разными vCenter, Long Distance vMotion (100 ms RTT), возможна без общего хранилища	Да, возможна без общего хранилища (Shared Nothing), поддержка компрессии и SMB3, неограниченное число одновременных миграций
	Горячая миграция хранилищ VM	Да, Storage vMotion, возможность указать размещение отдельных виртуальных дисков машины	Да
	Профили хранилищ	Да, Storage policy-based management	Да, Storage Classifications
	Кластер непрерывной доступности VM	Да, Fault Tolerance с поддержкой до 4 процессоров VM, поддержка различных типов дисков, технология vLockstep	Нет
Конфигурации виртуальных машин	Виртуальных процессоров на VM	128 vCPU	64 vCPU
	Память на одну VM	4 ТБ	1 ТБ

1	2	3	4
	Последовательных портов (serial ports)	32	Только присоединение к named pipes
	Поддержка USB	До 20 на одну машину (версии 1,2 и 3)	Нет (за исключением Enhanced Session Mode)
	Горячее добавление устройств	(CPU/Memory/Disk/NIC/PCIe SSD)	Только диск и память (память только, если настроена функция Dynamic memory)
	Устройства Floppy в ВМ	2	1
	Сетевые адаптеры/интерфейсы	10 (любая комбинация из поддерживаемых адаптеров)	8 устройств типа Network Adapter, 4 типа Legacy
	Виртуальные диски IDE	4	4
	Емкость виртуального диска машины	62 ТБ	64 ТБ для формата VHDX
	Поддержка гостевых ОС	174 ОС и их версий	Поддержка всех windows систем и сертифицированных под Hyper-V других систем
Кластеры и управление ресурсами	Максимальное число узлов в кластере	64	64
	Виртуальных машин в кластере	8000	8000
	Функции высокой доступности при сбоях хост-серверов	High Availability (HA)	Failover Clustering

1	2	3	4
	Перезапуск виртуальных машин в случае сбоя на уровне гостевой ОС	Да	Да
	Обеспечение доступности на уровне приложений	Да (App HA)	Да (Failover Clustering)
	Непрерывная доступность ВМ	Да, Fault Tolerance	Нет
	Репликация виртуальных машин	Да, vSphere Replication	Да, Hyper-V Replica
	Автоматическое управление ресурсами кластера	Да, Dynamic Resource Scheduler (DRS)	Да, Dynamic Optimization
	Пулы ресурсов	Да (Resource Pools)	Да (Host Groups)
	Проверка совместимости процессоров при миграциях машин	Да, EVC (Enhanced vMotion Compatibility)	Да, Processor Compatibility
Хранилища и сети	Поддерживаемые хранилища	DAS, NFS, FCoE (HW&SW), iSCSI, FC, vFRC, SDDC	SMB3, FC, Virtual FC, SAS, SATA, iSCSI, FCoE, Shared VHDX
	Кластерная файловая система	VMFS v5 (Virtual Machine File System)	CSV (Cluster Shared Volumes)
	Поддержка Boot from SAN	Да (FC, iSCSI, FCoE - hw&sw)	Да (iSCSI, diskless, FC)
	Диски, растущие по мере наполнения данными (thin provisioning)	Да (thin disk и se sparse)	Да, Dynamic disks
	Поддержка Boot from USB	Да	Нет

1	2	3	4
	Хранилища на базе локальных дисков серверов	VMware Virtual SAN 6.0 с поддержкой конфигураций All Flash	Storage Spaces, Tiered Storage
	Уровни обслуживания для подсистемы ввода-вывода	Да, Storage IO Control (работает и для NFS)	Да, Storage QoS
	Поддержка NPIV	Да (для RDM-устройств)	Да (Virtual Fibre Channel)
	Поддержка доступа по нескольким путям (multipathing)	Да, включая расширенную поддержку статусов APD и PDL	Да (DSM и SMB Multichannel)
	Техники кэширования	Да, vSphere Flash Read Cache	Да, CSV Cache
	API для интеграции с хранилищами	Да, широкий спектр VASA+VAAI+VA MP	Да, SMI-S / SMP, ODX, Trim
	Поддержка NIC Teaming	Да, до 32 адаптеров	Да
	Поддержка Private VLAN	Да	Да
	Поддержка Jumbo Frames	Да	Да
	Поддержка Network QoS	Да, NetIOC (Network IO Control), DSCP	Да
	Поддержка IPv6	Да	Да
	Мониторинг трафика	Да, Port mirroring	Да, Port mirroring

Проведены тесты на отказоустойчивость кластеров высокой доступности. Во время работы кластеров были произведены плановые миграции виртуальных машин, а так же тестовое внеплановое отключение физического сервера. Данные представлены в таблице 3.12

Таблица 3.12 – Результаты замеров скорости операций кластеров

Типовые операции	Hyper-V 2012. VM	VMware	Аппаратный компьютер
Время запуска сервера	16 с	25с	230с
Плановая миграция,	Потеря 1 ping	Потеря 1 ping	
Миграция при выходе из строя физического сервера	30с	70с	
Миграция при выходе из строя физического сервера при отключенном сервере управления	34с	245с Миграция не выполнена	

После изучения сравнительного анализа кластеров виртуализации и проведения тестов мы видим, что по масштабируемости и ограничениям - системы виртуализации примерно равны, одинаково хорошо работают с хранилищами данных, настраивается гибкое сетевое взаимодействие. По возможностям лидирует системы виртуализации так же примерно равны, но есть некоторые возможности у VMware, отсутствующие у Hyper-V, такие как:

- USB Redirection;
- возможность горячего добавления CPU\RAM
- Fault Tolerance;
- Более обширный список поддерживающих систем для виртуализации.
- Distributed Resource Scheduler и Storage DRS системы использующиеся для балансировки нагрузок между ресурсами хостов.

В свою очередь в Hyper-V присутствует возможность не только добавлять дисковое пространство, но и убавлять.

Вместе с тем работа виртуальных машин под управлением VMware немного быстрее, но для работы кластера VMware требуется отдельно стоящий компьютер, в отличие от Hyper-V где роль главного арбитра выполняет один из гипервизоров.

Выводы по 3 главе

В результате выполнения 3 главы работы был проведен анализ методов и архитектурных решений серверной инфраструктуры, при этом были выбраны технологии для сравнительного анализа виртуализации, изучены и определены средства управления и администрирования виртуальной среды, проведено сравнение показателей технологических решений серверной виртуализации.

Глава 4 АНАЛИЗ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ АРХИТЕКТУРНЫХ РЕШЕНИЙ ВИРТУАЛИЗАЦИИ СЕРВЕРНОЙ ИНФРАСТРУКТУРЫ

4.1 Обобщение результатов сравнительного исследования технологий виртуализации серверной инфраструктуры

Одним из основных подходов к реализации серверной инфраструктуры является технология виртуализации - предоставление вычислительных ресурсов, абстрагированное от их реальной аппаратной реализации, например, одновременное выполнение нескольких, изолированных друг от друга, операционных систем и приложений на одном компьютере. Совокупность компьютерных ресурсов, эмулирующую работу отдельных компонентов аппаратного или программного обеспечения, или даже целого компьютера, принято называть виртуальной машиной. Наличие нескольких виртуальных на одном реальном компьютере обеспечивает возможность независимой работы на одном физическом сервере (узле) нескольких операционных систем и приложений.

При изучении серверной виртуализации были определены основные понятия этой технологии: сервер – компьютерное оборудование, предназначенное для поддержки функционирования информационных сервисов; виртуализация – технология представления вычислительных ресурсов, абстрагированных от аппаратной части; гипервизор – программа для реализации виртуализации; система хранения данных – комплекс оборудования, предназначенного для консолидации дискового пространства в рамках одной системы.

Виртуализация позволяет освободить приложения от ограничений единственного физического сервера. Она позволяет запускать несколько

приложений на одном сервере и дает каждому приложению возможность использовать ресурсы всей корпоративной сети.

Одним из серьезных преимуществ виртуализации является возможность динамического перемещения приложений с одного физического сервера на другой для обеспечения доступности ресурсов и непрерывности обслуживания. Виртуализация также предполагает обязательное использование сетей хранения данных (SAN) и сетевых хранилищ. Это делает сети основным элементом инфраструктуры и архитектуры ИТ.

Существует две основных технологии создания серверных систем вычислений, основанные на виртуализации серверов. В первой - виртуализация осуществляется с помощью гипервизора - программной надстройки над основной ОС, которая отделяет виртуальные машины от сервера и по мере необходимости динамически выделяет вычислительные ресурсы каждой ВМ (Amazon, Azure, VMWare) [30]. Вторая – использует изолированные контейнеры (OpenVZ, LXC (Linux Containers), Docker).

Подход с использованием виртуализации позволяет запускать в облаке операционные системы любых производителей, но теряя при этом в производительности от 8 до 12 процентов по сравнению с использованием физического сервера. Второй подход выгоднее с точки зрения вычислительной производительности системы и экономии дисковых ресурсов, так как контейнеры используют ядро основной системы. При этом пользователи ограничены в выборе ОС только дистрибутивами семейства GNU/Linux, что в большинстве случаев рассматривается как существенный недостаток контейнерной виртуализации. В тоже время, существенный выигрыш в производительности позволяет в этом случае использовать ресурсы аппаратного обеспечения виртуального сервера даже для высокопроизводительных вычислений.

На сегодняшний день существует несколько систем виртуализации. Рассмотрим наиболее функциональные из них: VMware, Microsoft Hyper-V, Linux KVM.

VMware реализовала динамическую трансляцию бинарных инструкций виртуальной машины. При этом подходе гипервизор анализирует все инструкции, которые пытается выполнить гостевая ОС. Привилегированные инструкции, а также «нехорошие» непривилегированные команды заменяются «на лету» на обращения к гипервизору. «Хорошие» непривилегированные инструкции передаются на выполнение непосредственно процессору [6]. VMware ESXi Server — это программный продукт для виртуализации уровня предприятия, предлагаемый компанией VMware. ESXi является гипервизором и ставится непосредственно «на железо», то есть при установке не требуют наличия на машине установленной операционной системы. ESXi позволяет разделить физический сервер на логические разделы, называемые виртуальными машинами. Включает в себя средства управления виртуальными ресурсами предъявляет определённый набор требований к аппаратному обеспечению: например, наличие поддержки виртуализации со стороны архитектуры процессора является обязательным.

Microsoft Hyper-V — система аппаратной виртуализации для x64-систем на основе гипервизора. Имеет широкий набор возможностей для управления ВМ. Отличительной чертой Hyper-V является его взаимосвязанность с ролью Active Directory (AD), которая, как и гипервизор, является компонентом Windows Server 2012. Hyper-V, в отличие от аналогов, не может использоваться отдельно от Windows Server.

Гипервизор Hyper-V в серверной операционной системе Windows Server позволяет создать универсальную платформу, повышающую эффективность работы виртуальных машин, с удобной возможностью добавления виртуальных машин и простым управлением ими. Расширенные функции, такие как создание

виртуальных подсетей и встроенные средства автоматизации, позволяют размещать и перемещать виртуальные машины в любое время и в любое место.

Гипервизор Hyper-V, встроенный в Windows Server, полезен вне зависимости от того уровня развития инфраструктуры предприятия. При этом возможна виртуализация любой рабочей нагрузки. Hyper-V способен поддерживать наиболее объемные и критически важные нагрузки, предоставляя доступ к большому числу различных функций.

KVM (Kernel-based Virtual Machine) — это программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86, которая поддерживает аппаратную виртуализацию на базе Intel VT (Virtualization Technology) либо AMD SVM (Secure Virtual Machine). Программное обеспечение KVM состоит из загружаемого модуля ядра, называемого `kvm.ko`, предоставляющего базовый сервис виртуализации, процессорно-специфического загружаемого модуля `kvm-amd.ko` либо `kvm-intel.ko`, и компонентов пользовательского режима (модифицированного QEMU). Все компоненты ПО KVM являются ПО с открытым исходным кодом.

Проверка представленных выше технологий, в которых использованы VMware ESXi 5.1, Mware ESXi 5.5 физический сервер, Windows Server 2012R2 HyperV (1Gen), Windows Server 2012R2 HyperV (2Gen), показала, что виртуальные машины первого и второго поколения Hyper-V практически не отличаются друг от друга. Замеры производительности на VMware. на виртуальной машине в большинстве случаев работала быстрее, чем на аппаратной платформе. Иногда показывая просто высокое превосходство – иногда до 40% меньше времени, чем на аппаратном компьютере. Отставание виртуальных машин на HyperV было более 108% и 126% для 2-го и 1-го поколения соответственно. Скорее всего, данный феномен происходит по причине лучшей работы с железом драйверов от VMWare, против аналогичных от Microsoft. Также, возможно, ESXi формирует кэш для хранения данных, и таким образом, обрабатывает информацию быстрее.

Работа кластеров высокой доступности показало не значительные отличия в кластерных решениях VMware и Hyper-V, однако для работы виртуальных машин в непрерывном режиме для VMware необходимо наличие отдельно стоящего сервера VMware vCenter server, для Hyper-V отдельно стоящий сервер для работы кластера не нужен.

4.2 Рекомендации по внедрению, использованию и поддержке технологий виртуализации серверной инфраструктуры

Для решения задач поддержки серверной инфраструктуры необходимо обеспечить круглосуточный мониторинг и обслуживание серверов, а также организовать группу специалистов для восстановления работоспособности оборудования. При рассмотрении вопросов поддержки серверной инфраструктуры ключевым становится разработка и реализация серверной архитектуры - чем проще архитектура (однотипные устройства, стандартизированные системы, описанные настройки и т. д.), тем меньше нужно ресурсов на ее поддержку, а значит – тем меньше затраты на поддержку.

Особое место в администрировании и поддержке серверной инфраструктуры занимает виртуализация. Разовые затраты на внедрение технологий виртуализации позволят экономить на поддержке серверов, обновлении оборудования, лицензиях Microsoft. При использовании технологий виртуализации время восстановления вышедшего из строя сервера занимает минуты, в то время как вышедший из строя «железный» сервер может восстанавливаться часами, так же виртуализация позволяет создавать вычислительные среды высокой доступности без значительной привязки к физическим серверам.

Один из основных инструментов управления серверами – система мониторинга, она позволяет в про-активном режиме отслеживать сбойные

сервисы серверов, загруженность памяти и процессоров, место на жестких дисках и немедленно реагировать на эти изменения.

Приведем образец оптимизации серверной инфраструктуры посредством программной платформы для виртуальных машин - VMware vSphere Enterprise 5.5. Указанную платформу необходимо развернуть на физических серверах, в качестве моделей серверов предложены HP ProLiant DL360 G6 и сервере HP ProLiant DL160 G6 с VMware vCenter, для консолидации и упрощения функций управления виртуальными машинами. В качестве системы хранения данных была - СХД IBM System Storage DS3512 с iSCSI Host-интерфейсом. СХД хранит на себе все пользовательские данные, а также образы виртуальных машин.

Коммутационная сеть iSCSI предлагается в виде двух выделенных коммутаторов Gigabit Ethernet CISCO Catalyst WS2960G Для обеспечения функций резервного копирования, предлагается выделенный Backup сервер на базе IBM System x3620 M3 с ПО Veritas Backup и подключенная к нему ленточная библиотека IBM TS3100. На виртуальных машинах рекомендуется ПО Veeam Backup & Replication для резервного копирования виртуальных машин. Для хранения «оперативного» бэкапа использованы возможности сервер HP ProLiant DL180 G6. Схема предлагаемой инфраструктуры приведена на рисунке 4.1.

В результате разработанная инфраструктура может решить множество задач и позволяет перейти от классической архитектуры «одна задача – один сервер» к архитектуре с системой без единой точки отказа и виртуальными серверами. Реализована система централизованного резервного копирования с промежуточным быстрым хранилищем на основе жестких дисков и долговременным хранением на магнитных лентах. Процесс резервного копирования полностью автоматизирован.

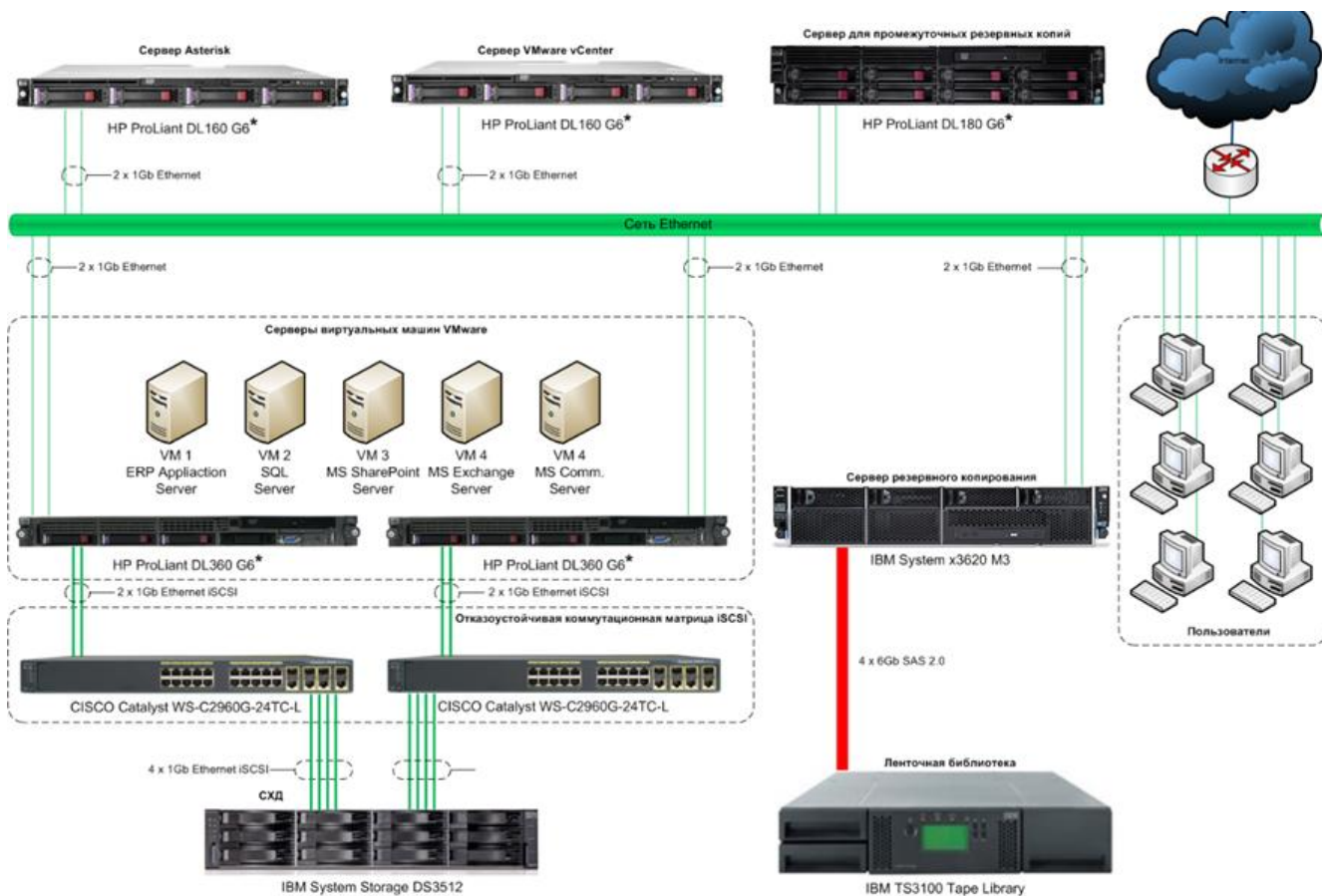


Рисунок 4.1 – Разработанная серверная архитектура

Использование технологий виртуализации позволяет масштабировать вычислительные мощности путем простого добавления физических серверов к системе. Основная дисковая СХД имеет внушительный запас по производительности и позволяет масштабироваться до 96 жестких дисков SAS или SATA.

Выводы по 4 главе

При выполнении 4 главы работы было проведено обобщение результатов сравнительного исследования технологий виртуализации серверной инфраструктуры, определены более производительные технологии серверной виртуализации. После этого были разработаны рекомендации по внедрению,

использованию и поддержке технологий виртуализации серверной инфраструктуры. При этом была разработана легко масштабируемая серверная инфраструктура, которая позволяет перейти к архитектуре с системой без единой точки отказа и виртуальными серверами. Использование технологий виртуализации позволяет масштабировать вычислительные мощности путем простого добавления серверов к системе.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы были решены все поставленные задачи:

- изучено понятие и особенности технологий виртуализации серверной инфраструктуры;
- проведен анализ современных решений виртуализации серверной инфраструктуры;
- исследованы методы и архитектурные решения виртуализации серверной инфраструктуры;
- проанализированы результаты исследования архитектурных решений виртуализации серверной инфраструктуры.

При изучении понятия и особенностей технологии виртуализации серверной инфраструктуры были рассмотрены современные способы организации серверной инфраструктуры, определено понятие и значение технологии виртуализации, изучены исторические аспекты развития технологий серверной виртуализации и особенности практического применения технологий виртуализации серверной инфраструктуры.

Виртуализация – это технология, которая дает возможность объединить приложения на различных платформах и аппаратных средствах предыдущих поколений с использованием меньшего числа современных, более мощных серверов с низким энергопотреблением. Виртуализация серверов является новейшим и главным технологическим трендом в центрах обработки данных. Несмотря на ограничения, касающиеся виртуализации серверов, она дает огромный положительный результат в отношении расходов предприятий на электроэнергию и охлаждение, а также емкости и мощности центров обработки данных.

При выполнении анализа современных решений виртуализации серверной инфраструктуры было изучено программное и аппаратное обеспечение в

современных решениях серверной виртуализации, рассмотрены платформы виртуализации серверной инфраструктуры, определена проблематика и специфика внедрения технологий виртуализации.

При исследовании методов и архитектурных решений проведен анализ решений серверной инфраструктуры, при это были выбраны технологии для сравнительного анализа виртуализации, изучены и определены средства управления и администрирования виртуальной среды, проведено сравнение показателей технологических решений серверной виртуализации.

На заключительном этапе работы было проведено обобщение результатов сравнительного исследования технологий виртуализации серверной инфраструктуры, определены более производительные технологии серверной виртуализации. После этого были разработаны рекомендации по внедрению, использованию и поддержке технологий виртуализации серверной инфраструктуры. При этом была разработана легко масштабируемая серверная инфраструктура, которая позволяет перейти к архитектуре с системой без единой точки отказа и виртуальными серверами. Использование технологий виртуализации позволяет масштабировать вычислительные мощности путем простого добавления серверов к системе.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Амелин К. С., Амелина Н. О., Граничин О. Н., Кияев В. И. Разработка приложений для мобильных интеллектуальных систем на платформе Intel Atom. — СПб: Издательство ВВМ, 2012. — 211 с.
2. Бердник А. В. Проблемы безопасности облачных вычислений. Анализ методов защиты облаков от cloud security alliance. Альманах современной науки и образования. В: Альманах современной науки и образования. Тамбов: Грамота, 2013. - № 10. - С. 35-38.
3. Богданов, А. В. Виртуализация: новые возможности известной технологии: монография / А. В. Богданов, Е. Н. Станкова, В.В. Мареев. – СПб.: Институт высокопроизводительных вычислений и интегрированных систем, 2009. – 31 с.
4. Газуль С. М. Принципы проектирования гибридной информационной системы для поддержки образовательного процесса в высшем учебном заведении // Вестник СПбГЭУ. Серия: Экономика. Выпуск 5 (72). — СПб: СПбГЭУ, 2014. — С. 58–61
5. Газуль С. М., Бабаев Э. О., Горнов П. А. Интегральный показатель готовности информационной системы к работе в облаке // Международный научно-исследовательский журнал — Research Journal of International Studies, 2014. — № 4-2 (23). — С. 14-16.
6. Герольд Ф. Энергетический баланс между лентой и дисками / Журнал сетевых решений. — М.: Издательство "Открытые системы" - 2008. — №4. — 108 с.
7. Граничин О. Н., Кияев В. И. Информационные технологии и системы в современном менеджменте. — СПб: Изд-во ВВМ, 2014. — 897 с.

8. Денисов Д. В. Перспективы развития облачных вычислений / Д. В. Денисов - М., 2009. - С. 36.
9. Зверев Г. И. Угрозы и методы обеспечения информационной безопасности виртуальных сред // Молодой ученый. - 2015. - № 9. - с. 235-237.
10. Зенченко Е. С. Сравнительный анализ систем хранения данных / Е. С. Зенченко // Электронный журнал Cloud of Science. - 2013.- № 3. – С. 22 – 26.
11. Исаев Е. А. Обеспечение информационной безопасности облачных вычислений / Е. А. Исаев, Д. В. Думский, В. А. Самодуров, В. В. Корнилов // Математическая биология и биоинформатика. – 2015. – Т. 10 - № 2. – С. 567 – 579.
12. Исаев Е. А. Проблема обработки и хранения больших объемов научных данных и подходы к ее решению. Математическая биология и биоинформатика / Е. А. Исаев, В. В. Корнилов. - 2013. - Т. 8. № 1. - С. 49-65.
13. Кириллов А. Г. Особенности применения информационных технологий в управлении гуманитарным вузом / А. Г. Кириллов // Периодический журнал. Среднее профессиональное образование. – 2013. – № 3. – С. 9-12.
14. Кириллов А. Г. Поддержка системы менеджмента качества вуза средствами информационных технологий / А. Г. Кириллов, В.М. Гордиевских, Д.М. Гордиевских // Зауральский научный вестник. – 2014. – № 1 (5). – С. 65-70.
15. Кириллов А. Г. Условия эффективного применения информационных технологий в управлении / А. Г. Кириллов // Преподаватель XXI век. – 2013. – Т. 1, № 2. – С. 12-15.
16. Кирсанов, В.А. Организация учебного процесса с использованием облачных технологий и технологий виртуализации / В. А. Кирсанов, А.М. Мухаметшин // Казанский кооперативный институт (филиал) АНОО ВО ЦС РФ «Российский университет кооперации. – 2014. – № 12-3. – С. 961-963.
17. Клементьев И. П., Устинов В. А.: Введение в Облачные вычисления. - УГУ, 2009. - 233 с.

18. Корнилов В. В. Перспективы использования центров обработки данных при решении задач математической биологии и биоинформатики / В. В. Корнилов, Е. А. Исаев, К. А. Исаев // Математическая биология и биоинформатика. 2015. Т. 10. № 1. С. 60-71.

19. Лебедева Т. А. Облачные технологии работы с документами / Т. А. Лебедева // Документ в современном обществе: между прошлым и будущим : тезисы X Всероссийской студенческой научно-практической конференции, г. Екатеринбург, 7-8 апреля 2017 г. / Урал. федер. ун-т им. Б. Н. Ельцина, Рос. гос. проф.-пед. ун-т. - Екатеринбург: Издательство Уральского университета, 2017. - С. 140-143.

20. Монахов Д. Н., Монахов Н. В. Облачные Технологии. Теория и практика / Д. Н. Монахов, Н. В. Монахов. - МАКС Пресс Москва, МГУ, 2013–128 с.

21. Нейронные сети в прикладной экономике: [учебное пособие] / Е. А. Трофимова, В. Д. Мазуров, Д. В. Гилёв; [под общ. ред. Е. А. Трофимовой]; М-во образования и науки рос. Федерации, Урал. федер. ун-т. — Екатеринбург: изд-во Урал. ун-та, 2017. — 96 с.

22. Оптимизация информационной структуры организации средствами современных технологий виртуализации: монография / В.М. Гордиевских, Д.А. Слинкин, А.В. Коуров ; Шадр. гос. пед. ин-т. – Шадринск : ШГПИ, 2014. – 96 с.

23. Программный комплекс корпоративной виртуализации ЕС2015-Терм. Рабочая документация. – Москва. – 2016. – 26 с.

24. Рыженкова А. Л. Использование облачных сервисов для хранения и обработки информации / А. Л. Рыженкова, Р. О. Сычов // Международный школьный научный вестник. 2018. - № 5. – С. 67 – 71.

25. Старцев М. В. Электронная коммерция как способ интенсификации бизнес-процессов // Социально-экономические явления и процессы. Тамбов, 2011. № 5-6. С. 212-215.

26. Сулейманов А. А. Существующие возможности технологий серверной виртуализации / А. А. Сулейманов // Материалы Международной научно-технической конференции, 2 – 6 декабря 2013 г. – С. 21 – 23.

27. Сушков С. А. Проблемы внедрения технологий виртуализации в образовательный процесс государственного вуза / С. А. Сушков // Гаудеамус: психол.-пед. журн. – 2011. – Т. 2, № 18. – С. 79-80.

28. Трофимов В. В. Конвергенция информационных технологий. — СПб: Изд-во С.-Петербур. гос. ун-та экономики и финансов, 2011. — 21 с.

29. Хабибуллин И. Ш. Разработка Web-служб средствами Java / И. Ш. Хабибуллин. – СПб.: БХВ-Петербург, 2003. – 400 с:

30. Харинцев С. С. Оптические методы хранения информации / С. С. Харинцев – Казань: Казан. ун-т, 2016. – 74 с.

31. Шарапов Р. В. Вопросы применения ленточных библиотек в многоуровневых системах хранения экологических данных // Машиностроение и безопасность жизнедеятельности, 2011. — № 2. — С.33-36.

Электронные ресурсы

32. Анализ современных технологий виртуализации [Электронный ресурс] // Хабрахабр [сайт]. [2017]. URL: <http://habrahabr.ru/company/centosadmin/blog/212985> (дата обращения: 12.03.2019)

33. Батура Т. В. Облачные технологии: основные понятия, задачи и тенденции развития [Электронный ресурс] / Т. В. Батура. Режим доступа: <http://swsys-web.ru/cloud-computing-basic-concepts-problems.html> (дата обращения: 01.03.2019).

34. Виртуализация в высокопроизводительных вычислительных системах [Электронный ресурс] // Наука и Образование: науч.-техн. изд. – Режим доступа: <http://technomag.edu.ru/doc/168323.html> (дата обращения: 01.03.2019).

35. Виртуализация для хостинга: тупик или прорыв? [Электронный ресурс] // Администрирование серверов. Обслуживание компьютеров. – Режим

доступа: <http://ha-systems.ru/virtualizacija-dlja-hostinga> (дата обращения: 01.03.2019).

36. Виртуализация на уровне операционной системы [Электронный ресурс] – Режим доступа. – URL: <http://ru.wikipedia.org/wiki/8B> (дата обращения 03.03.2019).

37. Виртуальный Linux - Обзор методов виртуализации, архитектур и реализаций [Электронный ресурс] // IBM developerWorks Россия. – Режим доступа: <http://www.ibm.com/developerworks/ru/library/l-linuxvirt/index.html> (дата обращения: 01.03.2019).

38. Джонс, Тим М. Виртуализация для встроенных систем [Электронный ресурс] // DeveloperWorks®. – Режим доступа: <http://www.ibm.com/developerworks/ru/library/l-embedded-virtualization/> (дата обращения: 01.03.2019).

39. Жесткий диск и его настройка [Электронный ресурс] / Режим доступа <http://dppc.ru/data/attachments/library/03-zhestkij-disk-i-ego-nastrojka-28980.pdf> (дата обращения: 01.03.2019).

40. Зорин В. Технологии виртуализации и защищенность информационных систем. [Электронный ресурс] // URL: <http://www.itsec.ru/articles2/Oborandteh/tehnologii-virtualizacii-i-zaschischennostj-informacionnyh-sistem> (дата обращения 23.02.2019).

41. Каспаринский Ф. О. Оптимизация распределения данных, информации и медиаресурсов между локальными и облачными хранилищами [Электронный ресурс] / Ф. О. Каспаринский, Е. И. Полянская // Режим доступа: <http://www.keldysh.ru/abrau/2016/13.pdf> (дата обращения: 01.03.2019).

42. Милосердов А., Гриднев Д. Тестирование на проникновение с помощью Kali Linux 2.0. [Электронный ресурс] // URL: <https://codeby.net/forum/resources/testirovanie-na-proniknovenie-s-pomoschju-kali-linux-2-0-v-pdf-formate.2> (дата обращения 23.02.2019).

43. Обзор, установка, настройка и использование открытой системы виртуализации Xen [Электронный ресурс] – Режим доступа: <http://www.ibm.com> (дата обращения 23.02.2019).

44. Облачные хранилища данных. Сервис Яндекс Диск. Методические рекомендации [Электронный ресурс] / Режим доступа: <http://13.work.temocenter.ru/images/pages/projects/pedmasterskaya/3/Yandex.pdf> (дата обращения: 01.03.2019).

45. Самойленко А. Что такое виртуализация и виртуальные машины [Электронный ресурс] – Режим доступа. – URL: <http://www.ixbt.com/cm/virtualization.shtml> (дата обращения 03.03.2019).

46. Сандби Ник. Выбор решений для хранения данных. Отчет об исследовании [Электронный ресурс] / Режим доступа: <http://www.fujitsu.com/ru/Images/wp-idc-eternus-dx-s3-ww-ru.pdf> (дата обращения: 01.03.2019).

47. Сибиряков М. А. Способы повышения производительности систем хранения данных / М. А. Сибиряков // Программные системы: теория и приложения. – 2016. - №2-1(29). – С. 27 – 34.

48. Сибиряков М. А. Сравнительный анализ основных моделей интеллектуальных систем хранения данных в процессе их эволюции / М. А. Сибиряков // Информационные технологии в профессиональной деятельности и научной работе. – 2012. – Ч.1. – С. 99 – 105.

49. Силаков Д. Виртуализация на платформе x86 / Д. Силаков / Системный администратор. – 2013. - № 10 (131). – С. 80 – 84.

50. Системы хранения данных [Электронный ресурс] / Режим доступа: http://partners.aq.ru/business-partners/mediateka/makets-buklet/promomat/mpages/shd/CXD_broshura.pdf (дата обращения: 01.03.2019).

51. Смолевицкая М. Э. Магнитные носители информации [Электронный ресурс] / М. Э. Смолевицкая // Информатика. – 2008. – № 19 -20. Режим доступа:

<http://informat444.narod.ru/museum/pres/pdf/inf-19-2008.pdf> (Дата обращения: 01.2019).

52. Хабрахабр, Микросервисы (Microservices). [Электронный ресурс] – <https://habrahabr.ru/post/249183/> (дата обращения: 12.03.2019)

53. Хабрахабр, Основы Kubernetes. – <https://habrahabr.ru/post/258443/>

54. Хабрахабр, Функциональная безопасность [Электронный ресурс] Часть 2 из 7. МЭЖ 61508. – <https://habrahabr.ru/post/309636/> (дата обращения: 12.03.2019)

55. Чех И. М. Внедрение облачных технологий в бизнес компании [Электронный ресурс] / И. М. Чех // Материалы VIII Международной студенческой электронной научной конференции «Студенческий научный форум» Режим доступа: <http://www.scienceforum.ru/2017/2290/32987> (дата обращения: 01.03.2019).

56. Штомпель И., Павел Емельянов: «Разработка ядра Linux – это общение в «клубе по интересам». // «Системный администратор», № 7-8, 2013 г. – С 62-66. Режим доступа: <http://samag.ru/archive/article/2487> (дата обращения 23.02.2019).

57. Hyper-V: комплексная платформа виртуализации [Электронный ресурс] Режим доступа: www.microsoft.ru/windowsserver2012 (дата обращения 08.03.2019).

Литература на иностранном языке

58. An Updated Performance Comparison of Virtual Machines and Linux Containers [Электронный ресурс] // IBM Research [сайт]. [2014]. URL: <http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B> (дата обращения: 12.03.2019)

59. Antonopoulos N. Cloud Computing: Principles, Systems and Applications [Электронный ресурс] / N. Antonopoulos, L. Gillam. – L.: Springer, 2010. – 379 p.

60. CloudHarmony – Transparency for the cloud [Электронный ресурс] // Cloudharmony. – Режим доступа: <http://cloudharmony.com> (дата обращения: 12.03.2019)
61. CloudSleuth – Decoding the mysteries of the cloud [Электронный ресурс] // Compuware. – Режим доступа: <https://cloudsleuth.net> (дата обращения: 12.03.2019)
62. Docker, Docker Documentation [Электронный ресурс]. – <https://docs.docker.com/> (дата обращения: 12.03.2019)
63. Docker, Swarm mode overview [Электронный ресурс]. – <https://docs.docker.com/engine/swarm/> (дата обращения: 12.03.2019)
64. Fabrizio Soppelsa, Chanwit Kaewkasi. Native Docker Clustering with Swarm.– Packt Publishing, 2016. – 280 с.
65. Hess K. Practical Virtualization Solutions: virtualization from the trenches, 2010. — 336 p.
66. Jennings R. Cloud Computing with the Windows Azure Platform [Text] / R. Jennings. – Wrox, 2009. – 360 p.
67. Kubernetes, Kubernetes Documentation [Электронный ресурс]. <https://kubernetes.io/docs/home/> (дата обращения: 12.03.2019)
68. Martin Fowler, Microservices [Электронный ресурс]. – <https://martinfowler.com/articles/microservices.html> (дата обращения: 12.03.2019)
69. Martin L. Abbott. The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise. – Addison-Wesley Professional, 2015. – 624 с.
70. Minakov V.F., Ilyina O.P., Lobanov O.S. Concept of the Cloud Information Space of Regional Government//Middle-East Journal of Scientific Research, 2014.— № 21 (1). — P. 190-196
71. Musaev A. A., Gazul S. M., Anantchenko I. V. The information infrastructure design of an educational organization using virtualization technologies //

Известия Санкт-Петербургского государственного технологического института (технического университета), 2014. — № 27 (53). — С. 71-76

72. Pethuru Raj. Docker: Creating Structured Containers. – Packt Publishing, 2016. – 320 с.

73. Platform9, Container Orchestration Tools: Compare Kubernetes vs Docker Swarm [Электронный ресурс]. – <https://platform9.com/blog/compare-kubernetes-vs-docker-swarm/> (дата обращения: 12.03.2019)

74. Rajesh RV, Spring Microservices. – Packt Publishing, 2016. – 436 с.

75. Randall Smith. Docker Orchestration. – Packt Publishing, 2017. – 284 с.

76. Real Distribution of Response Time Instability in Service-Oriented Architecture [Text] / A. Gorbenko, V. Khar-chenko, S. Mamutov, et al. // Proc. 29th IEEE International Symposium on Reliable Distributed Systems (SRDS'2010). – Delhi (India), 2010. – P. 92-99.

77. Sam Newman. Building Microservices: Designing Fine-Grained Systems. – O'Reilly Media, 2015. – 280 с.

78. Server and Cloud Computing [Электронный ресурс] – Режим доступа: <http://www.microsoft.com> (дата обращения 23.02.2019).

79. Service-Oriented Architecture. [Электронный ресурс] // Microsoft Developer Network. – Режим доступа: <http://msdn.microsoft.com/en-us/library/aa480021.aspx> (дата обращения: 12.03.2019)

80. Sourabh Sharma, Rajesh RV, David Gonzalez. Microservices: Building Scalable Software. – Packt Publishing Limited, 2017. – 919 с.

81. UpCloud, Docker Swarm vs. Kubernetes: Comparison of the Two Giants in Container Orchestration [Электронный ресурс]. – <https://www.upcloud.com/blog/docker-swarm-vs-kubernetes/> (дата обращения: 12.03.2019).

82. Vikram Murugesan, Microservices Deployment Cookbook. – Packt Publishing, 2017. – 378 с.

83. Viktor Farcic. The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices. – Leanpub, 2016. – 430 c.