ГОСУДАРСТВЕННЫЙ КОМИТЕТ СВЯЗИ, ИНФОРМАТИЗАЦИИ И ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

На правах рукописи

УДК 004.056.5

АСТАШЕВА ЕВГЕНИЯ АЛЕКСАНДРОВИЧА

Решение вопросов безопасности и защиты информации в частном “облаке” образовательного учреждения

5А330201 – Компьютерные системы и их программное обеспечение

ДИССЕРТАЦИЯ

На соискание академической степени магистра

Научный руководитель

Рахимов Д.К.

Ташкент-2014

ГОСУДАРСТВЕННЫЙ КОМИТЕТ СВЯЗИ, ИНФОРМАТИЗАЦИИ И ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ РЕСПУБЛИКИ УЗБЕКИСТАН
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Факультет: Программный инжиниринг
Кафедра: «Информационные технологии и программное обеспечение»
Учебный год: 2013-2014

Студент: Асташев Е. А.
Научный руководитель: Рахимов Д.К.
Специальность: 5А330201- «Компьютерные системы и их программное обеспечение»

АННОТАЦИЯ МАГИСТЕРСКОЙ ДИССЕРТАЦИИ

Актуальность темы исследования. Ввиду появления новых компьютерных технологий актуальной проблемой остается обеспечение безопасности в современных компьютерных системах. Таким образом, актуальным является исследование и разработка решений вопросов безопасности частного облака образовательного учреждения.

Цель и задачи исследования. Целью диссертации является разработка мер и действий по обеспечению безопасности облачного вычислительного центра при Ташкентском Университете Информационных технологий.

Задачами исследования являются:

- выявление требований и задач предъявляемых к частному облаку образовательного учреждения;
- анализ существующих облачных технологий, их видов, типологии, уровней доступа, а также выделение основных признаков и создание правил и методов для обеспечения повышенной безопасности;
- проведение работ по администрированию компьютерных систем и реализация алгоритма и программного обеспечения в соответствии с выделенными требованиями к разрабатываемой системе;

Объект и предмет исследования. Объектом исследования являются компьютерные сети и системы, а предмет проведение исследования – сервис использующий созданную программную оболочку, предоставляющий доступ к ресурсам компьютерно-вычислительной системе, а также сценарии по настройке безопасности таких систем.

Методы и средства исследования. Были применены такие методы исследования: моделирование, доказательство, анализ. Использовались: информационные, математические и логические средства исследования.

Гипотеза исследования. Возможность повышения безопасности данных за счет использования результатов диссертации при построении облачных систем.

Научная новизна работы заключается в том, что исследовалось новое направление распределенных компьютерных систем и повышение их безопасности, а также при разработке программного обеспечения применялись новые методы и средства разработки программных приложений, и криптографические стандарты РУз, которые на данный момент малоизучены.

Научная и практическая значимость. Результаты исследования могут быть использованы при построении облачных систем, а разработанная программа может использоваться для безопасного хранения данных в облачных системах.

Состав диссертационной работы. Магистерская диссертационная работа состоит из введения, трех глав, заключения, списка литературы, общим объемом в 81 лист и двух приложений.

Выводы и предложения. Диссертационная работа отображает особенности разрабатываемой системы и требования к ней и подходы позволяющие выбрать уровень виртуализации системы при построении. На основании требовании и различного уровня спецификаций разработан алгоритм по организации безопасности в частном облаке образовательного учреждения. Также создан прототип частного облака, типа сервиса предоставления программного обеспечения как услуги. Также была создана система решающий круг поставленных проблем.

Предложением по дальнейшему применению результатов исследования является использование их при организации гомогенных вычислений.

Научный руководитель: _____ Рахимов Д.К.

Студент магистратуры: _____ Асташев Е.А.

STATE COMMITTEE FOR COMMUNICATIONS, INFORMATIZATION AND TELECOMMUNICATION TECHNOLOGIES

OF THE REPUBLIC OF UZBEKISTAN

TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES

Faculty: Software Engineering
Department: Software of Information Technologies
Academic year: 2013-2014

Undergraduate: Astashev E.A.
Research supervisor: Rakhimov D.K.
Specialty: 5A330201«Computer systems and software»

THE SUMMARY TO THE MASTER THESIS

Relevance of the research. Due to the emergence of new computer, technology remains an urgent problem of security in modern computer systems. Thus, the current study is the development of solutions and private cloud security issues of the educational institution.

The purpose and objectives of the study. The main purpose of this master thesis is the development of measures and actions to ensure the security of the cloud-computing center at Tashkent University of Information Technologies.

The objectives of the study are:

• Identify needs and challenges imposed on the private cloud educational institution;

• Analysis of existing cloud technologies, their species, typology, access levels, as well as the selection and creation of the main features of the rules and methods for enhanced security;

• Work on administration of computer systems and the implementation of the algorithm and software in accordance with the requirements allocated to the system developed;

Research object and subject of study. Objects of study are computer networks and systems, and the study subject - service created using a software shell that provide access to the resources of the computer- computer system, as well as scripts for configuring the security of such systems.

Methods and tools for research. Were applied research methods: modeling, evidence analysis. As research, tools used: information, mathematical and logical research.

Hypothesis of the study. Possibility of increasing the security of the data using the results of the dissertation in the construction of cloud systems.

Scientific novelty and practical significance of the research. The results can used in the construction of cloud systems and developed software can used for safe storage in the cloud systems.

Scientific novelty. Of the work, lies in the fact that a new direction investigated distributed computer systems and improving their safety, as well as software development to use new methods and tools for developing software applications and cryptographic standards of Uzbekistan, which is currently little studied.

Structure and scope of work. Master's thesis consists of an introduction, three chapters, conclusion, scientific literature, totaling 81 pages and 2 annexes.

Conclusions and suggestions. The science work is developed system displays the features and requirements and approaches to it allowing you to select the level virtualization system in the construction. Based on the requirements and different levels of specifications developed an algorithm for the organization of security in a private cloud educational institution. Also created a prototype of a private cloud, such as providing service software as a service. Also established a system of deciding their issues.

Proposal for the further application of research results is to use them in the organization of homogeneous computing.

Research supervisor: _____ Rakhimov D.K.

Undergraduate: _____ Astashev E.A.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

XXI век является веком информационных технологий. Это выражается в интенсивном совершенствовании средств вычислительной техники и техники связи, появление новых и в дальнейшем развитие существующих информационных технологий, а также в реализации прикладных информационных систем. Достижения в этой сфере заняли достойное место в организационном управлении, промышленности, проведении научных исследований и автоматизированном проектировании. Информатизация охватила и социальную сферу: образование, науку, культуру, здравоохранение.

Информатизация стала общепризнанным фактором деловой и общественной жизни. Широкая распространенность и возросшая пропускная способность создают условия, при которых выгодно решать многие задачи при помощи информационных технологий.

На заседании Кабинета Министров 19 января 2012 года Президент Республики Узбекистана Ислам Абдуганиевич Каримов отметил, что в Узбекистане активно развивается информатизация общества. [1]

В соответствии с «Постановлением Президента РУз от 21.03.2012 г. за № ПП-1730 "О мерах по дальнейшему внедрению и развитию современных информационно-коммуникационных технологий"» более широкое развитие и внедрение получили современные компьютерные системы, в частности развиваются облачные системы вычисления и хранения данных. Актуальным является в данном случае вопрос о безопасном хранении и передаче и использовании существующей информации. Традиционные системы требуют знания пароля, наличия ключа, идентификационной карточки, либо иного идентифицирующего предмета, который можно забыть или потерять. Но при правильном подходе защиты информации можно увеличить и поддерживать уровень безопасности на высоком уровне. И именно обеспечить данный уровень и стараются обеспечить облачные сервисы. Также «облака» помогают

осуществлять доступность к информации, что является одним из основных признаков развития информационного поля.

Актуальность темы исследования. Ввиду появления новых компьютерных технологий актуальной проблемой остается обеспечение безопасности в современных компьютерных системах. Таким образом, актуальным является исследование и разработка решений вопросов безопасности частного облака образовательного учреждения.

Объект и предмет исследования. Объектом исследования являются компьютерные сети и системы, а предмет проведение исследования – сервис использующий созданную программную оболочку, предоставляющий доступ к ресурсам компьютерно-вычислительной системе, а также сценарии по настройке безопасности таких систем.

Цель и задачи исследования. Целью диссертации является разработка мер и действий по обеспечению безопасности облачного вычислительного центра при Ташкентском Университете Информационных технологий.

Задачами исследования являются:

- выявление требований и задач предъявляемых к частному облаку образовательного учреждения;
- анализ существующих облачных технологий, их видов, типологии, уровней доступа, а также выделение основных признаков и создание правил и методов для обеспечения повышенной безопасности;
- проведение работ по администрированию компьютерных систем и реализация алгоритма и программного обеспечения в соответствии с выделенными требованиями к разрабатываемой системе;

Гипотеза исследования. Возможность повышения безопасности данных за счет использования результатов диссертации при построении облачных систем.

Обзор использованной литературы. В работе были использованы научные работы следующих ученых: М. Талллоч (специализируется в области организации работы серверных операционных систем), В.Станек (специализируется в области автоматизации процессов), Дж. Гудакре (специализируется в области работы гипервизоров и систем виртуализации); В работе были использованы и работы других ученых, подробный список которых указан в списке использованной литературы.

Методы и средства исследования. Были применены такие методы исследования: моделирование, доказательство, анализ. Использовались: информационные, математические и логические средства исследования.

Научная и практическая значимость. Результаты исследования могут быть использованы при построении облачных систем, а разработанная система может использоваться для безопасного хранения данных в облачных системах.

Научная новизна работы заключается в том, что исследовалось новое направление распределенных компьютерных систем и повышение их безопасности, а также при разработке программного обеспечения применялись новые методы и средства разработки программных приложений, и криптографические стандарты РУз, которые на данный момент малоизучены.

Состав диссертационной работы. Магистерская диссертационная работа состоит из введения, трех глав, заключения, списка литературы, общим объемом в 81 листа и двух приложений. Работа также включает в себя наличие, рисунков в количестве 24 (двадцати четырёх) штук и таблиц в количестве 2 (двух) штук.

ГЛАВА I. АНАЛИЗ СУЩЕСТВУЮЩИХ ВИДОВ ВИРТУАЛИЗАЦИИ И СИСТЕМ ОРГАНИЗАЦИИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

1. Разновидность облачных вычислений

На сегодняшний день стал актуальным вопрос облачных вычислений, хранения данных в облаке, построение систем облачного вычисления с использованием гипервизоров и т.п. Термин облако появилось в 2008 году, где впервые было использовано Эриком Шмидтом (одним из основателей Google), представляет собой метафору, скрывающую сложную архитектуру инфраструктуру и технические детали по созданию и поддержки таких систем. На современном IT рынке в мире и Узбекистане данная технология является быстро развивающейся и важной для общества. Согласно исследованию компании Gartner, к 2016 году мировой объем рынка сервисов облачных технологий достигнет 206,6 млрд. долларов.

Следует сказать, что сутью использования облачных технологий является то, что конечный пользователь получает возможность использовать вычислительные ресурсы, удаленный доступ к услугам компании или провайдера, платформам и программам через Интернет.

Существуют четыре типа модели развертывания облака:

- частное облако – представляет собой инфраструктуру, которая используется зачастую одной организацией и сотрудниками данной организации.
- публичное облако – представляет собой инфраструктуру, предназначено и используется большим количеством людей не всегда относящихся к какой-то определенной организации.
- общественное облако – представляет собой инфраструктуру, которая используется каким-то определенным кругом людей для

решения общих проблем (как правило, такой вид облака используется в организациях, где требуется построить обмен данными между двумя отделами, тесно работающими друг с другом).

- гибридное облако – представляет собой комбинацию нескольких облачных инфраструктур, которые являются уникальными объектами, но связанны между собой стандартными или уникальными технологиями.

На основе модели развертывания применяют три основных вида модели обслуживания:

SaaS (от *англ.* Software as a Service), (пер. Программное обеспечение как услуга) – данная модель, предоставляет возможность конечному потребителю использовать прикладное программное обеспечение провайдера, работающего в облачной среде и предоставляющий доступ с различных клиентских устройств интерфейса программы. Управление и контроль инфраструктуры облака осуществляется провайдером предоставляющим, в том числе сети, сервера, операционные системы, системы хранения, и даже индивидуальные возможности приложения.

PaaS (от англ. Platform as a Service), (пер. Платформа как услуга) – такая модель предоставляет потребителю использование облачной инфраструктуры как пространство для размещения программного обеспечения, в дальнейшем предусматривая размещение новых приложений которые могут быть как собственного производства, так и приобретённых приложений. Такие платформы содержат инструменты разработки, отладки, тестирования, выполнения программного обеспечения, промежуточное программное обеспечение, среды для компиляции и запуска кода, а также системы управления базами данных, которые предоставляются поставщиком услуг.

IaaS (от *англ.* Infrastructure as a Service), (пер. Инфраструктура как услуга) – такая модель даёт большое поле деятельности для потребителя услуг, предоставляя последнему использовать облачную инфраструктуру для самостоятельного управления ресурсами системы, обработки и хранения данных, управление сетями и т.п. к примеру, потребитель услуги может запускать и устанавливать любое программное обеспечение кроме того он может использовать в облаке такого типа операционные системы, а также требуемое для его работы программное обеспечение. Потребитель может настроить систему безопасности по требуемому уровню для организации осуществлять контроль физических и виртуальных систем, систем хранения данных и установленных приложения, а также вводить определенные ограничения для конкретного пользователя или группа сервиса или набора доступных сервисов (например, межсетевой экран, DNS). Контроль, обслуживание и управление инфраструктурой облака, в том числе сети, серверов, операционных систем, хранения осуществляется облачным провайдером, за исключением тех приложений (разработанных или установленных потребителем), а также, по возможности, параметров конфигурации среды (платформы).

2. Организация облачной инфраструктуры

Для организации облачной инфраструктуры следует выполнить достаточно большой этап подготовки материально-технических средств и разработать комплекс организационных методик для предоставления требуемых условий безопасности.

Организация облачной инфраструктуры, с технической стороны, организуется путем виртуализации физических машин и объединением их в вычислительный кластер.

Надо учитывать, что виртуализация бывает двух типов:

- аппаратная виртуализация;
- программная виртуализация;

Критерии для виртуальных машин были сформулированы в работе 1974 г. Жеральда Попека и Роберта Голдберга «Formal requirements for virtualizable third generation architectures» [12]. Для рассмотрения будем использовать упрощённое представление компьютера, состоящего из одного центрального процессора и оперативной памяти.

Выдвигаемые критерии к виртуальным машинам (ВМ):

1. изоляция — каждая виртуальная машина должна иметь доступ только к тем ресурсам, которые были ей назначены.
2. эквивалентность — любая программа, исполняемая под управлением ВМ, должна демонстрировать поведение, полностью идентичное её исполнению на реальной системе, за исключением эффектов, вызванных двумя обстоятельствами: различием в количестве доступных ресурсов и длительностями операций.
3. эффективность — большинство инструкций должны симулироваться в режиме прямого исполнения.

В случае симуляторов, основанных на интерпретации инструкций, условие эффективности не выполняется, т.к. каждая инструкция гостя требует обработки симулятором.

Аппаратная виртуализация

Родоначальником в данной сфере была фирма IBM. Аппаратная виртуализация представляла интерес ещё до появления микропроцессоров, во времена преобладания больших систем, ресурсы которых были дорогостоящими, и их простаивание было экономически недопустимо.

Виртуализация избавляла пользователей и прикладных программистов от необходимости изменять своё ПО, так как виртуальная машина была идентична физической, а также позволяла повысить степень утилизации таких систем.

Классы инструкций

Процессор поддерживает два режима работы: режим супервизора, используемый операционной системой, и режим пользователя, в котором исполняются прикладные приложения. Память поддерживает режим сегментации, используемый для организации виртуальной памяти.

Состояние процессора содержит минимум три регистра [11]: $\boldsymbol{M}$, определяющий, находится ли он в режиме супервизора $\boldsymbol{S}$ или пользователя $\boldsymbol{U}$, $\boldsymbol{P}$ — указатель текущей инструкции и $\boldsymbol{R}$ — состояние, определяющее границы используемого сегмента памяти (в простейшем случае $\boldsymbol{R}$ задаёт отрезок, т.е. $\boldsymbol{R} = (\boldsymbol{1}, \boldsymbol{b})$, где l — адрес начала диапазона, b — его длина).

Машинные инструкции рассматриваемого процессора можно классифицировать по трём категориям:

- Привилегированные (*англ.* privileged). Инструкции, исполнение которых с $\boldsymbol{M} = \boldsymbol{U}$ всегда вызывает ловушку потока управления. Другими словами, такая инструкция может исполняться только в режиме супервизора, иначе она обязательно вызывает исключение.
- Служебные. Класс состоит из двух подклассов.

  1. инструкции, исполнение которых закончилось без ловушки защиты памяти и вызвало изменение $\boldsymbol{m}$ и/или $\boldsymbol{r}$. они могут менять режим процессора из супервизора в пользовательский или обратно или изменять положение и размер доступного сегмента памяти.

  2. инструкции, поведение которых в случаях, когда они не вызывают ловушку защиты памяти, зависят или от режима $\boldsymbol{M}$, или от значения $\boldsymbol{R}$.

- Безвредные (*англ.* innocuous). Не являющиеся служебными. Самый широкий класс инструкций, не манипулирующие ничем, кроме указателя инструкций $\boldsymbol{P}$ и памяти $\boldsymbol{E}$, поведение которых не зависит

от того, в каком режиме или с каким адресом в памяти они расположены.

При исполнении каждая инструкция $\boldsymbol{i}$ в общем случае может изменить как $\boldsymbol{(M,\ P,\ R)}$, так и память $\boldsymbol{E}$, т.е. она является функцией преобразования: $(\boldsymbol{M1}, \boldsymbol{P1}, \boldsymbol{R1}, \boldsymbol{E1}) -> (\boldsymbol{M2}, \boldsymbol{P2}, \boldsymbol{R2}, \boldsymbol{E2})$.

Считается, что для некоторых входных условий инструкция вызывает исключение (ловушки), если в результате её исполнения содержимое памяти не изменяется, кроме единственной ячейки $\boldsymbol{E}[\boldsymbol{0}]$, в которую помещается предыдущее состояние процессора $(\boldsymbol{M1}, \boldsymbol{P1}, \boldsymbol{R1})$. Новое состояние процессора $(\boldsymbol{M2}, \boldsymbol{P2}, \boldsymbol{R2})$ при этом копируется из $\boldsymbol{E}[\boldsymbol{1}]$. Другими словами, ловушка позволяет сохранить полное состояние программы на момент до начала исполнения её последней инструкции и передать управление обработчику, в случае обычных систем обычно работающему в режиме супервизора и призванного обеспечить дополнительные действия над состоянием системы, а затем вернуть управление в программу, восстановив состояние из $\boldsymbol{E}[\boldsymbol{0}]$.

Далее, ловушки могут иметь два признака.

1. вызванные попыткой изменить состояние процессора (ловушка потока управления).
2. обращения к содержимому памяти, выходящему за пределы диапазона, определённого в (ловушка защиты памяти).

Следует отметить, что результатом исполнения могут быть одновременно ловушка потока управления и защиты памяти.

Достаточное условие построения гипервизора ВМ

Соблюдение трёх сформулированных выше условий возможности построения гипервизора виртуальных машин: множество служебных инструкций является подмножеством привилегированных инструкций (Рис. 1.1)[10,12]. Соблюсти оптимальный баланс между тремя свойствами: изоляцией, эквивалентностью и эффективностью, — является

невозможным, из-за необходимости программного контроля за исполнением ими служебных, но не привилегированных инструкций, так как сама аппаратура не обеспечивает этого (Рис. 1.2).

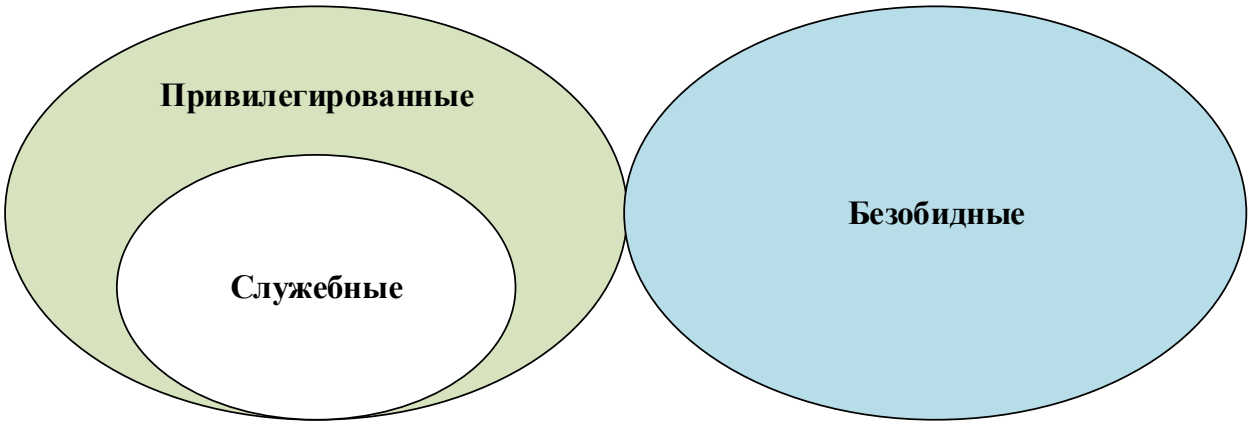


Рис. 1.1. Выполнение условия виртуализации. Множество служебных инструкций является подмножеством привилегированных

Даже единственная такая инструкция, исполненная напрямую ВМ, угрожает стабильной работе гипервизора, и поэтому гипервизор вынужден сканировать весь поток гостевых инструкций. Чаще всего расплачиваться приходится скоростью работы виртуальных машин.

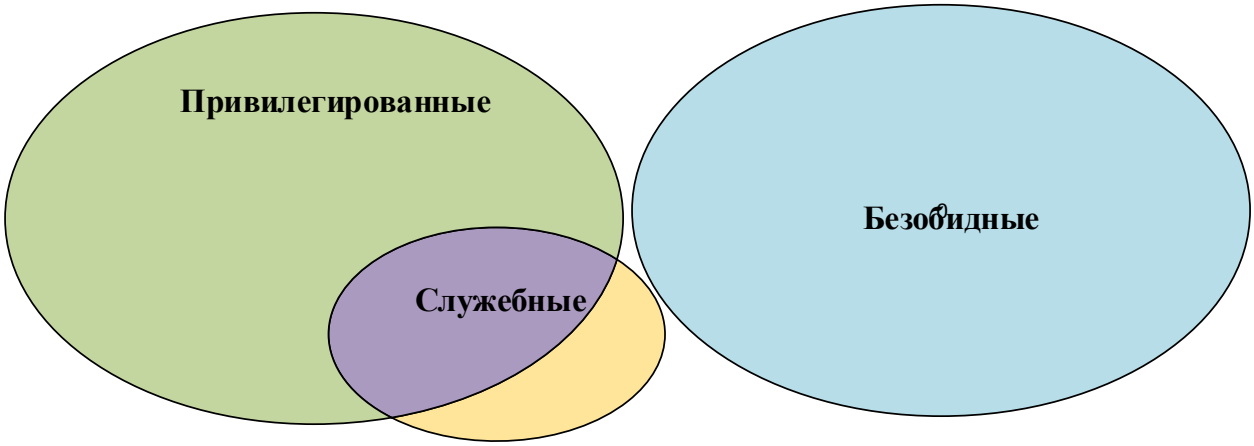


Рис. 1.2. Невыполнение условия виртуализации. Служебные, но не привилегированные инструкции требуют реализации сложной логики.

Структура гостевых программ

Для эффективной работы программ внутри ВМ необходимо, чтобы большая часть команд являлись безвредными. Как правило, это верно для прикладных приложений. Операционные системы, в свою очередь, предназначены для управления ресурсами системы, что подразумевает использование ими привилегированных и служебных инструкций, поэтому приходится их перехватывать и интерпретировать что соответственно дает падение производительности. В идеале в наборе инструкций должно быть, как можно меньше привилегированных и служебных инструкции для того, чтобы частота возникновения ловушек была минимальной[14].

Периферия

Поскольку периферийные устройства являются служебным ресурсом, очевидно, что для обеспечения условий изоляции и эквивалентности необходимо, чтобы все попытки доступа к ним были контролируемы гипервизором ВМ так же, как они контролируются в многозадачной операционной системе её ядром. В настоящее время доступ к устройствам производится через механизм отражения их в физической памяти системы (*англ*. memory mapped I/O)[14].

В зависимости от класса устройств периферии и от приложений с ними взаимодействующими может происходить замедление в работе при виртуализации. Различают следующие классы устройств:

- выделенное устройство — устройство, доступное исключительно внутри одной гостевой системы. примеры: клавиатура, гипервизор.
- разделяемое — общее для нескольких гостей. такое устройство или имеет несколько частей, каждая из которых выделена для нужд одного из них (*англ.* partitioned mode), например, жёсткий диск с несколькими разделами, или подключается к каждому из них поочерёдно (*англ.* shared mode). пример: сетевая карта.

- полностью виртуальное — устройство, отсутствующее в реальной системе (или присутствующее, но в ограниченном количестве) и моделируемое программно внутри гипервизора. Примеры: таймеры прерываний — каждый гость имеет собственный таймер, несмотря на то, что в хозяйской системе есть только один, и он используется для собственных нужд гипервизора.

Многопроцессорные системы

На сегодняшний день практически все современные компьютеры содержат в себе более одного ядра или процессора. Кроме того, внутри одного гипервизора могут исполняться несколько ВМ, каждая из которых может иметь в своём распоряжении несколько виртуальных процессоров. Рассмотрим, как это влияет на условия виртуализации. Здесь зачастую используется гипервизор. Гипервизор это программа находящаяся на реальной физической машине и осуществляющая ретрансляцию и выполнение команд виртуальных машин [15].

Аппаратная синхронизация в условиях виртуализации

Необходимо обратить внимание на выполнение условий эффективности работы многопоточных приложений внутри ВМ. В отличие от однопоточных, для них характерны процессы синхронизации частей программы, исполняющихся на различных виртуальных процессорах. При этом все участвующие потоки ожидают, когда все они достигнут заранее определённой точки алгоритма, т.н. барьера. В случае виртуализации системы один или несколько гостевых потоков могут оказаться неактивными, вытесненными гипервизором, из-за чего остальные будут попусту тратить время.

Примером такого неэффективного поведения гостевых систем является синхронизация с задействованием циклических блокировок (*англ.* spin lock) внутри ВМ[9]. Будучи неэффективной и поэтому неиспользуемой для однопроцессорных систем, в случае нескольких

процессоров она является легковесной альтернативой другим, более тяжеловесным замкам (*англ.* lock), используемым для входа в критические секции параллельных алгоритмов. Чаще всего они используются внутри операционной системы, но не пользовательских программ, так как только ОС может точно определить, какие из системных ресурсов могут быть эффективно защищены с помощью циклических блокировок. Однако в случае виртуальной машины планированием ресурсов на самом деле занимается не ОС, а гипервизор ВМ, который в общем случае не осведомлён о них и может вытеснить поток, способный освободить ресурс, тогда как второй поток будет выполнять циклическую блокировку, бесполезно тратя процессорное время.

Оптимальным решением при этом является деактивация заблокированного потока до тех пор, пока нужный ему ресурс не освободится.

Существующие решения для данной проблемы описаны ниже.

1. гипервизор ВМ может пытаться детектировать использование циклических блокировок гостевой ос. это требует анализа кода перед исполнением, установки точек останова по адресам замка. способ не отличается универсальностью и надёжностью детектирования.
2. гостевая система может сигнализировать гипервизору о намерении использовать циклическую блокировку с помощью специальной инструкции. способ более надёжный, однако требующий модификации кода гостевой ОС.

Прерывания

Прерывания являются механизмом оповещения процессора о событиях внешних устройств, требующих внимания операционной системы. В случае использования виртуальных машин гипервизор должен иметь возможность контролировать доставку прерываний, так как часть

или все из них необходимо обрабатывать именно внутри гипервизора. Например, прерывание таймера может быть использовано им для отслеживания/ограничения использования гостями процессорного времени и для возможности переключения между несколькими одновременно запущенными ВМ. Кроме того, в случае нескольких гостей заранее неясно, какому из них следует доставить прерывание, и принять решение должен гипервизор.

Простейшее решение, обеспечивающее изоляцию, — это направлять все прерывания в гипервизор ВМ. Эквивалентность при этом будет обеспечиваться им самим: прерывание при необходимости будет доставлено внутрь гостя через симуляцию изменения его состояния. Гипервизор может дополнительно создавать виртуальные прерывания, обусловленные только логикой его работы, а не внешними событиями. Однако эффективность такого решения не будет оптимальной. Как правило, реакция системы на прерывание должна произойти в течение ограниченного времени, иначе она потеряет смысл для внешнего устройства или будет иметь катастрофические последствия для системы в целом. Более эффективным является аппаратный контроль за доставкой прерываний, позволяющий часть из них сделать безвредными для состояния системы и не требовать каждый раз вмешательства программы гипервизора. Наконец, следует отметить, что схемы доставки и обработки прерываний в системах с несколькими процессорами также более сложны, и это приходится учитывать при создании гипервизора ВМ для таких систем, при этом его эффективность может оказаться ниже, чем у однопроцессорного эквивалента.

Преобразование адресов

Модель машинных инструкций, использованная ранее для формулировки утверждения об эффективной виртуализации, использовала простую линейную схему трансляции адресов, основанную на

сегментации. Она является вычислительно простой, не изменяется при введении гипервизора ВМ, и поэтому анализа влияния механизма преобразования адресов на эффективность не производилось.

В настоящее время механизмы страничной виртуальной памяти и применяют нелинейное преобразование виртуальных адресов пользовательских приложений в физические адреса, используемые аппаратурой. Участвующий при этом системный ресурс — регистр-указатель адреса таблицы преобразований (чаще всего на практике используется несколько таблиц, образующих иерархию, имеющую общий корень). В случае работы ВМ этот указатель необходимо виртуализировать, так как у каждой гостевой системы содержимое регистра своё, как и положение/содержимое таблицы. Стоимость программной реализации этого механизма внутри гипервизора высока, поэтому приложения, активно использующие память, могут терять в эффективности при виртуализации.

Для решения этой проблемы используется двухуровневая аппаратная трансляция адресов (Рис. 1.3). Гостевые ОС видят только первый уровень, тогда как генерируемый для них физический адрес в дальнейшем транслируется вторым уровнем в настоящий адрес.

Преобразование адресов для периферийных устройств

Кроме процессоров к оперативной памяти напрямую могут обращаться и периферийные устройства — с помощью технологии DMA (*англ.* Direct Memory Access). При этом обращения в классических системах без виртуализации идёт по физическим адресам. Очевидно, внутри виртуальной машины необходимо транслировать такие адреса, и это превращается в накладные расходы и понижение эффективности гипервизора. Решение состоит в использовании устройства IOMMU (*англ.* Input output memory management unit), позволяющего контролировать обращения пользовательских устройств к физической памяти.

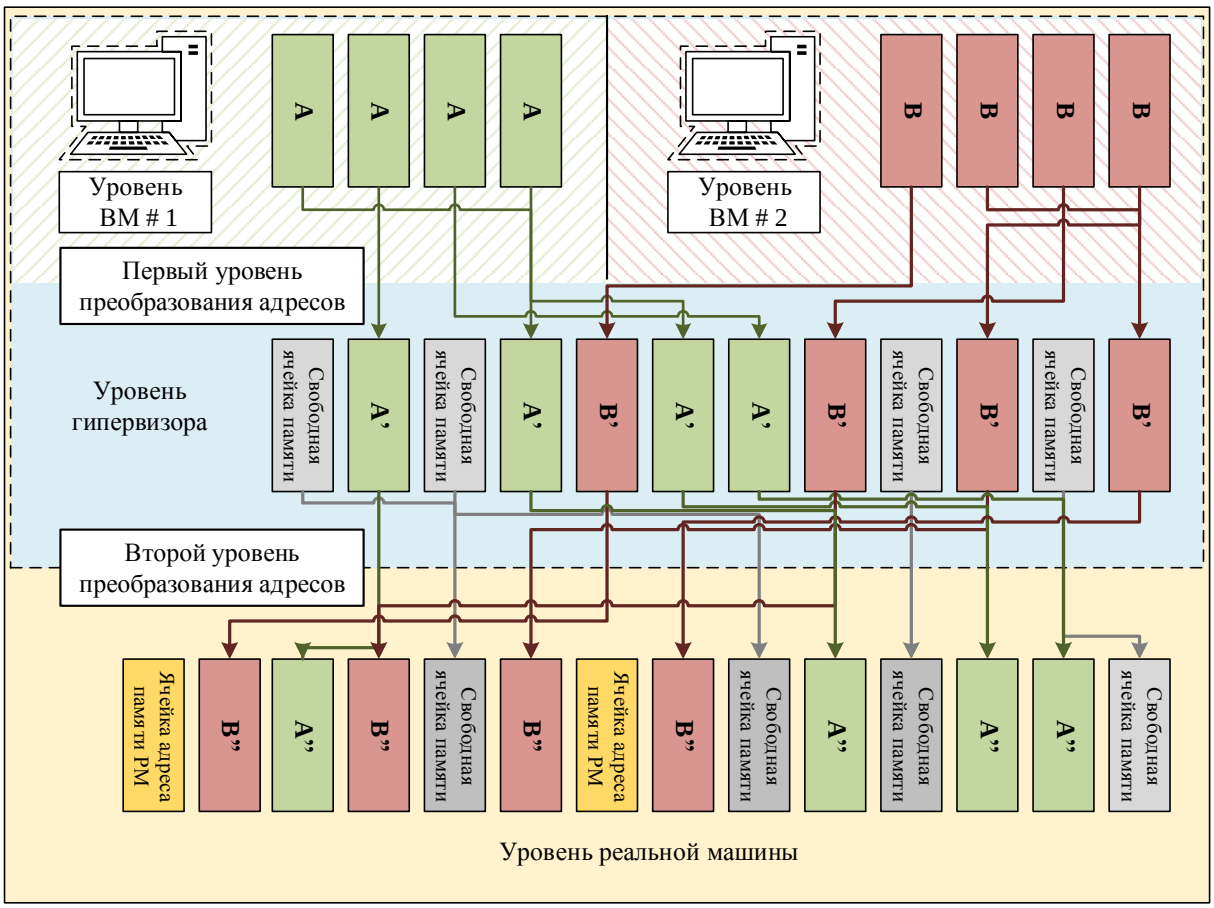


Рис. 1.3. Двухуровневая трансляция адресов. Первый уровень контролируется гостевыми ОС, второй — гипервизором виртуальных машин

Расширение принципа

Заменим слово «инструкция» на «операция» для того чтобы расширить условие виртуализации тогда множество служебных операций является подмножеством привилегированных**.** При этом под операцией подразумевается любая архитектурно определённую активность по чтению

или изменению состояния системы, в том числе инструкции, прерывания, доступы к устройствам, преобразования адресов и т.п.

При этом условие повышения эффективности виртуализации будет звучать следующим образом: в архитектуре системы должно присутствовать минимальное число служебных операций. Достигать его можно двумя способами: переводя служебные инструкции в разряд безвредных или уменьшая число привилегированных. Для этого большинство архитектур пошло по пути добавления в регистр состояния **M** нового режима **r** — режима гипервизора ВМ (*англ.* root mode). Он соотносится с режимом ***S*** так, как ***S*** — с ***U***; другими словами, обновлённый класс привилегированных инструкций теперь вызывает ловушку потока управления, переводящую процессор из ***S*** в ***r***. Для обеспечение эффективной виртуализации требуется:

Уменьшение частоты и выходов в режим гипервизора с помощью пред просмотра инструкций

Частые прерывания работы виртуальной машины из-за необходимости выхода в гипервизор негативно влияют на скорость симуляции. Прямое исполнение с использованием виртуализации оказывается неэффективным, для увеличения производительности имеет смысл переключиться на другую схему работы, например, на интерпретацию или двоичную трансляцию.

Рекурсивная виртуализация

Когда гипервизор виртуальных машин запускается под управлением другого гипервизора, непосредственно исполняющегося на аппаратуре, называется рекурсивной виртуализацией (Рис.1.4.). Теоретически она может быть не ограничена только двумя уровнями — внутри каждого гипервизора ВМ может исполняться следующий, тем самым образуя иерархию гипервизоров.

Рассмотрим одно из затруднений, связанных со спецификой вложенного запуска гипервизоров ВМ — обработку ловушек и прерываний. В простейшем случае за обработку всех типов исключительных ситуаций всегда отвечает самый внешний гипервизор, задача которого — или обработать событие самостоятельно, тем самым «спрятав» его от остальных уровней, или передать его следующему. Как для прерываний, так и для ловушек это часто оказывается неоптимальным — событие должно пройти несколько уровней иерархии, каждый из которых внесёт задержку на его обработку.

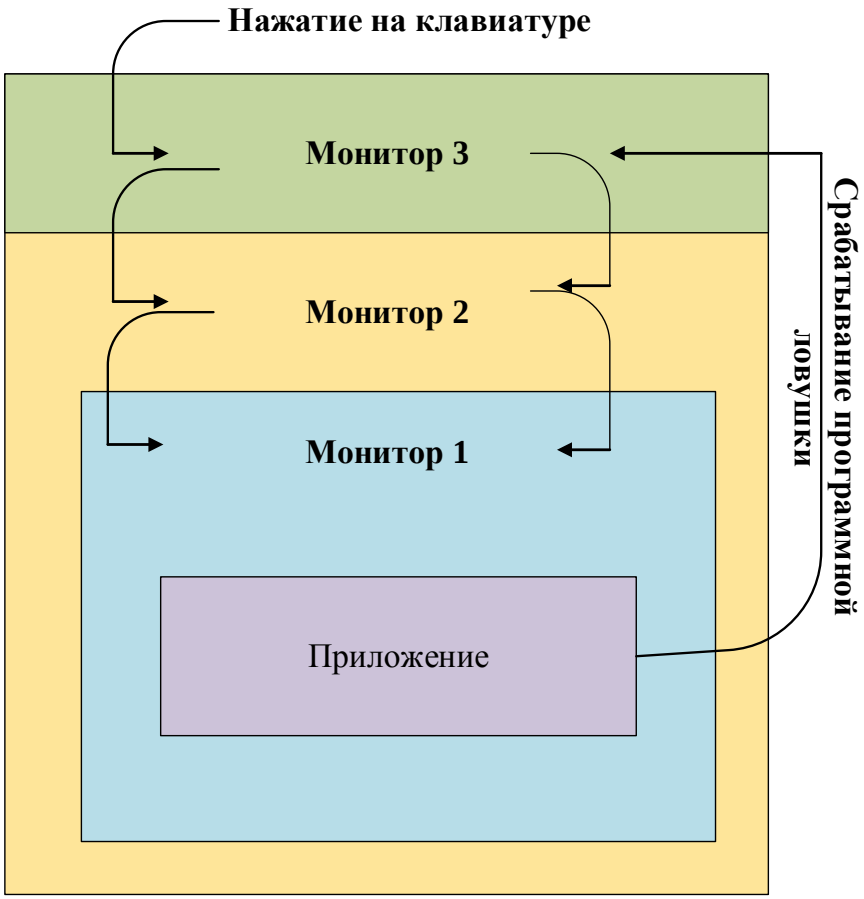


Рис. 1.4. Рекурсивная виртуализация. Все события должны обрабатываться внешним гипервизором, который спускает их вниз по иерархии, при этом формируется задержка

На Рис. 1.4 показана обработка двух типов сообщений — прерывания, возникшего во внешней аппаратуре, и ловушки потока управления, случившейся внутри приложения.

Для оптимальной обработки различных типов ловушек и прерываний для каждого из них должен быть выбран уровень иерархии гипервизоров ВМ, и при возникновении события управление должно передаваться напрямую этому уровню, минуя дополнительную обработку вышележащими уровнями и без связанных с этим накладных расходов процессорного времени.

Следует отметить, что реализация аппаратной виртуализации позволяет ускорить процесс обработки данных разбивая обработку на несколько потоков.

Программная виртуализация

Программная виртуализация (Рис.1.5.) – предоставление части из набора вычислительных ресурсов или их логического объединения, отделенное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию вычислительных процессов нескольких ВМ, выполняемых на одном физическом ресурсе. Данные подходы отличаются по количеству одновременных разделов, масштабируемости, производительности и разнообразию поддерживаемых операционных систем. Следует сказать, что программная виртуализация осуществляет эмуляцию работы реальной машины. При программной виртуализации уровень гипервизора заменяется уровнем хостовой операционной системы, т.к. именно она осуществляет контроль по выделению ресурсов. Данный тип виртуализации позволяет эмулировать не только поведение периферии через специальное API, но также позволяет организовать виртуальную среду для безопасного исполнения приложений.

Плюсы программной виртуализации:

- скорость работы виртуальных машин – создание VPS, переустановка ОС, загрузка сервера и тому подобные операции занимают секунды.
- экономия ресурсов нода — ядро загружается один раз и используется всеми VPS,
- стоимость VPS основанной на программной виртуализации ниже, чем стоимость VPS на базе технологий основанной на аппаратной виртуализации.

Минусы программной виртуализации:

- недостаточно жесткое разделение ресурсов
- возможность оверселлинга.

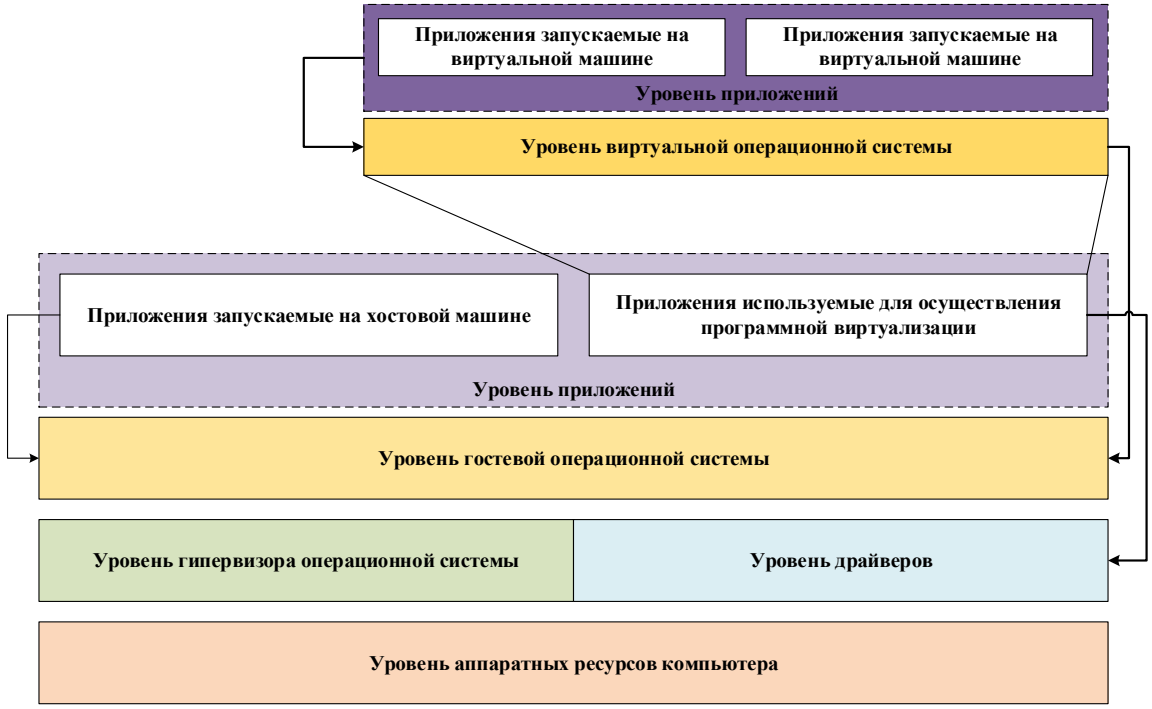


Рис. 1.5. Пример реализации программной виртуализации

Существуют следующие технологии программной виртуализации:

- полная виртуализация
- паравиртуализация
- виртуализация на уровне ядра ОС
- Виртуализация приложений

Полная виртуализация

При полной виртуализации (Рис.1.6.) используются не модифицированные экземпляры гостевых операционных систем, а для поддержки работы этих ОС служит общий слой эмуляции их исполнения поверх хостовой ОС, в роли которой выступает обычная операционная система.

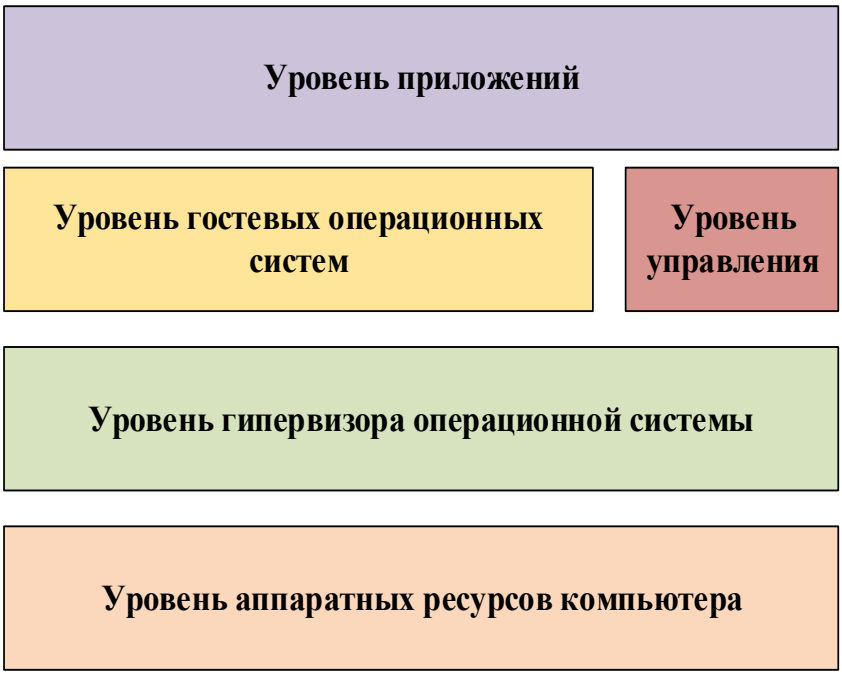


Рис. 1.6. Полная виртуализация

К достоинствам данного подхода можно причислить:

- относительную простоту реализации, универсальность и надежность решения;
- все функции управления берет на себя хост-ОС.

К недостаткам данного подхода можно причислить:

- высокие дополнительные накладные расходы на используемые аппаратные ресурсы,
- отсутствие учета особенностей гостевых ОС,
- меньшая гибкость в использовании аппаратных средств.

Паравиртуализация

При паравиртуализации (Рис.1.7.) модификация ядра гостевой ОС выполняется таким образом, что в нее включается новый набор API, через который она может напрямую работать с аппаратурой, не конфликтуя с другими виртуальными машинами. При этом нет необходимости задействовать полноценную ОС в качестве хостового ПО, функции которого в данном случае исполняет специальная система, получившая название гипервизора.

Достоинства данной технологии заключаются:

- в отсутствии потребности в хостовой ОС – ВМ
- эффективное использование аппаратных ресурсов.

Недостатки данной технологии заключаются:

- в сложности реализации подхода
- необходимости создания специализированной ОС-гипервизора.

Именно этот вариант является сегодня наиболее актуальным направлением развития серверных технологий виртуализации и применяется в VMware ESX Server, Xen, Microsoft Hyper-V.

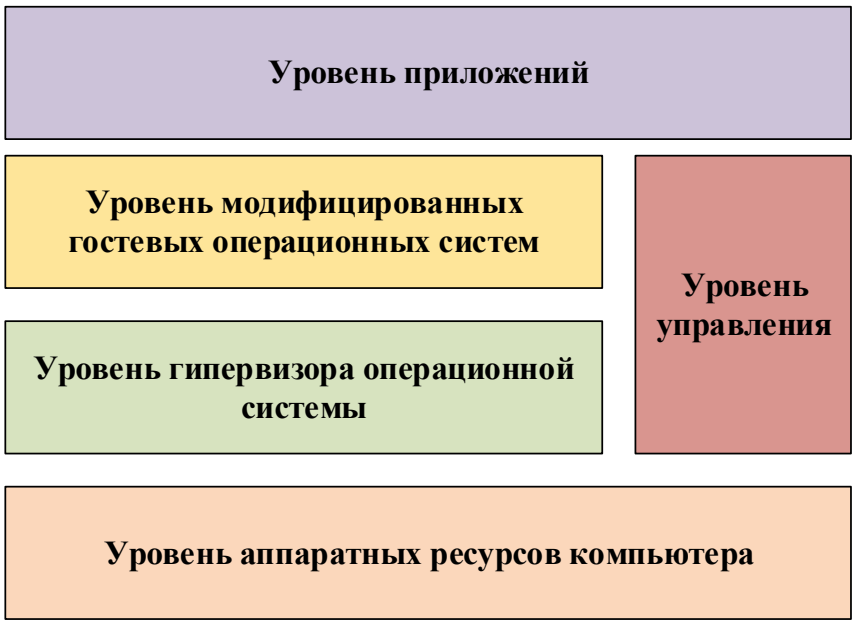


Рис. 1.7. Паравиртуализация

Виртуализация на уровне ядра ОС

Виртуализация на уровне ядра ОС (Рис. 1.8.) – этот вариант подразумевает использование одного ядра хостовой ОС для создания независимых параллельно работающих операционных сред. Для гостевого ПО создается только собственное сетевое и аппаратное окружение.

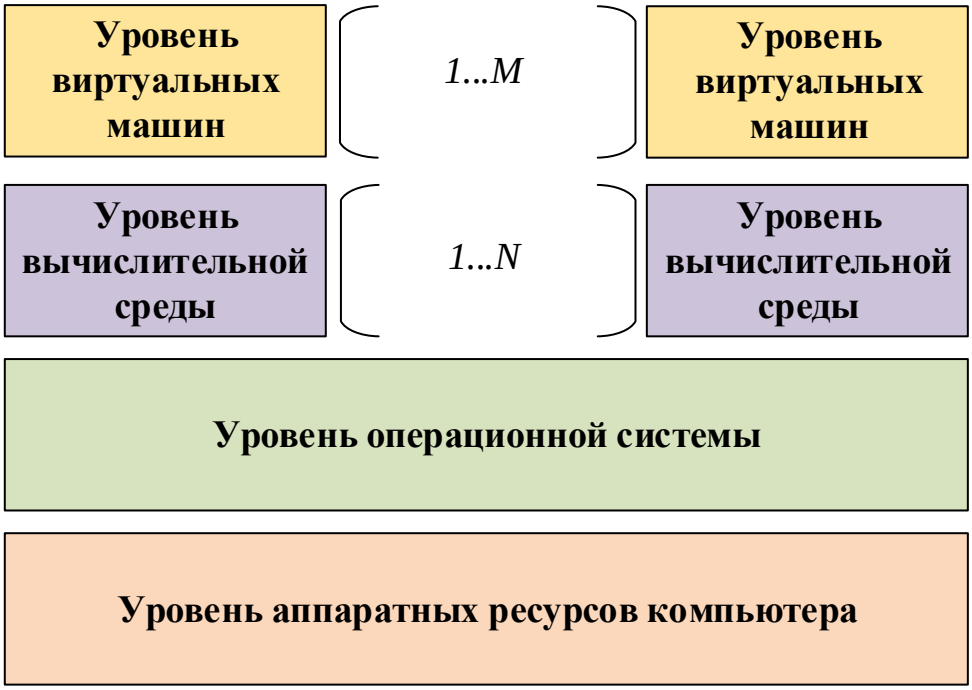


Рис. 1.8. Виртуализация на уровне ОС

Достоинства данной технологии заключаются:

- высокой эффективности использования аппаратных ресурсов,
- низких накладных технических расходах
- отличной управляемости
- минимизации расходов на приобретение лицензий.

недостатки данной технологии заключаются:

- способности реализовать только однородные вычислительные среды.

Виртуализация приложений

Персональный компьютер, ставший за последние десятилетия неотъемлемым атрибутом офиса и средством выполнения большинства офисных задач, перестает успевать за растущими потребностями пользователей. Реальным инструментом пользователя оказывается программное обеспечение, которое лишь привязано к ПК, делая его промежуточным звеном корпоративной информационной системы.

Рис. 1.9. Виртуализация приложений

С ростом масштабов организаций, использование в ИТ-инфраструктуре пользовательских ПК вызывает ряд сложностей:

- большие операционные издержки на поддержку компьютерного парка;
- сложность, связанная с управлением настольными ПК;
- обеспечение пользователям безопасного и надежного доступа к ПО и приложениям, необходимым для работы;
- техническое сопровождение пользователей;
- установка и обновление лицензий на ПО и техническое обслуживание;
- резервное копирование и т.д.

Уйти от этих сложностей и сократить издержки, связанные с их решением, возможно благодаря применению технологии виртуализации приложений (Рис.1.9.) подразумевает применение модели изоляции прикладных программ с управляемым взаимодействием с ОС, при котором производится виртуализация каждого экземпляра приложений, все его основные компоненты: файлы (включая системные), реестр, шрифты, INI-файлы, COM-объекты, службы.

Приложение исполняется без процедуры инсталляции в традиционном ее понимании. Такой подход имеет очевидные преимущества: ускорение развертывания приложений и возможность управления ими, сведение к минимуму не только конфликтов между приложениями, но и в отсутствии потребности тестирования приложений на совместимость в запускаемых средах (Рис.1.10.).

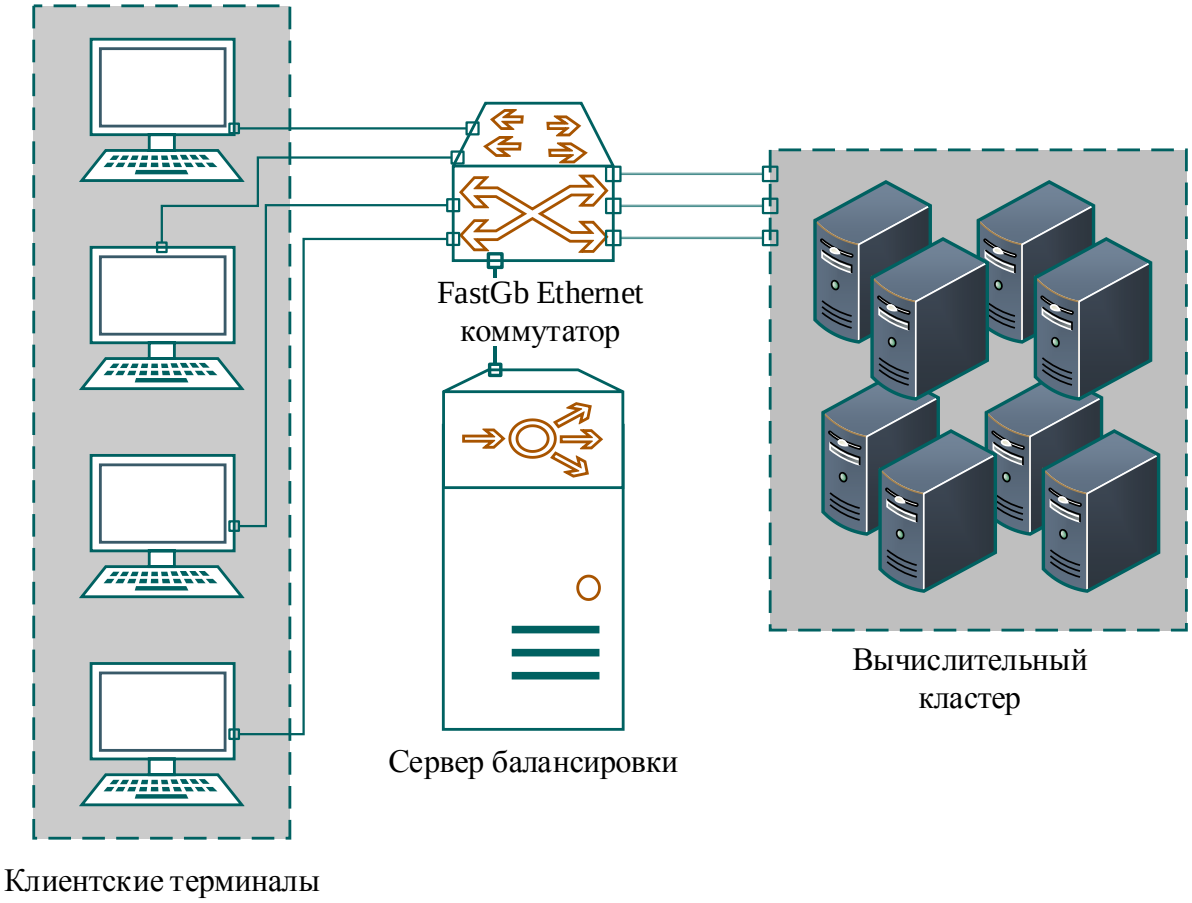


Рис. 1.10. Виртуализация представлений

При этом приложения будут работать независимо друг от друга, не внося никаких изменений в операционную систему, пользователь видит приложение и работает с ним на своём терминале, хотя на самом деле приложение выполняется на удалённом сервере, а пользователю передаётся лишь картинка удалённого приложения.

В результате активное развитие получают "облачные" вычисления, пользователи имеют доступ к собственным данным, но не управляют и не задумываются об инфраструктуре, операционной системе и собственно программном обеспечении, с которым они работают.

3. Методы организации защиты в облаке

Виртуализация предоставляет такие выгоды как энергоэффективность и производительность, и основными вопросами которые возникают при организации виртуализации являются угрозы безопасности. Организация безопасности на крупных предприятиях строится на основе стандартов, установленных разными нормативными требованиями (Рис.1.11).

Так как облачные технологии являются новыми технологиями, то проблема безопасности только обостряется, так как:

- существует достаточно большое количество разнообразных типов облаков как по уровням доступа так и по уровням предоставления услуг.
- не разработано достаточное количество стандартов для обеспечения защиты в облаке.
- провайдерам предоставляющим услуги облачных вычислений нужно будет выявлять и устранять новые бреши в системе

безопасности, характерные не только для реальных сред, но и для виртуальных сред.

Поэтому сегодня важным и актуальным является разработка новых методов и подходов в реализации облачных инфраструктур.

Рис. 1.11. Логическая схема организации информационной безопасности

Принципы организации безопасности

Надо сказать, что при организации безопасности виртуализации и облака мы имеем дело с данными, данные должны соответствовать трем каноническим правилам информационной безопасности:

- целостность
- доступность
- конфиденциальность

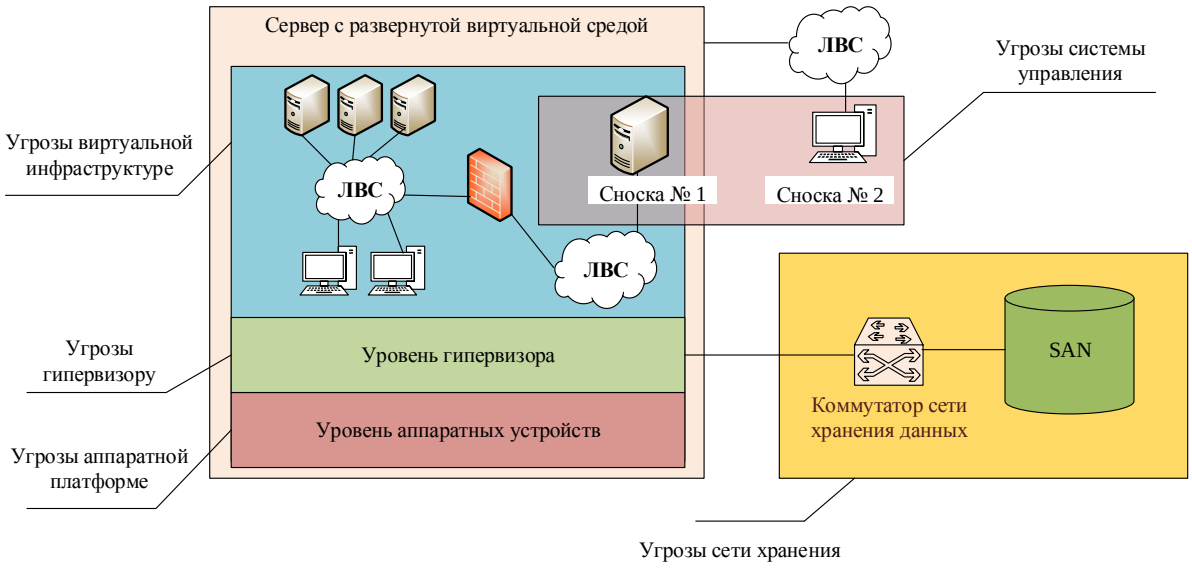
Поэтому при построении уровней безопасности целевой системы требуется создать:

- безопасность на уровне аппаратуры
- безопасность на программном уровне

Как уже было отмечено, построение уровней безопасности в организациях осуществляется в соответствии со стандартами и

нормативами. Существует определенный набор возможных типов угроз при организации виртуализации (Рис.1.12):

- угрозы виртуальной инфраструктуре
- угрозы гипервизору
- угрозы аппаратной платформе
- угрозы сети хранения
- угрозы системы управления



Сноска № 1 – Сервер управления виртуальной средой;

Сноска № 2 – автоматизированное рабочее место администратора;

Рис. 1.12. Возможные угрозы при организации виртуализации

Следует отметить, что при разработке методов и средств защиты чаще всего требуется использовать комплексный организационно-технический подход.

Виды угроз осуществляемые на аппаратном уровне

Рассмотрим основные угрозы имеющие место при организации аппаратной безопасности.

Нарушение в работе аппаратных компонентов серверного оборудования с установленными компонентами виртуальной среды. При таком виде угроз происходят следующие типы нарушений в безопасности:

- нарушается доступность к персональным данным пользователей;
- происходит остановка виртуальных машин, запущенных на аппаратной платформе;
- нарушение работы зависящих от их работоспособности смежных ИТ-систем;
- потеря данных пользователей, нарушение целостности, в процессе остановки работы виртуальной машины;
- уничтожение данных пользователей, размещенных на поврежденных носителях данных;

Рассмотрим обстоятельства при которых может происходить такой тип нарушений:

- отказы и сбои в работе смежных обеспечивающих систем как несанкционированные так и случайные (электропитание, кондиционирование и пр.);
- умышленное или случайное нанесение физических повреждений аппаратным компонентам виртуальной среды;
- несанкционированное извлечение, замена или кража компонентов оборудования;

Помимо аппаратных компонентов системы надо учесть и безопасность сетевого уровня в организации безопасности. При таком виде угроз происходят следующие типы нарушений в безопасности:

- неавторизированный пользователь получает возможность проведения сетевых атак из не доверенных сетевых сегментов на виртуальные машины, развернутые на аппаратной платформе;
- нарушение работы информационной системы персональных данных, развернутых в виртуальной среде;

Рассмотрим обстоятельства при которых может происходить такой тип нарушений:

- подключение сетевых интерфейсов аппаратной платформы к не доверенным сетевым сегментам;
- изменение сетевых настроек;

Рассмотрим также виды угроз для сети хранения данных. При таком виде угроз происходят следующие типы нарушений в безопасности:

- нарушается доступность к персональным данным пользователей;
- персональные данные пользователей могут быть искажены;
- злоумышленник может получить доступ к персональным данным пользователя;
- имея физический доступ к носителям информации находящимися в составе СХД осуществить кражу носителе(ей) информации для дальнейшей кражи личной информации пользователя;

Рассмотрим обстоятельства при которых может происходить такой тип нарушений:

- нарушитель получил физический доступ к дискам СХД что позволяет ему осуществить кражу или порчу оборудования;
- нарушитель получил доступ к автоматизированному рабочему месту администратора и смог подобрать пароль;
- нарушитель получил несанкционированный (удаленный) доступ к управляющим интерфейсам компонентов сети хранения данных;

Виды угроз осуществляемые на программном уровне

Рассмотрим основные угрозы имеющие место при организации уровня программной безопасности.

Нарушение состояния работы системного ПО виртуализации (гипервизора). При таком виде угроз происходят следующие типы нарушений в безопасности:

- нарушение работы информационной системы персональных данных, развернутых в виртуальной среде посредством воздействия на гипервизор;
- уничтожение/кража/искажение персональных данных, обрабатываемых в рамках виртуальной машины, к которой произошло несанкционированное подключение при условии неправильной настройки безопасности на гипервизоре;
- несанкционированный доступ к ресурсам виртуальных машин вследствие некорректных настроек гипервизора;
- подмена исполняемых модулей по гипервизора;
- вследствие сетевых атак типа «переполнение буфера» на открытые сетевые порты сервера с гипервизором в случае возникновения в его по уязвимостей, злоумышленник может получить несанкционированный удаленный доступ к ресурсам гипервизора;
- вследствие атак типа «отказ в обслуживании» в отношении виртуальных машин может произойти истощение вычислительных ресурсов сервера с гипервизором;

Рассмотрим обстоятельства при которых может происходить такой тип нарушений:

- сотрудники имеющие легитимные права доступа могут умышленно или случайно могут изменить настройки безопасности гипервизора и испортить данные пользователей ВМ;
- злоумышленник находясь в сегменте сети может получить доступ к данным пользователя через бреши в безопасности неправильно настроенного гипервизора;
- злоумышленник может осуществить DDOS атаку на гипервизор и тем самым вызвать истощение вычислительных ресурсов;

- злоумышленник получивший доступ к ВМ пользователя и сумевший обойти защиту гипервизора;

Следующим типом является угрозы для системы управления виртуальной средой. При таком виде угроз происходят следующие типы нарушений в безопасности:

- получение несанкционированного доступа к консоли управления виртуальной инфраструктурой (АРМ администратора виртуальной среды);
- получение несанкционированного доступа к настройкам виртуальных машин;
- получение несанкционированного удаленного доступа к интерфейсу системы управления;

Рассмотрим обстоятельства при которых может происходить такой тип нарушений:

- MITM (man in the middle) атаки, обход уровня безопасности, перехват сессии с подключением к сегменту управления;
- получение доступа к консоли управления и подбор пароля, возникает из-за неправильной настройки политики безопасности;
- получение доступа к смежному сегменту сети и осуществление выше описанных атак;

Рассмотрим также угрозы направленные на ИТ инфраструктуру, реализованную в рамках виртуальной среды. При таком виде угроз происходят следующие типы нарушений в безопасности:

- развертывание новых плохо защищенных виртуальных машин с возможностью компрометации их в дальнейшем;
- «смешение» информации различного уровня конфиденциальности в рамках единой аппаратной платформы;
- несанкционированное сетевое подключение к виртуальной машине;

- подмена и/или перехват данных и съём образа оперативной памяти виртуальных машин в процессе их миграции средствами виртуальной среды;
- проведение сетевых атак между виртуальными машинами;
- вирусное заражение виртуальных машин;

Рассмотрим обстоятельства при которых может происходить такой тип нарушений:

- использование зараженных сменных носителей, получение вирусов средствами электронной почты, через интернет;
- реализация атак типа «переполнение буфера», SQL-injection и пр. с использованием системных и прикладных уязвимостей;
- удаленное проникновение в сетевой сегмент с виртуальными машинами и реализация различного рода сетевых атак;
- плохо настроенная групповая политика безопасности между ВМ находящимися в сети;

Следует помнить, что при реализации сложной системы уровень её безопасности уменьшается именно поэтому очень хорошо нужно относится к вопросам по ее организации.

Потенциальные нарушители

После рассмотрений типов угроз нужно также и указать кто чаще всего попадает в списки нарушителей безопасности:

- сотрудник, не имеющий права доступа в помещения с размещенными аппаратными компонентами;
- посетители;
- администраторы различного уровня как с возможностью полного доступа так и с ограниченными правами на определенные типы изменений параметров системы;

- сотрудник, имеющий легитимный доступ для осуществления определенных действий, и сотрудник имеющий ограниченные права на изменение свойств системы;
- сотрудник, имеющий сетевой доступ к сетевому сегменту, к которому подключен сервер с установленным гипервизором;
- нарушитель, действующий в/из-за пределов сети организации, удаленно проникший в сетевой сегмент;
- пользователи системы.

Выводы по первой главе

Итогом по результатам исследования является:

1. формирование теоретических аспектов по методам виртуализации и видам организации современных облачных систем;
2. рассмотрение теоретического подхода в разработке организации методов защиты;
3. приведение примеров работы математического аппарата при организации виртуализации на программном уровне;

Проведенный анализ исследования позволят выбрать подходящий уровень виртуализации системы и наиболее лучше использовать аппаратные средства вычислительных кластеров системы для построения облачной среды. Также при разработке стратегии безопасности требуется проанализировать угрозы безопасности, влияющие на новую среду. В результате анализа были выявлены угрозы обработки персональных данных в специфике рассматриваемой работы.

Рассмотрены методики организации хранения и передачи данных в условиях виртуализации систем, было выявлено что:

1. Данные передаются между виртуальными машинами внутри виртуальной среды. Поскольку виртуальная машина – это файл,

хранящийся в хранилище данных, то передача персональных данных между виртуальными машинами предполагает, что данные выходят из одной области хранилища данных, проходят сетевые коммутаторы, попадают на сервер с развернутой виртуальной средой, а затем в обратном порядке возвращается в хранилище, но уже для другой виртуальной машины.

2. Данные передаются между виртуальной средой и внешними средами (как реальной, так и виртуальной). Особенность в том, что из сетевого интерфейса одного физического сервера могут «выходить» данные, относящиеся к различным виртуальным серверам или информационным системам и, соответственно, возникает вопрос интеграции механизмов защиты виртуальной среды и внешних физических компонентов.

Выполнив анализы развития современных облачных систем и возможных угроз персональным данным пользователей, а также определив источники возникновения этих угроз определены требования по организации облачной среды:

1. провести анализ потребностей в облачных технологиях для образовательного учреждения,
2. выработать алгоритм реализации как облачного сервиса так и системы безопасности облачной инфраструктуры.

ГЛАВА II. РАЗРАБОТКА АЛГОРИТМА ОРГАНИЗАЦИИ ЧАСТНОГО ОБЛАКА ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ

1. Изучение потребностей образовательного учреждения и составление параметров требований к системе

Образовательное учреждение представляет собой не только административное здание и входящий в него обслуживающий персонал но также и набор состоящий из парка вычислительной техники, оргтехники, средств радиопередачи и т.п.

Облачные решения призваны не только облегчить труд администраторов но и решить ряд кардинально важных вопросов в организации работы образовательного учреждения, например:

- обеспечения контроля версионности программного обеспечения;
- своевременное обновление модулей операционной системы и антивирусных баз;
- обеспечение контроля доступа к данным пользователей;
- обеспечение защиты работающих в системе образовательного учреждения компьютеров пользователей;

Используя методы виртуализации и предоставив удобный пользовательский интерфейс можно наиболее лучше задействовать аппаратные ресурсы организации уменьшив время их простоя.

При Ташкентском Университете Информационных Технологий (ТУИТ) читается широкий круг по различным направлениям технических дисциплин. За основу был выбран предметный комплекс специальных предметов преподаваемых на кафедре ПОИТ программное обеспечение информационных технологий факультета ПИ при ТУИТ (таблица 2.1). Целью является выявление требований и задач предъявляемых к облачным технологиям в среде образовательного учреждения. Отмечу что

разрабатываемые методики построения облачным технологиям в среде образовательного учреждения образовательного учреждения могут быть применены и к другим кафедрам при ТУИТ с введением поправок по используемому программному обеспечению.

Таблица № 2.1

Предметный комплекс специальных предметов кафедры ПОИТ

Дисциплина	Используемое программное обеспечение
Бакалавриатура	
Технология программирования	Пакет офисных программ, набор сред и средств исполнения, Microsoft Visual Studio, Borland C++; Microsoft SQL Server, NetBeans, IntellyIDEA, набор специальных программ для организации проведения урока (по просьбе преподавателя).
Операционные системы	Пакет офисных программ, набор сред и средств исполнения, набор различных вариантов ВМ с различными операционными системами, набор специальных программ для организации проведения урока (по просьбе преподавателя).
Объектно-ориентированные языки программирования	Набор сред и средств исполнения, Microsoft Visual Studio, Borland C++, NetBeans, IntellyIDEA, набор специальных программ для организации проведения урока (по просьбе преподавателя).
Основы системного проектирования и моделирования	Пакет офисных программ, набор сред и средств исполнения, Matlab (и совокупные с ним мат. пакеты), средства проектирования, набор специальных программ для организации проведения урока (по просьбе преподавателя).
Прикладные математические	Пакет офисных программ, набор сред и средств исполнения, Matlab (и совокупные с ним мат.

программные пакеты	пакеты), средства проектирования, набор специальных программ для организации проведения урока (по просьбе преподавателя).
Магистратура	
Язык программирования Java	Пакет офисных программ, набор сред и средств исполнения, NetBeans, IntellyIDEA, компоненты языка Java, Matlab, набор специальных программ для организации проведения урока (по просьбе преподавателя).
SQL-технологии	Пакет офисных программ, набор сред и средств исполнения, Microsoft Visual Studio, Borland C++; Microsoft SQL Server, NetBeans, IntellyIDEA, Matlab, набор специальных программ для организации проведения урока (по просьбе преподавателя).
Интеллектуальный анализ данных	Пакет офисных программ, набор сред и средств исполнения, Microsoft Visual Studio, Borland C++; Microsoft SQL Server, NetBeans, IntellyIDEA, Matlab, набор специальных программ для организации проведения урока (по просьбе преподавателя).
Объектно-ориентированный анализ и проектирование	Пакет офисных программ, набор сред и средств исполнения, средства для проведения анализа, Microsoft Visual Studio, Borland C++; Microsoft SQL Server, NetBeans, IntellyIDEA, Matlab, набор специальных программ для организации проведения урока (по просьбе преподавателя).
Математические системы и их	Пакет офисных программ, набор сред и средств исполнения, Matlab (и совокупные с ним мат.

программирование	пакеты), средства проектирования, набор специальных программ для организации проведения урока (по просьбе преподавателя).
Web-технологии	Пакет офисных программ, программы для написания web-проектов (HTML/CSS, JS, PHP редакторы), набор фреймворков и CMS, набор специальных программ для организации проведения урока (по просьбе преподавателя).

2. Анализ выбора основы построения облака образовательного учреждения

Для проведения анализа по выбору типа облака рассмотрим эталонную архитектуру облачных вычислений NIST (рис. 2.1). Она является, по сути, обобщённой концептуальной моделью высокого уровня и служит эффективным инструментом при формировании требований, структур и действий облачных вычислений. Архитектура определяет состав участников, деятельность и функции, которые могут быть реализованы в процессе разработки архитектур облачных вычислений и устанавливает взаимоотношения между участниками облачных вычислений. Она содержит некоторые примеры и описания, являющиеся основой при обсуждении требований и стандартов для облачных вычислений. На рисунке 2.1 показаны сервисы, с точки зрения задачи данной работы представляет интерес элемент «переносимость/ интероперабельность».

Однако, вычисления являются только одной из областей применения «облаков». Второй областью является хранение информации. И, наконец, третьей областью является гибридное использование облаков: для сбора, хранения данных и в качестве средства обработки данных. Остановимся на

проблеме интероперабельности облаков с целью раскрыть содержание (рис.2.1).

Рис. 2.1 Эталонная архитектура облачных вычислений NIST.

Как видно, архитектура облачных вычислений NIST содержит пять главных действующих субъектов: облачный потребитель, облачный провайдер, облачный аудитор, облачный брокер, облачный оператор связи.

Рассмотрим архитектуру вычислительной среды в аудитории кафедры ПОИТ проекта TEMPUS. Как видно из рисунка (Рис. 2.2) это набор рабочих станций и сервера. Следует сказать, что существующие виды облаков любого типа предполагают использование мощных систем хранения данных и мощного вычислительного оборудования.

В главе 1 были представлены три вида облаков, это:

- частные
- публичные
- гибридные

Так как пользователями облака, на начальном этапе, будут преподаватели и студенты только ташкентского филиала ТУИТ, то было предпринято решение о выборе модели частного облака образовательного учреждения. Данный подход позволит использовать безопасную и отказоустойчивую среду, произвести необходимое количество экспериментов на ограниченном круге пользователей в рамках университета для дальнейшего развития этого направления и его внедрения в производство.

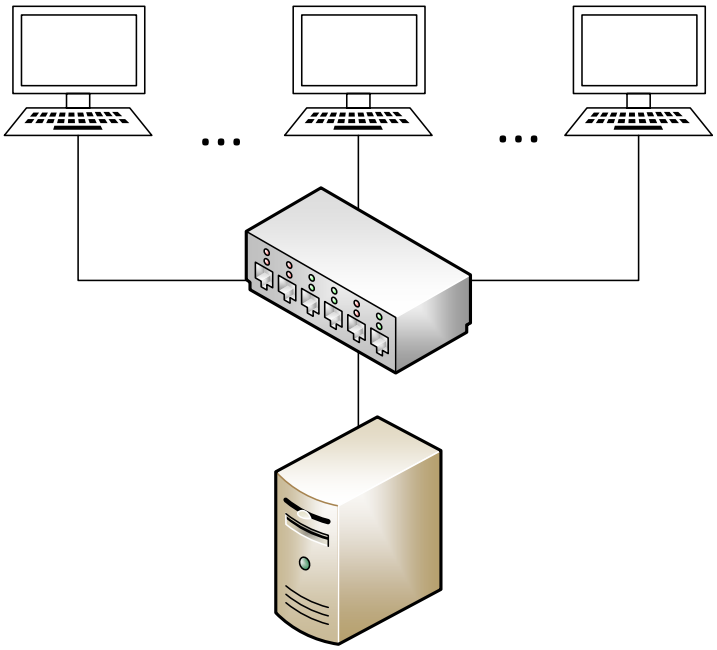


Рис.2.2. Архитектура компьютерной системы в аудитории проекта TEMPUS при кафедре ПОИТ.

Использовалась в первоначальном варианте работа аппаратной части вычислительной среды в аудитории кафедры по проекту TEMPUS. Выделим набор основных характеристик:

Пользовательские компьютеры в количестве 15 (пятнадцати) штук:

- процессор: *Intel core i3* с частотой *2.3 ГГц*.

- оперативная память объемом 2 Гб.
- жесткий диск объемом: 500 Гб.
- видеокарта: интегрированная.

Сервер:

- процессор: *Intel Celeron* с частотой *1.7 ГГц.*
- оперативная память объемом 2 Гб.
- жесткий диск объемом: 250 Гб.

Сетевые устройства:

- сетевые устройства представлены свичом первого уровня с пропускной способностью 100 Mb/s.
- сетевые кабеля связи.

Как видно мощности компьютера подходят для организации учебного процесса на местах. Задачей в данной работе являлось:

- создание частного облака образовательного учреждения;
- разработка решений для обеспечения безопасности в частном облаке образовательного учреждения;

С целью рационального распределения ресурсов сначала требуется создать кластер из существующих компьютеров и произвести его настройку (рис 2.3).

Перечислим требования предъявляемые к облаку:

- в облаке должны находится образы виртуальных операционных систем
- в облаке должен находится набор всех программ используемых на кафедре для обучения.

Выше описанные требования подходят под создание облака типа SaaS – данная модель, предоставляет возможность конечному потребителю использовать прикладное программное обеспечение провайдера, работающего в облачной среде и предоставляющий доступ с различных клиентских устройств интерфейса программы. Управление и контроль

инфраструктуры облака осуществляется провайдером предоставляющим, в том числе сети, сервера, операционные системы, системы хранения, и даже индивидуальные возможности приложения.

Следовательно методом предварительного анализ был выбран подход в реализации частного облака образовательного учреждения типа SaaS.

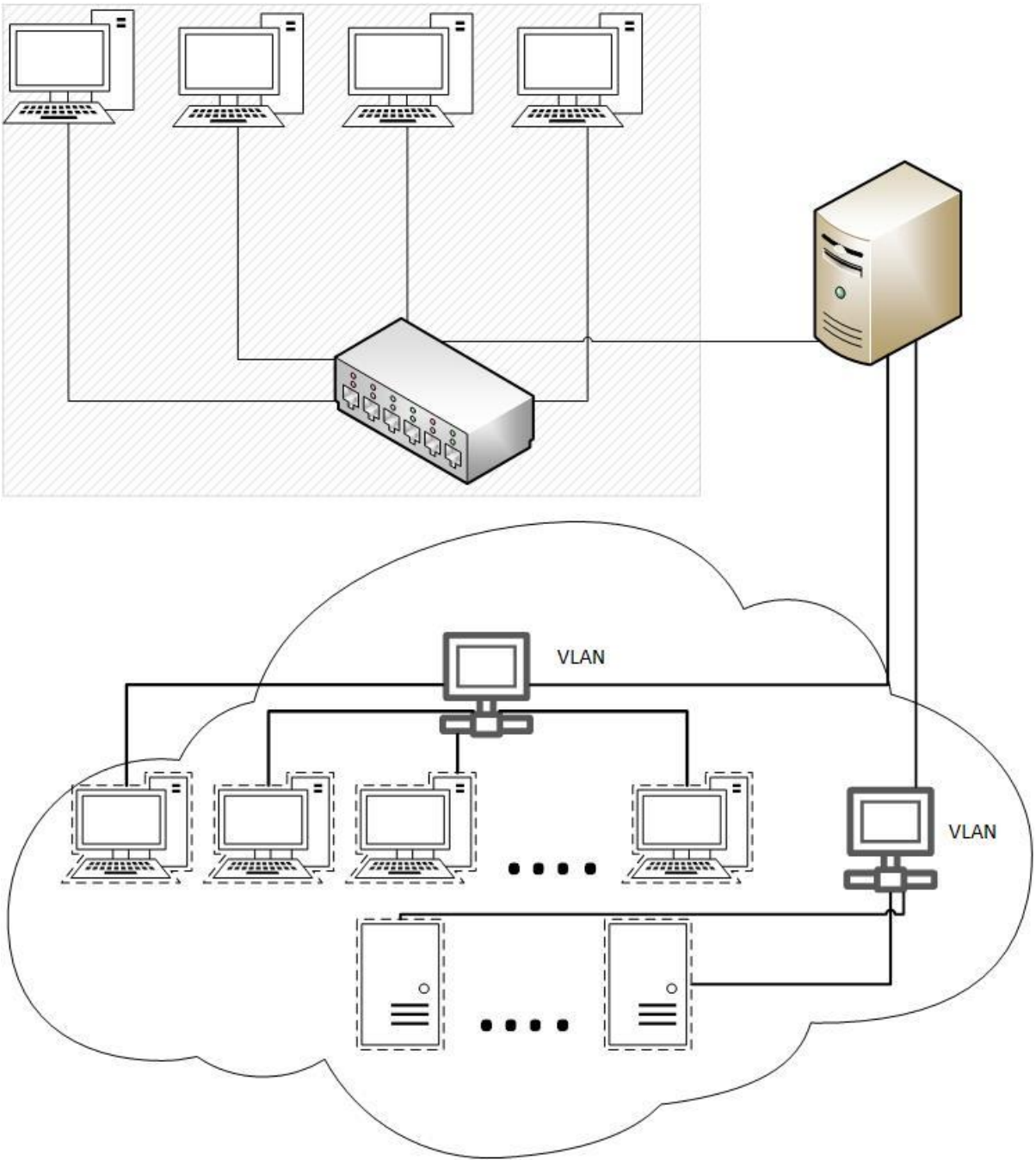


Рис.2.3. Архитектура разрабатываемой системы.

3. Сравнение систем виртуализации и выработка алгоритма построения частного облака образовательного учреждения

Для решения поставленной задачи будем рассматривать технологию паравиртуализации, т.к. она позволяет наиболее лучше использовать аппаратные ресурсы машины и предоставляет механизмы для организации базовой защиты.

Паравиртуализацию рассмотрим на основе трех продуктов:

1. Hyper-V – разработка компании Microsoft

Hyper-V поддерживает разграничение согласно понятию раздел. Раздел — логическая единица разграничения, поддерживаемая гипервизором, в котором работают операционные системы. Каждый экземпляр гипервизора должен иметь один родительский раздел. Стек виртуализации запускается на родительском разделе и обладает прямым доступом к аппаратным устройствам. Затем родительский раздел порождает дочерние разделы, на которых и располагаются гостевые ОС. Родительский раздел создает дочерние при помощи API гипервизора. Дочерний раздел также может породить собственные дочерние разделы.

Виртуализованные разделы не имеют ни доступа к физическому процессору, ни возможности управлять его реальными прерываниями. Вместо этого, у них есть виртуальное представление процессора и гостевой виртуальный адрес, зависящий от конфигурации гипервизора, необязательно занимающий все виртуальное адресное пространство. Hyper-V может определять подмножество процессоров для каждого раздела. Hyper-V управляет прерываниями процессора и перенаправляет их в соответствующий раздел, используя логический контроллер искусственных прерываний (Synthetic Interrupt Controller или сокр. SynIC).

Hyper-V может аппаратно ускорять трансляцию адресов между различными гостевыми виртуальными адресными пространствами при помощи IOMMU, которое работает независимо от аппаратного управления памятью, используемого процессором.

2. VMware ESX Server – разработка компании VMware Inc.

Как и Hyper-V, VMware ESX Server поддерживает разграничение согласно понятию раздел. Также и схож процесс работы с Hyper-V. Выделим ключевые компоненты архитектуры ESX Server:

- уровень виртуализации ESX server: отделяет основные физические ресурсы от виртуальных машин;
- менеджер ресурсов: создает виртуальные машины и обеспечивает их ресурсами процессора, памяти, сети и дисковой подсистемы. эффективно отображает физические ресурсы на виртуальные;
- служебная консоль: управляет установкой, настройкой, администрированием, устранением неисправностей и техническим обслуживанием ESX server. служебная консоль работает в своей собственной виртуальной машине.
- компоненты аппаратного интерфейса, в том числе драйверы устройств: обеспечивают зависящие от аппаратуры службы, скрывая аппаратные различия между разными частями системы.

3. Xen – разработка компании Citrix.

Xen добавляет уровень виртуализации между аппаратной частью системы и виртуальными машинами, превращая оборудование системы в пул логических вычислительных ресурсов, которые Xen может динамически выделять любой гостевой операционной системе. Операционные системы, работающие в виртуальных машинах, взаимодействуют с виртуальными ресурсами, как если бы это были физические ресурсы. Выделим ключевые компоненты архитектуры Xen.

- возможность полной виртуализации.

- Xen может исполнять несколько гостевых ос, каждую на своей виртуальной машине.
- вместо драйвера, исполнение происходит в Xend, демоне Xen.

Таблица 2.2.

Сравнение возможностей систем виртуализации

Название	Платный	Имеет промышленный стандарт	Количество поддерживаемых гостевых систем	Простота в использовании	Триальный период
Xen	-	-	8	+	Неогр.
Hyper-V	-	+	15	+	Неогр.
VMware ESX Server	+	+	12	+/-	60 дней

По выше отмеченным фактам можно составить таблицу (таблица 2.2), в качестве сравнения выделим ключевые факторы:

- наличие промышленного стандарта
- является ли продукт проприетарным, если да то какое количество составляет триальный период использования
- простота в использовании
- количество поддерживаемых семейств ОС ВМ

Из таблицы можно сделать вывод что наиболее вероятным является выбор гипервизора Hyper-V. Следовательно в работе будет использована программная экосистема от компании Microsoft и ее решения в области построения облачных решений.

4. Алгоритм реализации облака для образовательного учреждения

Для выработки алгоритма создания частного облака выделим основные задачи при его организации:

- достижение выгодности за счет контролируемого постоянного улучшения функционала
- создание ощущение непрерывной доступности
- подход с точки зрения поставщика услуг
- оптимальное использование ресурсов
- целостный подход при проектировании доступности
- сведение к минимуму вмешательства человека
- повышение прогнозируемости
- создание безупречных условий работы для пользователей
- регулирование параметров желаемого поведения

Рассмотрим перечисленные пункты более подробно.

Достижение выгодности за счет контролируемого постоянного улучшения функционала

Все вложения в ИТ-услуги должны быть четко связаны с получением прибыли. Часто возникают ситуации, когда прибыль от крупных вложений в стратегические инициативы контролируется лишь на начальном этапе, а затем контроль ослабевает, из-за чего прибыль снижается. Постоянное измерение выгодности, которую обеспечивает какая-либо служба, дает возможность применять улучшения, обеспечивающие наивысшую выгодность. Таким образом, постоянно совершенствующиеся технологии работают на благо и потребителей, и поставщиков услуг. При правильном соблюдении этот принцип обеспечит постоянное развитие ИТ-служб, предоставляющих гибкие возможности, которые требуются при организации облачных вычислений.

Создание ощущение непрерывной доступности

Как правило, удовлетворение требований пользователей в отношении доступности было для ИТ-отделов непростой задачей. Технологические ограничения, архитектурные решения, недостаточная продуманность процессов — все это способствовало перебоям в работе ИТ-среды. Добиться высокой доступности можно, но лишь при весьма значительных затратах на построение надежной дублированной инфраструктуры. По соображениям безопасности доступ к большинству служб возможен лишь в пределах офисов компаний. Облачные службы должны предоставлять возможность обеспечения высокой доступности при невысоких затратах, а также решать проблемы безопасности, чтобы можно было предоставлять доступ к службам через Интернет.

Подход с точки зрения поставщика услуг

ИТ-отделы компаний зачастую используют изолированный подход, который неизбежно приводит к неэффективности. Архитекторы решений могут считать, что предоставление инфраструктуры для совместного использования несколькими решениями слишком рискованно. Невозможно устранить влияние одного решения на другое, поэтому каждое решение использует собственную инфраструктуру, а общий доступ к ресурсам осуществляется лишь при наличии гарантий безопасности. В результате создаются проекты, из-за которых неэффективность только увеличивается (виртуализация, консолидация серверов в центрах обработки данных).

Облачная служба всегда совместно используется несколькими потребителями. Ее необходимо создать таким образом, чтобы потребитель с уверенностью мог ею пользоваться; характеристики мощности, производительности и доступности должны быть четко определены. В то же время облако должно приносить организации выгоду. Поскольку поставщики услуг продают свои услуги потребителям, существует четкое разделение между поставщиком и его клиентами (потребителями). Такая

модель отношений вынуждает поставщика четко определять свои услуги с точки зрения их возможностей, ресурсов, производительности, доступности и стоимости. ИТ-отделы компаний должны применять именно такой подход, предлагая свои услуги компаниям.

Оптимальное использование ресурсов

Оптимизация ресурсов способствует повышению эффективности и сокращению затрат; оптимизация достигается главным образом за счет совместного использования ресурсов. Отделение платформы от физической инфраструктуры позволяет реализовать этот принцип за счет совместного использования ресурсов, объединенных в пул. Если разрешить нескольким потребителям совместно использовать ресурсы, это приведет к более полному использованию ресурсов и, следовательно, к более производительному и эффективному использованию инфраструктуры. Оптимизация путем разделения поддерживает многие другие принципы и в конечном итоге помогает сокращать расходы и повышать гибкость.

Целостный подход при проектировании доступности

При использовании традиционной модели ИТ-отделы предоставляли высоко доступные службы, используя стратегию избыточности, или дублирования. В случае отказа одного компонента нагрузка мгновенно передавалась на дублирующий компонент, который до этого находился в состоянии ожидания. Избыточность часто применялась на нескольких уровнях технологий, поскольку каждый уровень не может исходить из того, что нижележащий уровень будет высоко доступным. Такая избыточность, особенно на уровне инфраструктуры, приводит к существенному увеличению как капитальных, так и эксплуатационных расходов.

Основной принцип облачной среды заключается в предоставлении высоко доступных служб за счет устойчивости. Вместо того чтобы

пытаться предотвращать сбои, при проектировании облака принимается и ожидается, что компоненты облака могут выходить из строя. Усилия направлены на то, чтобы сократить ущерб от сбоев и быстро восстановить работоспособность после них. Виртуализация, обнаружение в реальном времени и автоматическое реагирование на нарушения работоспособности позволяют быстро переносить нагрузки с отказывающих компонентов инфраструктуры; зачастую при этом даже не возникает заметных перебоев в работе службы.

Сведение к минимуму вмешательства человека

Высокий уровень устойчивости, необходимый для работы облачной среды, невозможен без высокой степени автоматизации. Если обнаружение условий сбоев и реагирование на них осуществляются с участием человека, то непрерывная доступность обслуживания невозможна без полностью избыточной инфраструктуры (т.е. инфраструктуры, все компоненты которой дублированы). Следовательно, нужна полностью автоматическая система управления структурой, которая будет динамически выполнять оперативные задачи, автоматически обнаруживать условия сбоев и реагировать на них, а также сможет поддерживать гибкое добавление или удаление ресурсов в соответствии с требуемой нагрузкой. Важно понимать различия между ручными, частично автоматизированными или полностью автоматическими действиями.

Ручной процесс — это процесс, все этапы которого требуют участия человека. При частичной автоматизации некоторые этапы автоматизированы, но определенные действия человека все еще нужны (например, человек должен обнаруживать необходимость запуска каких-либо процессов, запускать сценарии). При полной автоматизации ни один этап процесса, от обнаружения до реагирования, не должен требовать участия человека.

Повышение прогнозируемости

Как правило, в обычной ИТ-среде уровень качества обслуживания было невозможно предсказать. Недостаточная прогнозируемость мешает компаниям в полной мере воспользоваться стратегическими преимуществами ИТ-среды. По мере развития общедоступных облачных ИТ-решений компании могут выбирать общедоступные предложения (вместо внутренней ИТ-среды) для достижения более высокой прогнозируемости. Для того чтобы внутренние ИТ-отделы сохранили свое место в структуре компаний, они должны предоставлять услуги с прогнозируемым уровнем качества, как у общедоступных предложений.

Создание безупречных условий работы для пользователей

ИТ-стратегии все чаще сочетают услуги различных поставщиков при разработке наиболее выгодного решения для организации вычислительного процесса. По мере увеличения количества услуг, предоставляемых потребителям разными поставщиками, повышается вероятность сбоев, поскольку транзакции пересекают границы между поставщиками. Тот факт, что услуги, предоставляемые потребителю, осуществляются несколькими поставщиками, не должен иметь никаких видимых отрицательных последствий: пользователь не должен сталкиваться с перебоями и нарушениями в работе.

В качестве примера можно привести пользователя, использующего портал для доступа к информации в организации (например, для проверки состояния заказа на закупку). Пользователь может просмотреть заказ, используя систему управления заказами компании; затем пользователь может щелкнуть ссылку, ведущую к более подробным сведениям о покупателе, которые, в свою очередь, хранятся в CRM-системе в общедоступном облаке. Таким образом, пользователь пересекает границу между внутренней системой компании и системой во внешнем общедоступном облаке. При этом пользователь не должен столкнуться ни с какими препятствиями и помехами в работе. Не должно быть никаких

дополнительных запросов проверки подлинности, интерфейс должен выглядеть и работать одинаково, и уровень производительности в обоих случаях также не должен различаться.

Регулирование параметров желаемого поведения

Если задать пользователям вопрос, какой уровень доступности того или иного приложения им требуется, в большинстве случаев они назовут такие показатели, как 99,999 % или даже 100 % времени работы без сбоев при обращении к ИТ-отделу для предоставления услуги. Такие запросы обычно обусловлены недостаточным пониманием фактических расходов на предоставление услуг и со стороны потребителя, и со стороны ИТ-поставщика. Если, например, ИТ-поставщик предложит своего рода каталог услуг, в котором будут четко и явно указаны затраты на удовлетворение требований к доступности в 99,999 %, то потребности (а значит, и ожидания от работы ИТ-отдела) будут мгновенно скорректированы.

Можно привести другой, более технический пример. Во многих организациях, применивших виртуализацию, возникло новое явление – разрастание количества виртуальных серверов, поскольку виртуальные машины создавались по требованию, но не поощрялась остановка виртуальных машин и их удаление, когда они были уже не нужны. Ощущение бесконечности ресурсов может привести к тому, что потребители будут просто использовать больше ресурсов, не стремясь рационально управлять нагрузкой. Ощущение бесконечности вычислительных ресурсов можно считать благом с точки зрения качества и гибкости услуги, но при безответственном использовании возникает отрицательное влияние на вычислительную мощность облака.

В приведенном выше случае поставщику нужно побуждать потребителей использовать только действительно необходимые им

ресурсы. Такого результата можно добиться посредством аудита об использовании ресурсов.

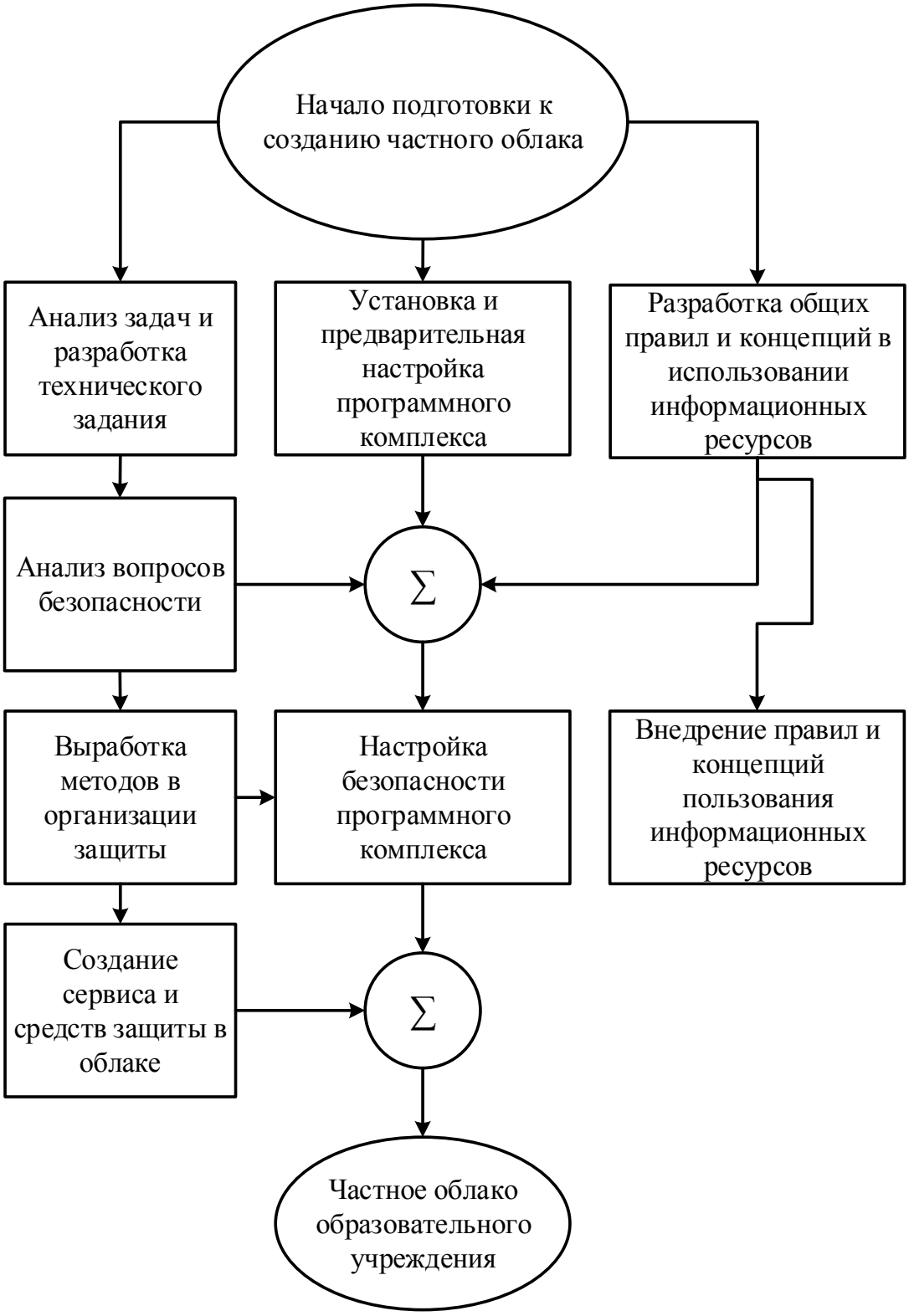


Рис.2.4. Алгоритм реализации облака образовательного учреждения.

Поощрение желаемого поведения потребителей является важнейшим принципом и связано также с работой по модели поставщика услуг.

На основании описанных выше пунктов главы и основных принципов опишем и построим алгоритм реализации облака образовательного учреждения (Рис.2.4).

Во первых стоит разработать техническое задание и выделить ряд лиц ответственных за создание и поддержку работоспособности в организации частного облака образовательного учреждения.

Далее следует произвести установку и предварительную настройку программного комплекса на предоставленную университетом аппаратуру.

После этого следует организовать дальнейшую настройку уровней безопасности и произвести разделение пользователей по ролям. С выделением для каждой роли определенных правил безопасности в системе.

Не стоит забывать и о том что, образовательное учреждение это место в котором из года в год происходит смена обучающихся студентов и каждый из них имеет своеобразный набор личностных и поведенческих качеств. Именно поэтому также важна и выработка общих правил и концепций в использовании информационных ресурсов ВУЗа как таковых.

Выводы по второй главе

В данной главе в результате проведенного исследования были выделены:

1. основные требования, предъявляемые к частному облаку образовательного учреждения;
2. выделены задачи, которые должна решать вычислительная среда облака;

3. разработан алгоритм организации облака образовательного учреждения.

При проведении анализа построения облачной среды необходимо понимать, насколько хорошо продумана архитектура системы и привносит ли она выгоду в целевую систему, а если нет, то следует определить, какой элемент стратегии требуется пересмотреть.

В результате были определены следующие требования к организации облачной инфраструктуры:

1. для предоставления вычислительных мощностей по запросу придется использовать высокоразвитую стратегию управления вычислительными ресурсами.
2. в качестве единиц построения масштабируемых систем следует использовать заранее заданные единицы сетей, хранилищ и вычислительных систем.
3. требуется тщательно запланировать время, необходимое для приобретения и развертывания каждой такой единицы. Средства управления должны быть запрограммированы так, чтобы учитывать единицы масштабирования, время подготовки и развертывания систем, текущие и прошлые тенденции использования вычислительных мощностей (поскольку может потребоваться развертывание дополнительных единиц).
4. требуется организовать диалог с потребителем, чтобы оценить новые и изменяющиеся инициативы, которые могут повлиять на сложившиеся тенденции использования вычислительных мощностей.

ГЛАВА III. РЕАЛИЗАЦИЯ ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ЧАСТНОГО ОБЛАКА ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ПО УЛУЧШЕНИЮ БЕЗОПАСНОСТИ

На основании проведенных анализов описанных во второй главе и на основании результатов первой главе были разработаны основные концепции безопасности. Но в ходе практической реализации были выявлены следующие недостатки:

- при запуске виртуальных машин происходит стартовая нагрузка на облачную инфраструктуру, что может привести к падению системы как таковой.
- при включении виртуальной ос не сразу запускаются групповые политики безопасности, что может привести к исполнению вредоносного кода со съемных носителей
- пользователь виртуальных платформ может произвести попытку взлома и тем самым нарушить работу системы
- коммунальность данных или их потеря. вопросы CIA. подделка данных. уничтожение данных.
- возможности неавторизированного доступа.

Именно поэтому и стало необходимым решать проблемы безопасности имеющие частный характер в облаке образовательного учреждения.

1. Формирование требований по безопасности частного облака

Концепция требований по безопасности частного облака образовательного учреждения получила общее описание при рассмотрении главы 1. Отмечу что все требования должны быть задокументированы и на основании их и политик безопасности должен быть разработан концепт

правил по использованию облачных вычислений при ТУИТ. Сформируем требования безопасности:

- Организация отказоустойчивости в работе как с персональными данными пользователей так и с виртуальными машинами.
- Организация уровней безопасности и включение ролей безопасности моментально в процессе работы
- Организация доступности частного облака образовательного учреждения
- Организация и соблюдение разработанных правил поведения при работе с вычислительными ресурсами облака а также его программной составляющей.

2. Разработка мер облачной безопасности

При разработке мер облачной безопасности требуется организовать аппаратную и программную безопасность. В выработке необходимых подходов должны участвовать специалисты, а в основу их выводов должны ложится разработанные стандарты как регионального так и глобального характера.

На сегодняшний день важным элементом использования облачных технологий является отказоустойчивость, обычно отказоустойчивость организуется простым дублированием. Существует ряд международных стандартов, по облачным технологиям, поэтому «легкость» использования и многозначность термина «облака» будут постепенно уходить. К ожидаемым стандартам в этой сфере относятся: ISO/IEC CD 17788 «Информационные технологии – Распределенные прикладные платформы и сервисы – Облачные вычисления – Общие положения и словарь»; ISO/IEC WD 17789 «Информационные технологии – Облачные

вычисления – Эталонная архитектура» (Information Technology – Cloud Computing – Reference Architecture).

Объективно, в случае использования частного облака (как для целей IaaS, так и для SaaS) общий уровень безопасности повышается, поскольку используются единые надежные механизмы доступа ко всем сервисам вместо разрозненных решений на уровне каждого отдельного приложения. Уже появились международные стандарты (ISO/IEC WD TS 27017 – Руководство по мерам информационной безопасности для использования сервисами облачных вычислений, ISO/IEC WD 27018 – Свод практики по мерам защиты персональных данных при оказании публичных облачных услуг).

Требования предъявляемые к аппаратной части

На основании выше описанных стандартов рассмотрим требования предъявляемые к аппаратной части.

Для аппаратной среды важным является уровень надежности самой системы в целом. И при создании центров обработки данных выделяют два основных документа, которые чаще всего упоминаются при обсуждении стандартов центров обработки данных: это стандарт TIA 942 и классификация по уровням от Uptime Institute.

Рассмотрим классификацию по уровням:

Tier I — без резервирования. Доступность 99.671%.

Tier II — резервирование критических узлов. Доступность 99.741%.

Tier III — резервирование критических узлов, путей получения электроэнергии и трасс доставки холодоносителя. При этом есть возможность вывода любого узла из эксплуатации для его обслуживания с сохранением полной функциональности объекта в целом. Доступность 99.982%

Tier IV — это самый отказоустойчивый уровень, где допускается одна авария (а не плановый вывод узла из эксплуатации) в один момент

времени. Как пример аварии – критичная человеческая ошибка. По сути — это два Tier-вторых, которые построены в одном здании вокруг серверных стоек. Доступность 99.995%, что обеспечивает отказоустойчивость всего 26 минут в год.

На практике де-факто удалось реализовать вышеописанный стандарт только по TIER II (TIA 942).

Требования этого уровня:

- Наличие сетевой инфраструктуры;
- Наличие дублирования критических узлов системы (домен контроллеров, серверов балансировки);
- Наличие системы бесперебойного питания;

Следует сказать, что облако в нашем случае формировалось на основе распределенной системы обработки данных с центрами на узлах балансировщиков и единым центром управления в качестве которого выступал сервер системного центра.

Требования предъявляемые к используемому программному обеспечению

На основании выше описанных стандартов рассмотрим требования предъявляемые к программной части.

Требования этого уровня:

- отказоустойчивость при возникновении ошибок;
- отказоустойчивость при попытках взлома;
- блокирование попыток взлома;
- предоставление удобного интерфейса для использования ресурсами облака;
- использование программы должно быть простым и понятным пользователю

Из выводов по анализу проблемной области был выбран гипервизор от Microsoft, именно поэтому было решено использовать и остальные продукты серверного семейства этого производителя.

В работе был использован следующий программный комплекс:

- Windows Server 2012 R2 (триальная версия) – основная серверная ОС.
- Windows Hyper-V 2012 R2 (триальная версия) – гипервизор.
- Microsoft SQL Server 2012 (триальная версия) – СУБД.
- внутреннее ПО серверной ОС (службы и роли).
- программное обеспечение собственной разработки.

3. Разработка программного приложения для организации безопасности частного облака образовательного учреждения

Разработка программного приложения разделена на три этапа:

1) разработка скриптов по настройке базовых служб и ролей серверов системы. проектирование и реализация базы данных для приложения;
2) разработка веб-службы для предоставления унифицированной объектной модели.
3) разработка сайта и приложения для организации предоставления услуг в частном облаке образовательного учреждения.

Рассмотрим каждый пункт в отдельности.

Настройка базовых служб и ролей серверов системы. Проектирование и реализация базы данных для приложения.

В работе при реализации проблемной области использовался язык PowerShell v 3.0. PowerShell – это расширяемое средство позволяющая производить автоматизацию при работе с системами Windows.

PowerShell был выбран по ряду своих качеств, например:

- является интегрированным и построенным на архитектуре .Net;
- предоставляет удобство в работе с COM, WMI, ADSI;
- позволяет создавать единое окружение для осуществления администрирования;
- предоставляет механизм встраивания, благодаря чему компоненты PowerShell могут быть встроены в другие программы.

Как пример рассмотрим один из таких скриптов. Данный скрипт позволяет настроить сервер как домен контроллер службы Active Directory.

```
# Сценарий Windows PowerShell для развертывания AD DS
Import-Module ADDSDeployment
Install-ADDSForest `
-CreateDnsDelegation:$false `
-DatabasePath "C:\Windows\NTDS" `
-DomainMode "Win2012R2" `
-DomainName "tuit.cloud" `
-DomainNetbiosName "TUIT" `
-ForestMode "Win2012R2" `
    -InstallDns:$true `
    -LogPath "C:\Windows\NTDS" `
    -NoRebootOnCompletion:$false `
    -SysvolPath "C:\Windows\SYSVOL" `
    -Force:$true
```

Пример 3.1. Скрипт PowerShell для установки службы Active Directory

Стоит отметить что при установке службы Active Directory автоматически создается и база данных доменного каталога. Но при разработке была использована собственная база данных, это было сделано для организации уровня абстракции данных в целях повышения уровня безопасности системы. В качестве основной среды разработки базы данных был выбран продукт компании Microsoft, СУБД SQL Server 2012.

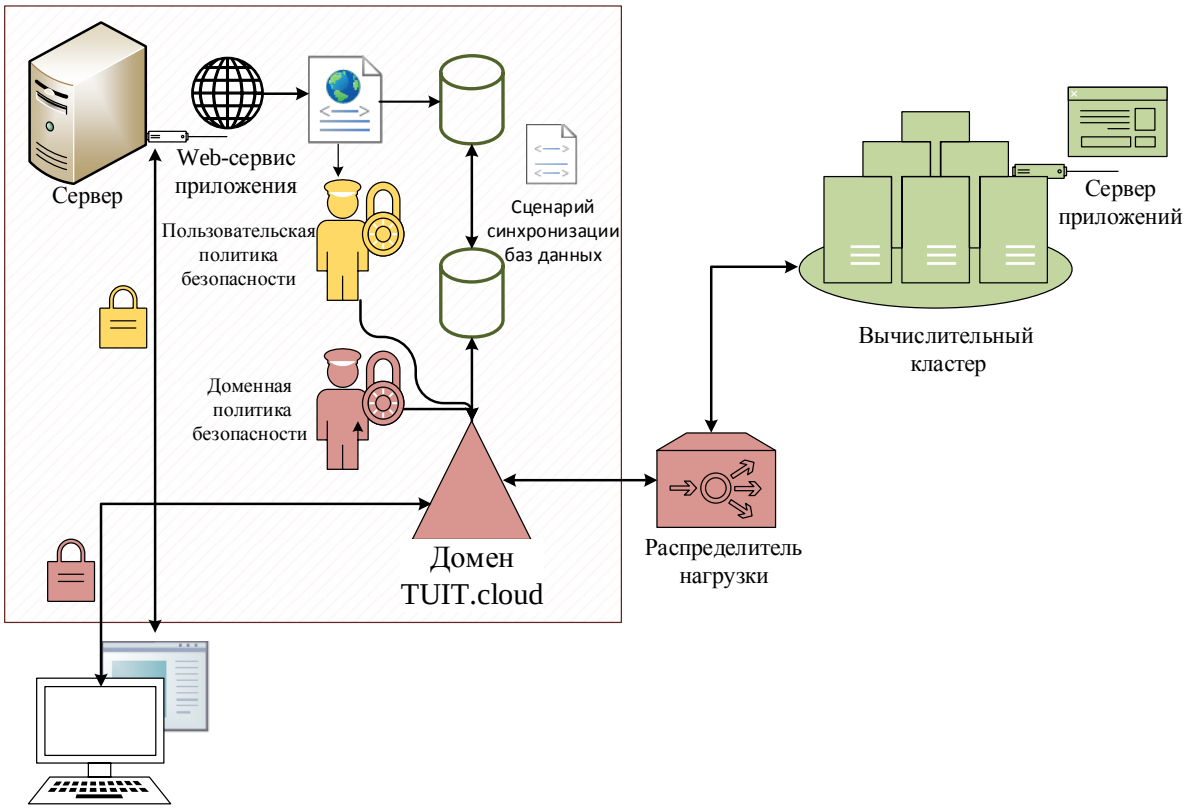


Рис. 3.1. Общая структура реализации частного облака образовательного учреждения

Разработка веб-службы для предоставления унифицированной объектной модели.

Web-сервис приложения реализует взаимодействие пользователя системы с внутренним механизмом облака посредством взаимодействия через API предоставляемое web-службой (Рис. 3.2). Все передаваемые данные шифруются и передаются по защищенному каналу.

Алгоритм работы:

- на сервис приходит запрос от приложения получить определенный набор данных;
- далее сервер создает файл сценария и реализует изменения в таблицах базы данных приложения;
- файл сценария создает временную групповую политику безопасности и применяет ее к выбранной определенной группе домена с временем действия указанного в приложении.

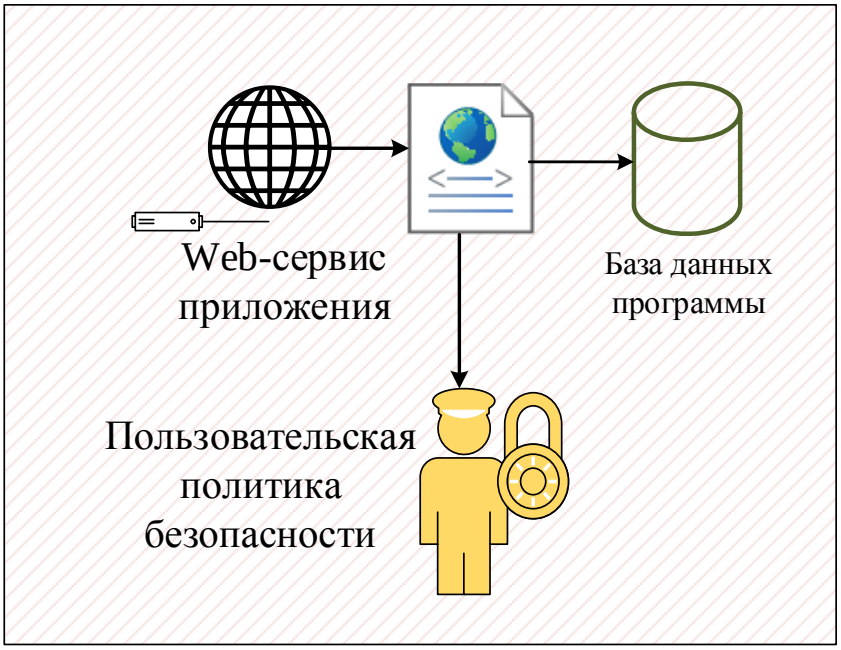


Рис. 3.2. Общая структура работы Web-сервиса

Разработка приложения для организации предоставления услуг в частном облаке образовательного учреждения.

Так как при реализации частного облака образовательного учреждения была использована продукция Microsoft, то и при разработке приложения был использован язык C#. Основными доводами в пользу использования C# являются:

- использование платформы .NET, что позволяет сократить размер программы, а также увеличить скорость работы программы;

- предоставление удобных инструментов для организации программного интерфейса и web-сервиса;
- предоставление некоторых базовых наборов библиотек для организации безопасности и внутренние доменные механизмы взаимодействия;

Основной задачей на данном этапе является:

- организация простого пользовательского интерфейса;
- сбор необходимых данных для web-сервиса;
- организация безопасной передачи персональных данных пользователя и файла настроек программы;

Опишем алгоритм работы программы:

- пользователь (должен входить в группу преподавателей или администраторов) запускает ярлык программы на рабочем столе созданный групповой политикой безопасности домена;
- далее пользователь использует свой доменный логин и пароль для входа в систему (Рис. 3.3);
- после процесса успешной авторизации настраивает набор программных приложений для проведения занятия указывая группы к которой следует применить настройки и время проведения занятия
- программа по установленным параметрам формирует запрос и отсылает его на web-сервис, который создает временную политику безопасносности и применяет ее к группам указанных пользователем;

В приложении имеется три вида ошибок (рис. 3.4), при возникновении которых выдаётся исключение, это:

- При попытке войти в систему оставив поля логина и пароля пустыми;
- При попытке ввода не правильного логина/пароля;

- При отсутствии подключения или в соответствии с ограничениями действующими на учетную запись пользователя;

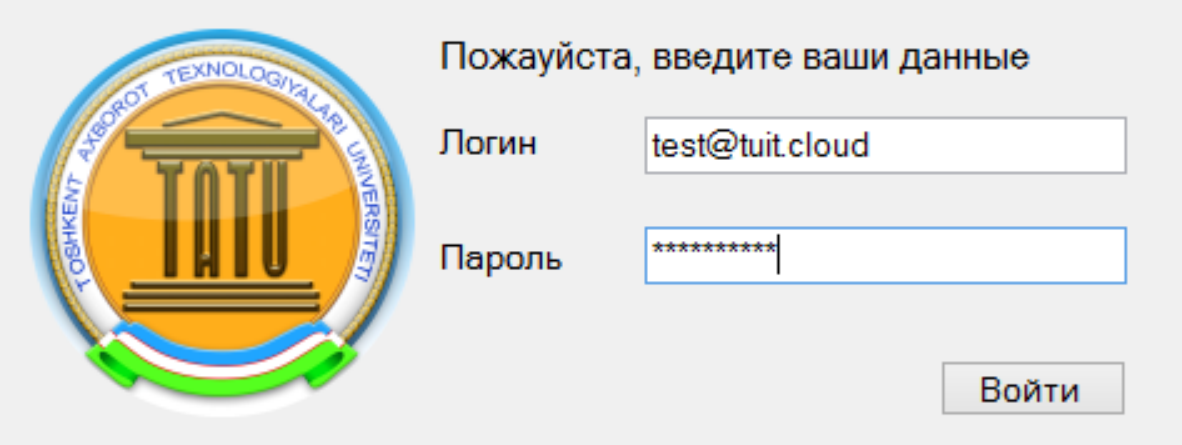


Рис.3.3. Форма входа с указанием логина и пароля

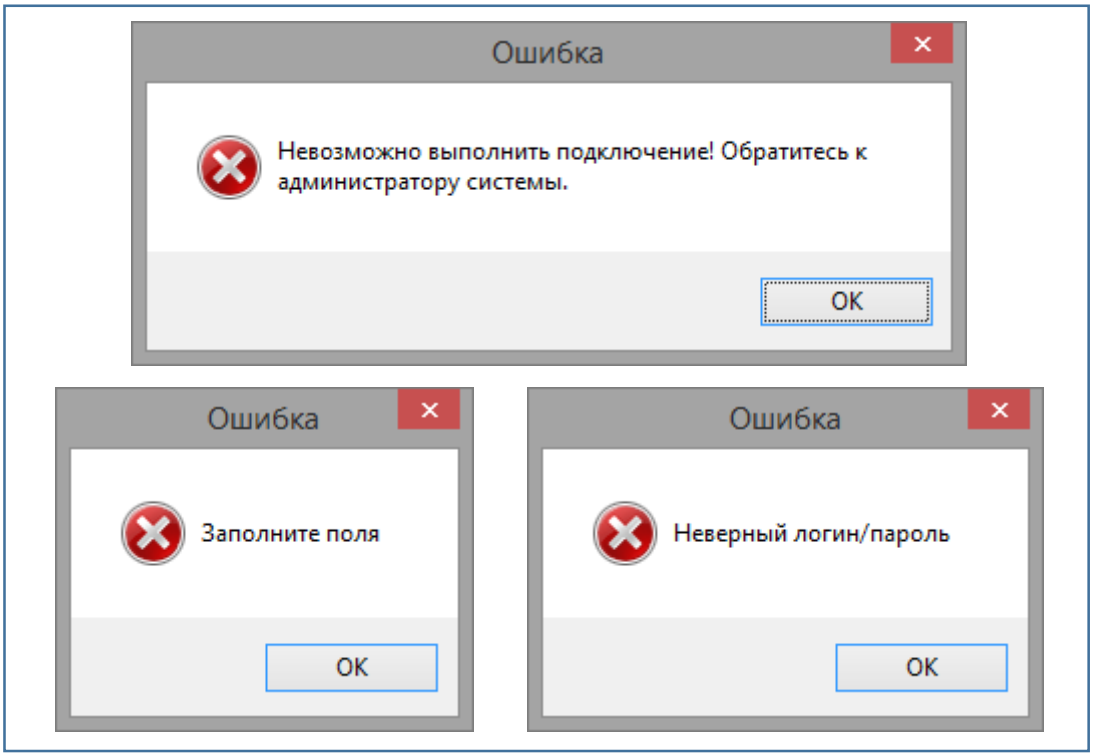


Рис. 3.4. Виды ошибок программы

При успешном входе программы пользователю отображается главное окно программы. Если пользователь впервые заходит в программу то ему отображается страница на которой он должен согласиться с

условиями сервиса (внизу страницы) (Рис. 3.5). Заблокированы следующие пункты меню: Программы, Сервисы, Настройка, Служба поддержки. После соглашения пользователь получает полный доступ к программе. При дальнейшем использовании приложения в самом начале показывается справка. В справке отображаются основные возможности и методы взаимодействия пользователя и программы (Рис.3.6). Далее пользователь нажимает на интересуемый его пункт меню в боковой части и выбирает программы которые требуется применить к группе. В верхней части меню есть возможность указать группы к которым применима политика назначения программ, указать дату и время действия политики.

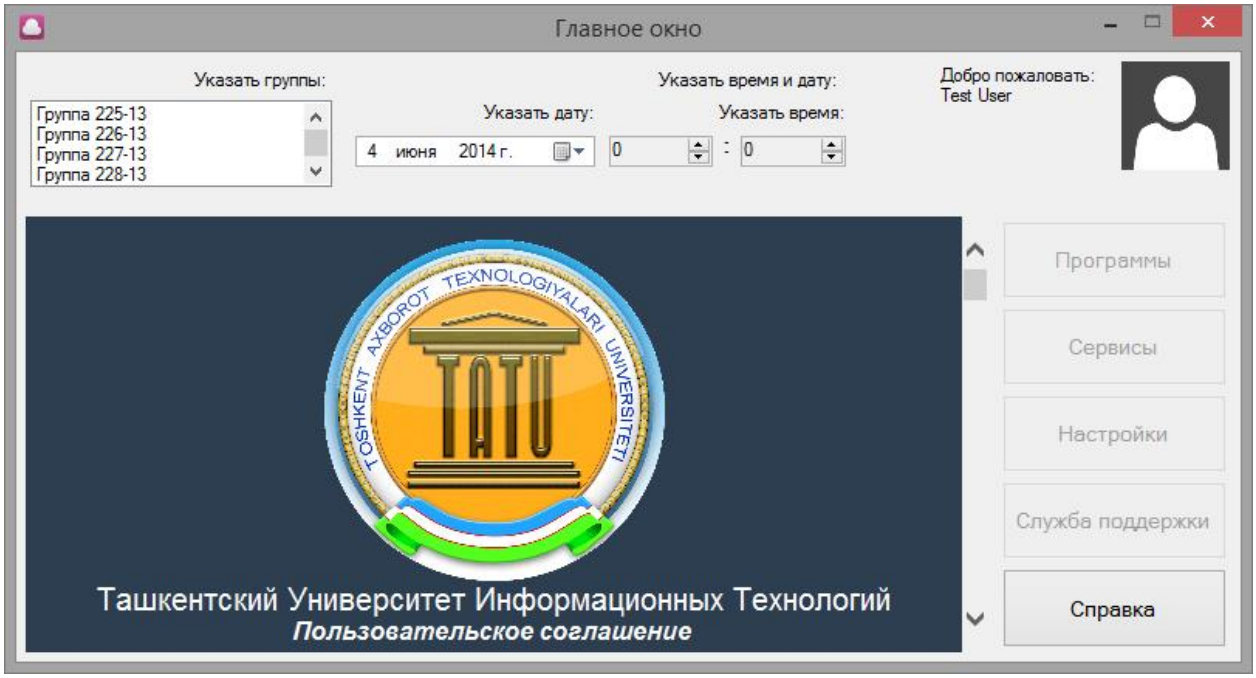


Рис. 3.5. Главное окно программы (первый запуск)

Пользовательский интерфейс делиться на три условных зоны:

- верхняя зона приложения (на Рис.3.7, помечена желтым цветом)
- центральная зона приложения (на Рис.3.7, помечена зеленым цветом)

- боковая зона приложения (на Рис.3.7, помечена синим цветом)

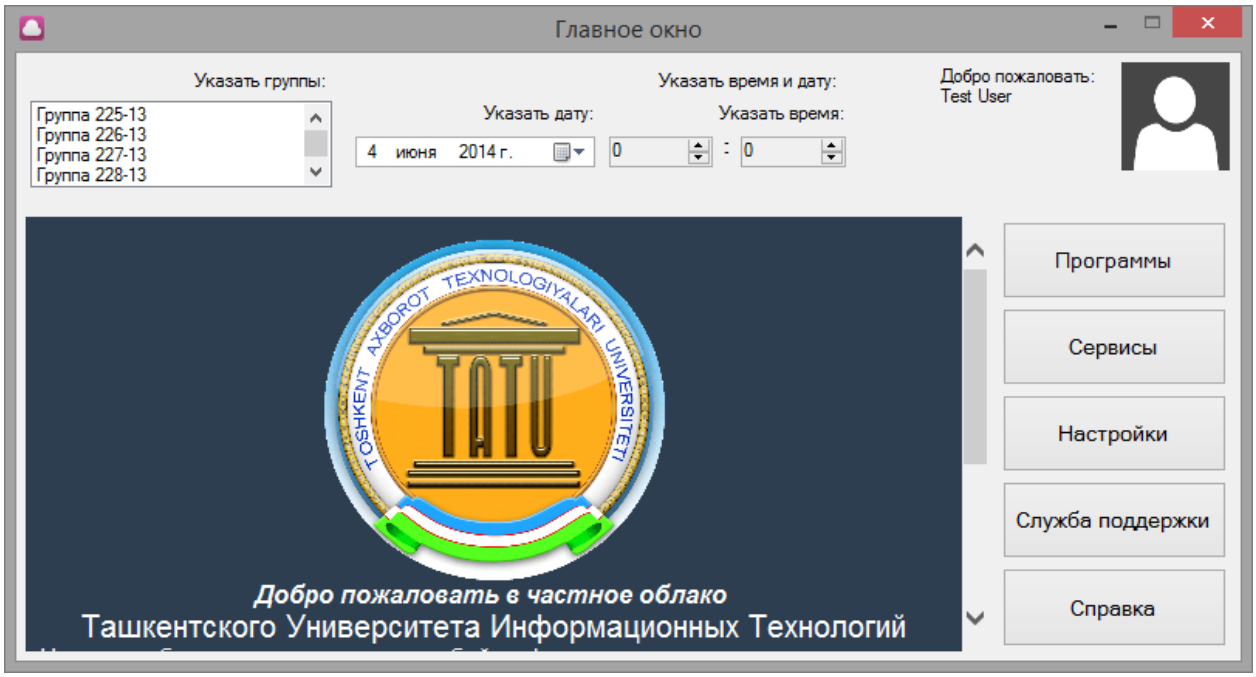


Рис. 3.6. Главное окно программы (пользователь принял условия лицензионного соглашения)

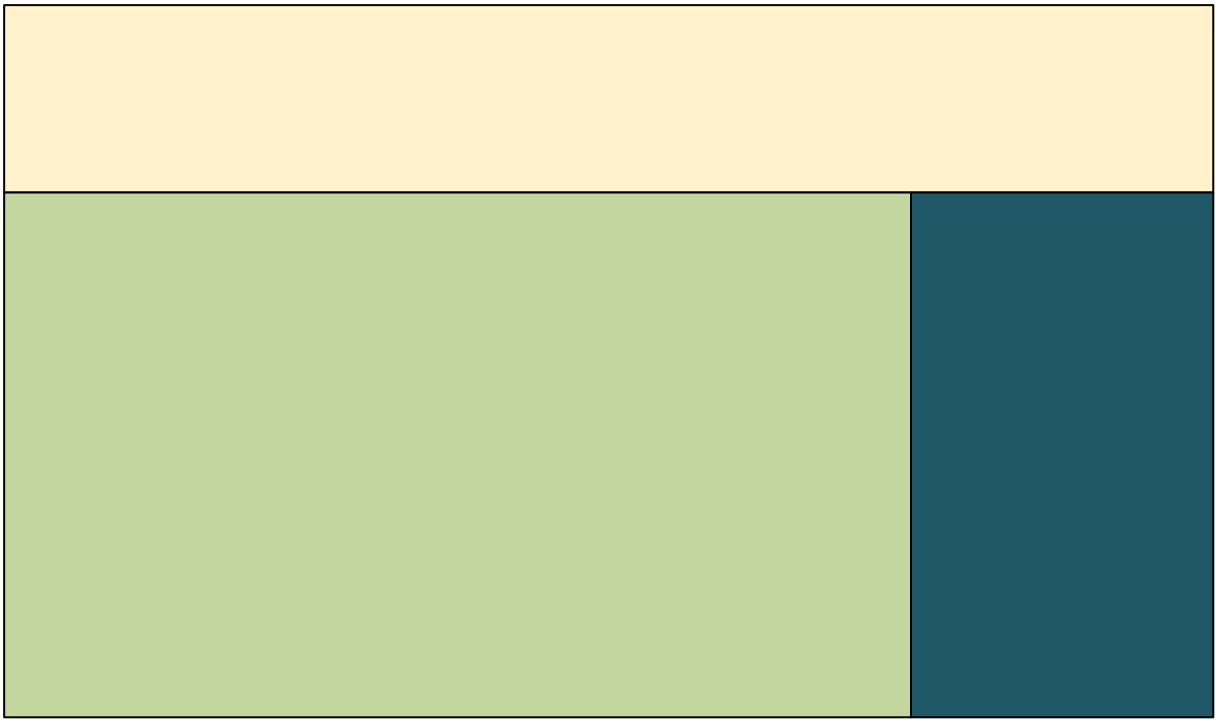


Рис.3.7. Условное разделение зон приложения

В верхней части имеются следующие возможности:

- указание групп обучающихся, которые проходят обучение у преподавателя
- указать дату на которую следует запланировать применение политики безопасности и время ее действия;
- также отображается информация о пользователе, его имя, фамилия и фото;

Центральная часть программы используется для отображения функционала доступного пользователю.

В боковой части отображается перечисление пунктов меню программы:

Пункт «Программы» - при нажатии на данный пункт в центральной части отображаются, программы доступные пользователю. Программы делятся на следующие виды:

- «Офисные программы» - пакет офисных программ от Microsoft, LibreOffice и набора программ для чтения электронных книг в формате PDF и Djvu.
- «Графические программы» - пакет программ от Adobe, пакет программ от CorelDraw, пакет программ сторонних производителей;
- «Программы для разработки» - представлены набором программ для разработки приложений на ООП высокого уровня;
- «Среды программирования» - представлены набором программ для запуска скомпилированных приложений.

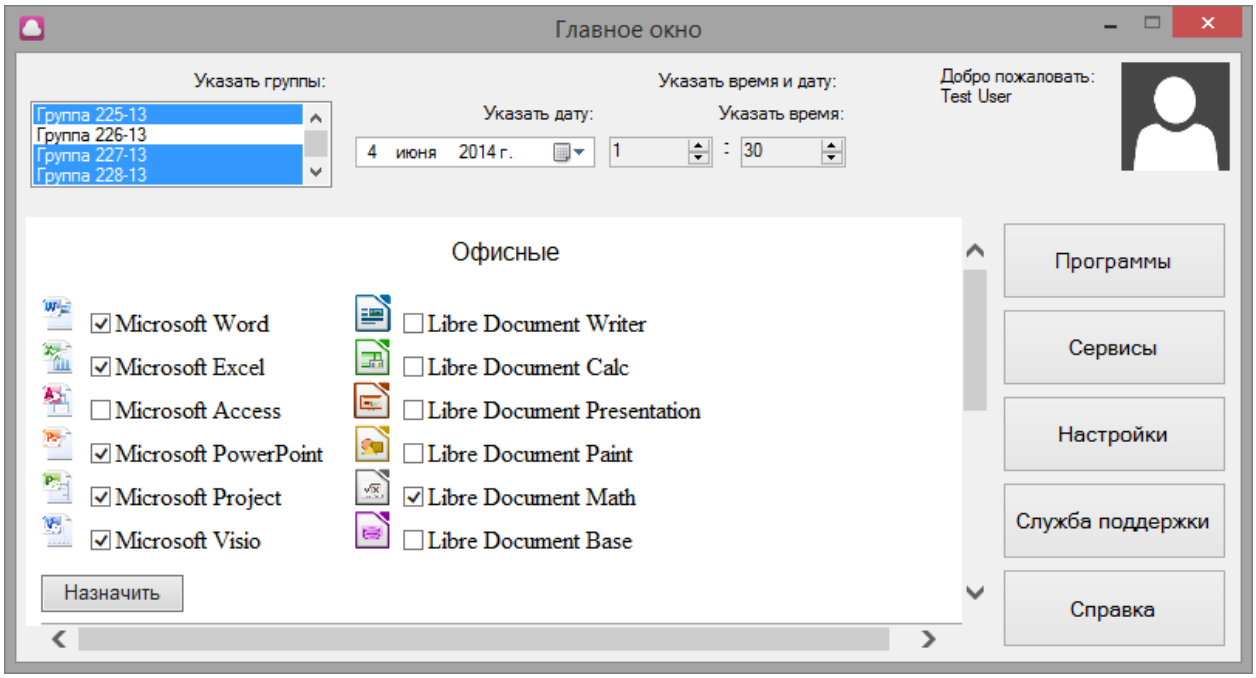


Рис.3.8. Принцип работы программы

Пункт «Сервисы» - при нажатии на данный пункт в центральной части отображаются, различные сервисы доступные пользователю. Сервисы делятся на следующие виды:

- «CMS» (*англ.* content manager system) - представлены набором известных систем для контроля контента;
- «CMF» (*англ.* content manager framework) - представлены набором фреймворков с помощью которых реализуются CMS.
- «Библиотеки к программам» - представлены набором библиотек для подключения к приложениям разрабатываемых студентами в процессе обучения;
- «Скрипты» - представлены набором скриптов для автоматизации процесса обучения;
- «Локальные Web-сервера» - представлены набором программ для организации локального запуска разработанных web-приложений;

Пункт «Настройки» - при нажатии на данный пункт в центральной части отображаются, различные настройки доступные пользователю.

Настройки включают в себя следующие виды: «Изменение личных данных» (Кроме пароля, он изменяется через доменные политики безопасности администратором системы), «Настройка задач сервиса», «Добавление групп», «Монитор приложений».

Выводы по третьей главе

В заключении по главе следует сказать, что на основе результатов анализа проблемной области были выделены:

1. цели и задачи при построении уровня безопасности частного облака образовательного учреждения
2. организована разработка свода правил для организации безопасности частного облака образовательного учреждения.
3. основании алгоритма разработанного во второй главе магистерской работы было организовано частное облако модели SaaS
4. разработаны прикладное приложение для клиентской стороны, база данных программы, набор функциональных скриптов и web-сервис.

Разработанное облако и программный продукт являются одним целым и используются совместно.

В результате исследования были выявлены и решены следующие требования:

1. отказоустойчивость при возникновении ошибок;
2. отказоустойчивость при попытках взлома;
3. блокирование попыток взлома;
4. предоставление удобного интерфейса для использования ресурсами облака;

ЗАКЛЮЧЕНИЕ

В результате выполнения работы по данной теме были проделаны следующие этапы:

Рассмотрены особенности разрабатываемой системы и требования к ней. Проанализированы подходы позволяющие выбрать уровень виртуализации системы при построении облака для наиболее лучшего использования аппаратных средств. Также рассмотрены различные виды угроз и стратегии безопасности.

При проведении анализа было выявлена необходимость в использовании высокоразвитой стратегии управления вычислительными ресурсами. По проведенному анализу требуется организовать построения масштабируемых систем и использовать заранее заданные единицы сетей, хранилищ и вычислительных систем. Средства управления должны быть запрограммированы так, чтобы учитывать единицы масштабирования, время подготовки и развертывания систем, текущие и прошлые тенденции использования вычислительных мощностей (поскольку может потребоваться развертывание дополнительных единиц).

Проведен анализ диалога с потребителем, с целью оценить новые и изменяющиеся инициативы, которые могут повлиять на сложившиеся тенденции использования вычислительных мощностей. План наращивания вычислительных мощностей крайне важен для эффективного управления вычислительными ресурсами. А при анализе угроз была выявлена специфика обработки персональных данных в виртуальной среде.

На основании требовании и различного уровня спецификаций произведена выработка методов и правил защиты в частном облаке, а также создание частного облака типа сервиса предоставления программного обеспечения как услуги. В ходе работы были обнаружены проблемы безопасности которым подвергаются облачные системы на

различных уровнях организации. Также был создан программный комплекс решающий круг поставленных проблем.

В состав программного комплекса разработаны прикладное приложение для клиентской стороны, база данных программы, набор функциональных скриптов и web-сервис который предоставляет удобное API.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Постановлением Президента РУз от 21.03.2012 г. за № ПП-1730 «О мерах по дальнейшему внедрению и развитию современных информационно-коммуникационных технологий»
2. Доклад Президента Республики Узбекистан Ислама Каримова на заседании Кабинета Министров, посвященном основным итогам 2013 года и приоритетам социально-экономического развития на 2014 год. – 23.03.2014
3. Lynn S. - Windows Server 2012 - Up and Running, Издательство O'Relly, Год 2012, 258стр.
4. Tulloch M. - Training Guide Installing and Configuring Windows Server 2012, Издательство Microsoft Press, Год 2012, стр.609
5. Carvalho L. - Windows Server 2012 Hyper-V Cookbook, Издательство Packt Publishing, Год 2012, стр.304.
6. William Stanek - Microsoft PowerShell, VBScript, and JScript Bible, Издательство Willey Publishing, Год 2009, стр.915
7. Don Jones - VBScript WMI and ADSI Unleashed, Издательство SAMS, Год 2007, стр.574.
8. Ben Gan I. - Microsoft SQL Server 2012 T-SQL Fundamentals, Издательство Microsoft Press, Год 2012, стр.442.
9. Моримото Р., Ноэл М., Драуби О., Мистри Р., Амарис К. - Microsoft Windows Server 2008 R2. Полное руководство, Издательство SAMS, Год 2011, стр.1457.
10. Siddaway R. - PowerShell and WMI, Издательство Manning Publications, Год 2012, стр.550.
11. Рахимов Д.К., Асташев Е.А. Модели развертывания и обслуживания в «облаке». Сборник докладов Республиканской научно-технической конференции молодых ученых, исследователей, магистрантов и студентов. Часть 1, Издательство ТУИТ, Год 2013.

12. Асташев Е.А. Применение современных «облачных» технологий в образовании. Сборник докладов Научно-методической конференции Ташкентского Университета Информационных Технологий и его филиалов. I Том, Издательство ТУИТ, Год 2014.
13. Goodacre John. Hardware accelerated Virtualization in the ARM Cortex™ Processors. 2011. xen.org/files/xensummit_oul11/nov2/2_XSAsia11_JGoodacre_HW_accelerated_virtualization_in_the_ARM_Cortex_processors.pdf
14. Hardware-assisted Virtualization with the MIPS® Virtualization Module. 2012. www.mips.com/application/login/login.dot?product_name=/auth/MD00994-2B-VZMIPS-WHT-01.00.pdf
15. Hypervisor/Sun4v Reference Materials. 2017. kenai.com/projects/hypervisor/pages/ReferenceMaterials
16. Intel® Virtualization Technology / F. Leung, G. Neiger, D. Rodgers et al. // Intel Technology Journal. 2006. Vol. 10. www.intel.com/technology/itj/2006/v10i3/
17. McGhan Harlan. The gHost in the Machine: Part 1 // Microprocessor Report. 2007. mpronline.com
18. McGhan Harlan. The gHost in the Machine: Part 2 // Microprocessor Report. 2007. mpronline.com
19. McGhan Harlan. The gHost in the Machine: Part 3 // Microprocessor Report. 2007. mpronline.com
20. Popek Gerald J., Goldberg Robert P. Formal requirements for virtualizable third generation architectures // Communications of the ACM. Vol. 17. 1974.
21. Southern Gabriel. Analysis of SMP VM CPU Scheduling. 2008. cs.gmu.edu/~hfoxwell/cs671projects/southern_v12n.pdf

22. Yang Rongzhen. Virtual Translation Lookaside Buffer. 2008. www.patentlens.net/patentlens/patent/US_2008_0282055_A1/en/.
23. Software techniques for avoiding hardware virtualization exits / Ole Agesen, Jim Mattson, Radu Rugina, Jeffrey Sheldon // Proceedings of the 2012 USENIX conference on Annual Technical Conference. USENIX ATC'17. Berkeley, CA, USA : USENIX Association, 2017. P. 35-35. www.usenix.org/system/files/conference/atc12/atc12-final158.pdf
24. Poon Wing-Chi, Mok A.K. Improving the Latency of VMExit Forwarding in Recursive Virtualization for the x86 Architecture // System Science (HICSS), 2012 45th Hawaii International Conference on. 2017. P. 5604-5617.
25. Osisek D. L., Jackson K. M., Gum P. H. ESA/390 interpretive execution architecture, foundation for VM/ESA // IBM Syst. J. — 1991— V. 30, No 1. — Pp. 34–51. — ISSN: 0018-8670. —DOI: 10.1147/sj.301.0034.
26. Andy Glew. SIE. — semipublic.comp-arch.net/wiki/SIE
27. The Turtles Project: Design and Implementation of Nested Virtualization / Muli Ben-Yehuda [et al.] //. — 2010. — P. 423–436. www.usenix.org/event/osdi10/tech/full_papers/Ben-Yehuda.pdf

Приложение 1

Обозначения и сокращения

Кластер — программно-аппаратный комплекс с массивно параллельной архитектурой предназначенный для решения всевозможных вычислительных задач математической физики, геологии, химии и множества других.

Узел — стандартная вычислительный единица Кластера, обычно представляющая собой сервер размером 1U или блок т.н. blade-серверов.

Управляющий узел (УУ) — один или несколько выделенных серверов в составе Кластера. УУ обеспечивает целостною работу кластера при помощи ряда функционирующих на нем Сервисов.

СХД — система хранения данных.

InfiniBand Фабрика — единый комплекс оборудования InfiniBand.

Baseboard Management Controller (BMC)— сервисный процессор в составе Узла позволяющий осуществлять гипервизоринг по протоколам HTTP, IPMI 7.0, SNMP и прочим. Также BMC предоставляет функции KVMoIP и VirtualMedia.

Сервис — отдельная служба выполняющая конкретную задачу Кластера. Как правило, за работу сервиса отвечает один или несколько системных служб (демонов).

Виртуализация в вычислениях — процесс представления набора вычислительных ресурсов, или их логического объединения, который даёт какие-либо преимущества перед оригинальной конфигурацией.

Хозяин (*англ.* host**)** — аппаратная система, на которой запущен гипервизор виртуальных машин или симулятор.

Гость (*англ.* guest**)** — виртуальная или моделируемая система, запущенная под управлением гипервизора или симулятора. Также иногда именуется как целевая система (англ. target system).

Приложение 2

КОД ОПЕРАЦИИ ПО ПОЛУЧЕНИЮ РЕЗУЛЬТАТА ПРОВЕРКИ ИМЕНИ ПОЛЬЗОВАТЕЛЯ И ПАРОЛЯ

```
1. public OperationResult<IList<DistrictDictionaryItem>>
   GetDistrictDictionary(string credential){
2.
   ServicePointManager.ServerCertificateValidationCallback =
3.                         (sender, certificate, chain,
   sslPolicyErrors) => true;
4.
5.             if (CheckAuthorization(credential).Result !=
   OperationResultEnum.Success)
6.             {
7.                 return
   OperationResult<IList<DistrictDictionaryItem>>.FromOperation
   Result(result);
8.             }
9.
10.                 var res = new
   OperationResult<IList<DistrictDictionaryItem>>();
11.                 try
12.                 {
13.                     var resultDictionary = new
   List<DistrictDictionaryItem>();
14.                     using (var model =
   ModelFactory.CreateTCDataModel())
15.                     {
16.                         resultDictionary.AddRange(
17.                             model.Districts.OrderBy(c =>
   c.NameRu)
18.                                 .Select(di => new
   DistrictDictionaryItem { Id = di.Id, NameRu = di.NameRu,
   NameUz = di.NameUz, Password = di.Password}));
19.                     }
20.                     res.Value = resultDictionary;
21.                     res.Result =
   OperationResultEnum.Success;
22.                 }
23.                 catch (Exception e)
24.                 {
25.                     res.ErrorMessage = e.Message;
26.                     res.Result =
   OperationResultEnum.Failed;
27.                     Logger.Log("Exception in
   GetDistrictDictionary", e);
28.                 }
29.                 return res;
```

```
30.             }
```