МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

На правах рукописи

УДК 004.051

ЖЕБРАК СЕМЁН МИХАЙЛОВИЧ

«Разработка инфраструктуры виртуализации автоматизированных экономических информационных систем»

5А330501 – Компьютерный инжиниринг (Компьютерный инжиниринг)

Диссертация

на соискание академической степени магистра

Допущен к защите
Зав. Кафедрой «КС»
_____Назаров А.И.
<<___>>_____2015г

Научный руководитель
к.т.н.доцент Назаров А.И.
_____
<<___>>_____2015г

Ташкент – 2015

ГОСУДАРСТВЕННЫЙ КОМИТЕТ СВЯЗИ, ИНФОРМАТИЗАЦИИ И ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Факультет «Компьютерный инжиниринг»

Кафедра «Компьютерные системы»

Учебный год – 2013 - 2015

Студент магистратуры Жебрак С.М.

Научный руководитель Назаров А.И.

Специальность «Проектирование компьютерных систем»

АННОТАЦИЯ МАГИСТЕРСКОЙ ДИССЕРТАЦИИ

Развитие современных информационных и коммуникационных технологий (ИКТ) имеет целенаправленную тенденцию к интенсификации и диверсификации, охватывая все новые отрасли экономики. В настоящее время персональный компьютер является необходимым инструментом практически во всех коммерческих и государственных организациях и предприятиях. Однако персональный компьютер в большинстве случаев не отвечает современным требованиям к масштабируемости и управляемости, а также вызывает сложности, связанные с техническим обслуживанием и единой политикой информационной безопасности. Как следствие, применение большого числа персональных компьютеров в корпоративной информационной системе компании является критическим фактором.

Кроме того, использование персональных компьютеров обуславливает постоянный рост затрат на их эксплуатацию: управление, установку обновлений программного обеспечения, резервное копирование информации,

техническую поддержку. Большая часть этих затрат - расходы на оплату обслуживающего персонала, которые, как правило, превышают затраты на приобретение оборудования, программного обеспечения и расходных материалов.

Однако, виртуализация серверов и рабочих станций ресурсов позволяет значительно сократить стоимость владения компьютерными системами. В результате компьютерные системы становятся доступными не только средним, но и небольшим компаниям.

Диссертационная работа посвящена разработке оптимальной инфраструктуры виртуализации АЭИС с использованием основных возможностей VPS/VDS и VDI на предприятиях Узбекистана. Выполнен анализ состояния и проблем внедрения технологий виртуализации. В ходе выполнения диссертационной работы сравнивались возможности между виртуализации на уровне операционной системы и полной виртуализации. Представлен расчёт основных оптимальных требований к аппаратному обеспечению терминальных серверов и виртуальных серверов.

В диссертационной работе поставлены следующие цели исследования:

- выявление оптимальных средства виртуализации АЭИС;
- определение оптимальных требований к аппаратному обеспечению для виртуализации серверов;
- определение оптимальных требований к аппаратному обеспечению терминальных серверов;
- разработка наиболее оптимальной инфраструктуры виртуализации для эффективного использования АЭИС в различных сферах Национальной экономики Республики Узбекистан.

Объектом исследования является технология и инфраструктура виртуализации подходящая для повышения эффективности АЭИС с учетом особенностей национальной экономики Узбекистана.

Предметом исследований являются средства инфраструктуры виртуализации серверов, рабочих станций и сетей.

Методами исследования наблюдение за развитием компаний на мировом рынке, изучались результаты аналитических исследований и проводились эксперименты на серверном оборудовании. Тестировались виртуальные среды с помощью программ тестирования производительности. А также расчёт требований к аппаратному обеспечению производился математическими методами.

Предложена оптимальная инфраструктура виртуализации АЭИС с использованием основных возможностей VPS/VDS и VDI для предприятий Узбекистана.

Предметом научной новизны полученных результатов являются следующие:

- определение требований и ожиданий узбекских клиентов от технологий виртуализации;
- исследование решений и технологий в области виртуализации серверов и рабочих мест в мире;
- разработка концепции внедрения средств виртуализации в организациях Узбекистана, учитывая специфику национальной экономики;
- сравнение возможностей, производительности между виртуализации на уровне операционной системы и полной виртуализации с помощью программ тестирования производительности.

Практическая значимость работы. Результаты исследования позволят выявить востребованную и необходимую для внедрения инфраструктуру технологии виртуализации серверов и рабочих мест на предприятиях в различных сферах экономики Узбекистана.

Магистерская диссертация состоит из введения, трёх глав и заключения, изложена на 94 страницах машинописного текста, содержит 24 рисунка.

Основными результатами работы являются созданная инфраструктуры виртуализации АЭИС с использованием основных возможностей VPS/VDS и VDI средствами opensource.

Научный руководитель _____ Назаров А.И.

Студент магистратуры _____ Жебрак С.М.

THE STATE COMMITTEE OF COMMUNICATION, INFORMATION AND TELECOMMUNICATION TECHNOLOGIES OF REPUBLIC UZBEKISTAN

THE TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES

Faculty "Computer Engineering"

Department of "Computer Systems"

Academic year - 2012 - 2014

The student of magistracy Jebrak S.M.

The supervisor of studies Nazarov A.I.

Specialty " Computer Systems Design"

THE SUMMARY OF THE MASTER'S DISSERTATION

The development of modern information and communication technologies (ICT) has a tendency to deliberate intensification and diversification, covering new sectors of the economy. At present, the personal computer is an essential tool in almost all commercial and government organizations and enterprises. However, a personal computer, in most cases does not meet modern requirements for scalability and manageability, as well as causing difficulties associated with maintenance and a single security policy. As a consequence, use of a large number of personal computers in the corporate information system of the company is a critical factor.

In addition, the use of personal computers causes a permanent increase in costs for their operation: management, installation of software updates, backup information, technical support. Most of these costs - expenses for staff, which usually exceeds the cost of purchase of equipment, software and consumables.

However, server virtualization and desktop resources can significantly reduce the cost of ownership of computer systems. As a result, computer systems are accessible not only to the average, but also small companies.

The thesis is devoted to the development of optimal virtualization infrastructure AEIS using basic features VPS / VDS and VDI enterprises of Uzbekistan. The analysis of the status and problems of implementation of virtualization technologies. In the course of the thesis were compared between the virtualization capabilities in the operating system and full virtualization. It presents the calculation of basic optimal hardware requirements for terminal servers and virtual servers.

This thesis studies the following objectives:

- Identification of optimal virtualization AEIS;
- To determine the optimal hardware requirements for server virtualization;
- To determine the optimal hardware requirements for terminal servers;
- Development of optimal virtualization infrastructure for efficient use of AEIS in various spheres of the national economy of the Republic of Uzbekistan.

The object of research is the technology and infrastructure virtualization is suitable for efficiency AEIS taking into account peculiarities of the national economy of Uzbekistan.

The subject of research are the means of infrastructure virtualization of servers, workstations and networks.

Methods of research monitoring of the development of companies in the global market, studied the results of analytical studies and experiments were conducted on the server hardware. Tested virtual environment using software performance testing. As well as the calculation of hardware requirements produced by mathematical methods.

The optimum infrastructure virtualization AEIS using basic features VPS / VDS and VDI for enterprises of Uzbekistan.

The subject of scientific novelty Scientific novelty of the results are as follows:

- Define the requirements and expectations of customers from the Uzbek virtualization technologies;
- Research solutions and technologies in the field of server virtualization and jobs in the world;
- Development of the concept of introduction of virtualization in organizations of Uzbekistan, given the specificity of the national economy;
- A comparison of features, performance between virtualization at the operating system and full virtualization with software performance testing.

The practical significance of the work. The study reveals the demand and the necessary infrastructure for the implementation of server virtualization technology and jobs in various sectors of the economy of Uzbekistan.

Master's thesis consists of an introduction, three chapters and a conclusion, stated on 94 pages of typewritten text, contains 24 drawings.

The main result is to create an infrastructure virtualization AEIS using basic features VPS / VDS and VDI tools opensource.

The supervisor of studies	_____	Nazarov A.I.
The student of a magistracy	_____	Jebrak S.M.

Оглавление

Введение

Обоснованность темы диссертации. Виртуализация - это технология, обеспечивающая абстрагирование процессов и их представления от вычислительных ресурсов.

Современное состояние общества и особенно его экономической сферы характеризуется процессами глобализации и информатизации, которые предопределили трансформацию всей социальной жизни.

В своем докладе на заседании Кабинета Министров, посвященном итогам социально-экономического развития в 2014 году и важнейшим приоритетным направлениям экономической программы на 2015 год, Президент Республики Узбекистан Ислам Каримов особо обратил внимание вопросам развития сферы связи, информатизации и телекоммуникационных технологий [1].

Развитие информационно-коммуникационных технологий (ИКТ) и их активное применение в отраслях экономики создали предпосылки для формирования нового типа общества – информационного. Осознавая особую и важную роль ИКТ в экономике и обществе, в 2013 году была принята Комплексная программа развития Национальной информационно-коммуникационной системы Республики Узбекистан на период 2013-2020 годы [1].

Актуальность. Развитие ИКТ связано с использованием их в информационных системах различных экономических объектов - предприятий, организаций различных сфер национальной экономики. Деятельность отдельных людей, коллективов и организаций сейчас все в большой степени зависит от их информированности и способностей эффективно использовать имеющуюся информацию. Это привело к возникновению следующих проблем, которые на основании разработки инфраструктуры виртуализации решены в диссертационной работе:

- Неэффективное использование ресурсов инфраструктуры.
- Повышение затрат на физическую инфраструктуру.
- Увеличение расходов на управление ИТ.
- Недостаточно надежная система аварийного восстановления и защиты в критических ситуациях.
- Трудоёмкость и сложность техобслуживания инфраструктуры.

ИКТ дало возможность объединить в одну связанную систему миллионы компьютеров в разных странах мира, структурировать глобальную базу данных, хранить и передавать информацию, а в конечном итоге - создавать и структурировать пространственные потоки капитала, денег, товаров. За счет виртуализации обеспечивается существенная экономия на аппаратном обеспечении, облуживании, повышается гибкость ИТ-инфраструктуры, упрощается процедура резервного копирования и восстановления после сбоев. Виртуальные машины, являясь независимыми от конкретного оборудования единицами, могут распространяться в качестве предустановленных шаблонов, которые могут быть запущены на любой аппаратной платформе поддерживаемой архитектуры.

Объектом исследования является технология и инфраструктура виртуализации подходящая для повышения эффективности АЭИС с учетом особенностей национальной экономики Узбекистана..

Предметом исследований являются средства виртуализации серверов, рабочих станций и сетей инфраструктуры АЭИС.

Целью настоящей работы является разработка наиболее оптимальной инфраструктуры виртуализации для эффективного использования АЭИС в различных сферах Национальной экономики Республики Узбекистан.

Задачи. Исходя из цели поставлены следующие задачи:

1. Изучить архитектуру экономических систем и требования к ней;
2. Изучить виды виртуализации;
3. Выбрать оптимальные средства виртуализации серверов;
4. Разработать рекомендации по расчёту оптимальных требований к аппаратному обеспечению для виртуализации серверов;
5. Разработать рекомендации по расчёту оптимальных требований к аппаратному обеспечению терминальных серверов;
6. Спроектировать инфраструктуру виртуализации автоматизированной экономической системы.

Научная новизна. Проектирование оптимальной инфраструктуры виртуализации АЭИС для субъектов хозяйствования Узбекистана. Применение технологии виртуализации VPS/VDS и VDI повысит эффективность АЭИС.

Основной задачей диссертационной работы является разработка инфраструктуры виртуализации автоматизированных экономических информационных систем с учётом национальной экономики Республики Узбекистан.

Гипотеза исследования. Создание концепции внедрения средств виртуализации серверов и рабочих мест в организациях Узбекистана, учитывая специфику национальной экономики.

Обзор литературы по теме исследования. В книгах Исаев Г. Н. «Информационные системы в экономике», Титоренко Г.А. «Автоматизированные информационные технологии в экономике», Банк В.Р., Зверев В.С. «Информационные системы в экономике». Анализируются общие вопросы разработки информационного обеспечения систем автоматизированного проектирования организационно-технологических

задач строительства и автоматизированного управления проектированием. В них рассмотрены методы разработки и моделирования информационного обеспечения; описаны средства построения баз данных нормативной и технико-экономической информации; приведены сведения о проектировании систем управления базами данных. Также в книге даны рекомендации по разработке проектной и эксплуатационной документации информационного обеспечения и защите информации. Кроме того, раскрываются теоретические основы проектирования экономических информационных систем на различных стадиях жизненного цикла. Рассматриваются методы и средства канонического и индустриального проектирования экономических информационных систем, а также управления процессом проектирования.

В книге Chris Wolf, Erick M. Halter . Virtualization: From the Desktop to the Enterprise приводится пошаговое руководство по установке Terminal Server внутри виртуальной машины и предоставлению удаленного доступа к получившемуся сервису. Так же даются рекомендации по установке Remote Desktop Web Connection и изменениям, вносимых в Firewall для того, чтобы клиенты могли использовать терминальный сервер.

В книгах Митч Таллок «Решения Майкрософт для виртуализации: от настольного компьютера до центра обработки данных» и «Создание всесторонней комплексной стратегии виртуализации» приводится пошаговое руководство по установке и настройке Virtual Server 2005 R2 SP1 и System Center Virtual Machine Manager. Утверждается, что данная связка позволит централизованно создавать, мигрировать системы из физических в виртуальные, и управлять распределенными системами виртуализации. Приводится руководство по изолированию приложений, работающих в филиальных офисах, друг от друга с помощью виртуальных машин Microsoft Virtual Server 2005 R2 SP1 на Intel совместимом аппаратном обеспечении. Так же рассказывается о повышении совместимости с устаревшими

приложениями и способах приведения систем в соответствие регулирующим нормативными документами.

Характеристики методик, применённых в исследовании. За время работы над диссертацией, были использованы методики исследования: наблюдение за развитием компаний на мировом рынке, изучение результатов аналитических исследований и проведение экспериментов на серверном оборудовании. Тестирование виртуальных сред с помощью программ тестирования производительности, а также расчёт требований к аппаратному обеспечению производилось математическими методами.

Теоритическая значимость результатов исследования диссертации заключается в том, что сформулированные в ней положения и выводы можно использовать в дальнейшем при разработке инфраструктуры АЭИС в конкретной предметной области национальной экономики, а также экономики других стран.

Практическое значение результатов исследований диссертации заключается в том, что её положения могут быть использованы в целях повышения эффективности АЭИС, так как помимо энергосбережения и сокращения расходов благодаря более эффективному использованию аппаратных ресурсов, виртуальная инфраструктура обеспечит высокий уровень доступности ресурсов, более эффективную систему управления IT-инфраструктурой, повышенную безопасность и усовершенствованную систему восстановления в критических ситуациях.

Характеристика структуры работы. Диссертационная работа изложена на 94 страницах машинописного текста в том числе 3-х глав, содержит 23 рисунка.

Глава I. Системный обзор автоматизированных экономических информационных систем и средств их виртуализации

1. Понятие и архитектура экономических систем

Экономическая информационная система (ЭИС) представляет собой систему, функционирование которой во времени заключается в сборе, хранении, обработке и распространении информации о деятельности какого-то экономического объекта реального мира. Такая информационная система создается для конкретного экономического объекта и должна в определенной мере копировать взаимосвязи элементов объекта.

Автоматизированная экономическая информационная система (АЭИС) включает в себя совокупность технико-экономической информации, экономико-математических методов и моделей, технических, технологических и программных средств и специалистов, предназначенных для обработки информации и принятия управленческих решений.

Цели экономических информационных систем – обеспечить эффективное управление и обработку информации в различных объектах организационно-экономического профиля, взаимодействие информационных технологий со специалистами, использующими в сфере своей деятельности информационные технологии (персональные компьютеры и развитые средства коммуникации) для выполнения своих профессиональных задач и принятия управленческих решений.

ЭИС предназначены для решения задач обработки данных, автоматизации конторских работ, выполнения поиска информации и отдельных задач, основанных на методах искусственного интеллекта.

Задачи обработки данных обеспечивают обычно рутинную обработку и хранение экономической информации с целью выдачи (регулярной или по запросам) сводной информации, которая может потребоваться для управления экономическим объектом.

Автоматизация конторских работ предполагает наличие в ЭИС системы ведения картотек, системы обработки текстовой информации, системы машинной графики, системы электронной почты и связи.

Поисковые задачи имеют свою специфику, и информационный поиск представляет собой интегральную задачу, которая рассматривается независимо от экономики или иных сфер использования найденной информации.

На современном этапе развития ЭИС принято осуществлять разбиение системы на два группы подсистем: функциональные и обеспечивающие. Так, обеспечивающие подсистемы выделяются по элементному принципу, а функциональные – по структурному, функциональному, виду управляемого ресурса и т. д.

Таким образом, совокупность функциональных подсистем и связей между ними составляет функциональную архитектуру; а обеспечивающих подсистем - системную архитектуру ЭИС.

Содержание функциональных подсистем зависит от уровня ЭИС и характера объекта управления; содержание обеспечивающих подсистем является стандартными для всех систем. [23]

Функциональная часть ЭИС, состоящая в свою очередь, из отдельных подсистем, представляет собой способы реализации функций управления и методы решения управленческих задач, что создает условия для выполнения и достижения целей системы управления. Функциональные подсистемы АЭИС информационно обслуживают определенные виды деятельности экономической системы и (или) функции управления. Интеграция функциональных подсистем в единую систему достигается за счет создания и функционирования обеспечивающих подсистем.

Рис. 1 Состав экономической информационной системы.

Обеспечивающие подсистемы ЭИС, состоят, в свою очередь, из отдельных подсистем и более развиты по сравнению с аналогичной частью традиционных систем управления.

Обеспечивающие подсистемы ЭИС представляют собой комплекс методов, средств, инструктивных и законодательных материалов, необходимых для работы функциональных подсистем. Сами по себе они не решают непосредственно задачи в ЭИС, но обеспечивают их решение в организационном, техническом, программном и других отношениях.

Выбор оптимальной структуры ЭИС – решающий фактор обеспечения ее жизнеспособности и эффективности. Оптимальной является такая структура, которая наиболее полно обеспечивает руководящее звено информацией для принятия решений, минимизирует затраты труда на подготовку и принятие решений, а также содержит набор наиболее типовых задач для определенной сферы национальной экономики.

Системная архитектура АЭИС представляет собой совокупность обеспечивающих подсистем.

Обеспечивающие подсистемы являются общими для всех АЭИС независимо от конкретных функциональных подсистем, в которых применяются те или иные виды обеспечения, что позволяет реализовать принципы совместимости систем в процессе их функционирования. В состав обеспечивающих подсистем входят подсистемы: информационного, программного, технического, организационного, правового, лингвистического и технологического обеспечения.

Информационное обеспечение (ИО) АЭИС. Назначение подсистемы информационного обеспечения состоит в своевременном формировании и выдаче достоверной информации для принятия управленческих решений. Представляет собой совокупность единой системы классификации и кодирования технико-экономической информации, унифицированной системы документации и информационной базы. [20]

В рамках информационного обеспечения выделяют два комплекса: компоненты внемашинного (внешнего) информационного обеспечения (классификаторы технико-экономической информации и документы) и внутримашинного информационного обеспечения (макеты/экранные формы для ввода первичных данных в ЭВМ или вывода результатной информации, структура информационной базы: входных, выходных файлов, базы данных).

Центральный компонент информационного обеспечения - база данных, через которую осуществляется обмен данными различных задач. База данных обеспечивает интегрированное использование различных информационных объектов в функциональных подсистемах.

Информационная база может быть создана как совокупность отдельных файлов, каждый из которых отражает некоторое множество однородных документов или как база данных. В последнем случае файлы будут зависимыми и структура одних файлов будет зависеть от структуры других, а

структура файлов базы данных не будет соответствовать структуре управленческих документов.

Внутримашинное (внутреннее) информационное обеспечение также содержит систему программ организации, накопления, ведения и доступа к данным.

В техническую документацию информационного обеспечения входит описание: информационного обеспечения, системы классификации и кодирования информации; входных сообщений; описание выходных сообщений; формы документов; структуры массивов информации; организации информационной базы и технологических процессов.

Организационное обеспечение (ОО). Подсистема организационного обеспечения - важнейшая подсистема АЭИС, от которой зависит успешная реализация целей и функции системы и представляет собой комплекс документов, регламентирующих процесс создания и функционирования системы. В составе организационного обеспечения выделяют следующие четыре группы компонентов: методические материалы, совокупность средств, необходимых для эффективного проектирования и функционирования АЭИС, техническая документация, персонал.

Правовое обеспечение АЭИС (Пр О). Подсистема предназначена для регламентации процесса создания и эксплуатации АЭИС, которая включает совокупность правовых (юридических) документов с констатацией регламентных отношений по формированию, хранению, обработке промежуточной и результатной информации системы.

Техническое обеспечение (ТО) АЭИС. Подсистема представляет собой комплекс технических средств (КТС) предназначенных для обработки данных в АЭИС, а также соответствующую документацию на эти средства и персонал. Цель создания ТО - выбор и оснащение АЭИС средствами сбора, регистрации,

хранения, накопления, обработки информации, создания условии для нормальной загрузки и надежности элементов системы при решении в установленном режиме всех необходимых функциональных задач.

КТС, как важнейшая составляющая ТО АЭИС, предназначена:

- для надежного и своевременного решения функциональных задач;
- представления результатов решения задач пользователям в необходимых разрезах и объеме;
- сопряжения и информационного взаимодействия этой АЭИС с внешними информационными системами;
- обеспечения функционирования и организации базы данных;
- передачи информации по техническим каналам связи;
- решения задач с различными режимами обработки данных (в пакетном, диалоговом, реальном масштабе времени и т. д.).

Для достижения заданной эффективности КТС АЭИС формируется во взаимоувязанный набор устройств обработки данных.

В состав комплекса технических средств входят:

- электронные вычислительные машины, осуществляющие обработку экономической информации, которые могут объединяться в вычислительные сети;
- средства съема, регистрации, сбора информации, подготовки данных на машинных носителях, средства передачи данных по каналам связи, накопления, хранения данных и выдачи результатной информации; организационная техника и вспомогательное оборудование;

- эксплуатационные материалы.

Выбор технических средств, организация их эксплуатации, технологическое оснащение оформляются посредством документации, которую условно можно подразделить на следующие три группы:

- общесистемную, включающую государственные и отраслевые стандарты по техническому обеспечению;

- специализированную, содержащую комплекс методик по всем этапам разработки технического обеспечения;

- нормативно-справочную, используемую при выполнении расчетов по техническому обеспечению.

Персонал включает специалистов по сопровождению, обслуживанию и эксплуатации комплекса технических средств.

Математическое обеспечение (МО) АЭИС. Подсистема включает в себя совокупность математических методов, моделей и алгоритмов решения задач управления и обработки информации, комплекса методов и средств, позволяющих строить экономико-математические модели задач управления и персонал. В функционирующей системе математическое обеспечение реализуется в составе программного обеспечения.

Программное обеспечение (ПО) АЭИС. В данном случае подразумевается совокупность программ (в том числе. программных средств) с программной документацией на них, необходимых для реализации целей и задач АЭИС, а также нормального функционирования комплекса технических средств.

Состав и структура программного обеспечения АЭИС определяются исходя из состава решаемых в системе задач и выбранного комплекса

технических средств. В состав программного обеспечения АЭИС в общем случае входят: общее и специальное программное обеспечение, техническая документация, описания и инструкции по их применению; персонал, занимающийся его разработкой и сопровождением на весь период жизненного цикла АЭИС.

Лингвистическое обеспечение (ЛО) АЭИС. Целью лингвистического обеспечения является повышение эффективности разработки автоматизированной обработки информации путем облегчения общения человека с АЭИС.

Технологическое обеспечение (ТехО). Под этим подразумевается технологический процесс, обеспечивающий преобразование данных, начиная от их сбора, и кончая, получением результатных данных и ее передачей пользователям. При этом под технологическим процессом понимается совокупность операции по преобразованию информации, их технические, информационные и организационные взаимосвязи, осуществляемые в соответствии с заданными требованиями.

Эргономическое обеспечение (ЭрО) АЭИС. Подсистема включает совокупность методов и средств, которые используются на разных этапах разработки и функционирования АЭИС, создающих оптимальные условия для деятельности человека в данной системе, быстрейшего ее освоения, а также обеспечения высокоэффективной и безошибочной деятельности в ней человека. Вопросы эргономического обеспечения разрабатываются для особо важных и ответственных систем и проектных решений.

2. Понятие и виды виртуализации

Средства вычислительной техники, прежде всего персональные ЭВМ, интенсивно используются в различных областях экономики и охватывая практически все сферы человеческой деятельности.

Спектр реализуемых ими функций чрезвычайно широк и разнообразен: от выполнения элементарных вычислений до построения сложных систем обработки и управления. Как было сказано выше, к таким системам относятся АЭИС.

Характерными особенностями указанных систем с точки зрения технического обеспечения является следующее:

- в качестве центрального звена используются ЭВМ того или иного класса;
- АЭИС представляют собой результат интеграции и совместного применения комплекса различных технических средств;
- основное назначение связано с повышением эффективности и производительности человеко-машинного труда на базе автоматизации современных и новых информационных технологий.

Вычислительные и логические возможности, эффективность работы и другие показатели систем обработки данных в значительной степени определяются совершенством их технической базы – комплекса технических средств, включающего в себя ЭВМ и периферийное оборудование, средства сбора и подготовки данных, средства передачи данных и оргтехники.

Выбор технического обеспечения является сложной и многоплановой задачей, требующей знаний принципов организации и работы технических средств информационных технологий, их возможностей и характеристик, способов совместного применения, выбора структуры и состава с учетом проектирования процессов сбора и обработки информации, с привязкой к принятым решениям по математическому, информационному и организационному обеспечению. Способы использования компьютера принято называть организационными формами использования машин. На практике их применяется два вида:

- Вычислительные центры.

- Локальные автоматизированные рабочие места (АРМы) и вычислительные сети.

Вычислительные центры применяются на крупных предприятиях, банках, государственных органах. Это специфические предприятия по обработке информации. Они оснащаются большими и сверхбольшими ЭВМ, а в качестве вспомогательных используются мини-ЭВМ, микро-ЭВМ. В Вычислительных центрах (ВЦ) есть система управления (руководства), отделы постановки задач, программирования, обслуживания машин, а также производственные подразделения: группы приемки документов, переноса информации на носители, администрация банков данных, выпуска информации, размножения материалов и т.д.

Для АРМов специалистов характерно размещение компьютеров на рабочих местах, по отдельным участкам работ.

Современные условия высокой скорости развития ИКТ позволяют сократить расходы на средства вычислительной техники с одновременным повышением производительности, эффективности использования и гибкости имеющегося в наличии вычислительного оборудования посредством использования различных видов и методов виртуализации.

В широком смысле, понятие виртуализации представляет собой сокрытие настоящей реализации какого-либо процесса или объекта от истинного его представления для того, кто им пользуется. Продуктом виртуализации является нечто удобное для использования, на самом деле, имеющее более сложную или совсем иную структуру, отличную от той, которая воспринимается при работе с объектом. Иными словами, происходит отделение представления от реализации чего-либо. В компьютерных технологиях под термином «виртуализация» обычно понимается абстракция вычислительных ресурсов и предоставление пользователю системы, которая «инкапсулирует» (скрывает в себе) собственную реализацию.

Виртуализация - актуальная тема для развивающихся экономических информационных систем. Существуют следующие виды виртуализации: виртуализация серверов, виртуализация приложений, виртуализация представлений, виртуализация сети. [17]

Ниже рассмотрим вышеперечисленные виды виртуализации более подробно.

Виртуализация серверов

Виртуализация серверов – это процесс запуска специализированного программного обеспечения под операционной системой, называемой хостом (Host OS), дающего возможность создавать виртуальные машины (Virtual Machine), обладающие заданными характеристиками реальных компьютеров, и запускать на них независимо друг от друга различные гостевые ОС (Guest OS).

Иначе говоря, на такой виртуальный сервер может быть установлена ОС, на которую в свою очередь могут быть установлены приложения и службы (Guest OS) и некие приложения. Работать все это будет как на полноценном сервере, только он невидим: он существует виртуально, внутри ОС на физическом сервере (применительно к этой ОС используется термин хостовая ОС, Host OS). При этом внутри одного физического сервера могут одновременно работать два и более, а иногда даже десятки таких виртуальных серверов.

Такая виртуализация все больше и больше используется в промышленном применении, так как использование виртуализации позволяет более рационально распределять аппаратные ресурсы серверов. Действительно, ведь большинство серверов использует от силы 10% от своих ресурсов - процессорных мощностей, объемов памяти и т.д. [16] Виртуализация позволяет вместо нескольких практически незагруженных серверов использовать один сервер, который будет загружен чуть сильнее. Один сервер, пусть даже чуть более мощный, будет стоить дешевле, чем

несколько отдельных, а также один сервер будет потреблять намного меньше электроэнергии и занимать меньше места.

Еще одно очень важное преимущество - удобство администрирования. Использование виртуализации позволяет получать доступ к консолям виртуальных серверов непосредственно с рабочего места администратора, и необходимость в экскурсиях в серверную практически отпадает. Кроме этого, сильно упрощаются операции резервного копирования и аварийного восстановления серверов. Использование виртуализации позволяет создавать резервные копии дисков серверов «на лету», незаметно для пользователей, а восстановление сводится всего лишь к копированию нескольких файлов.

Существует несколько видов платформ виртуализации, в каждом из них осуществляется свой подход к понятию -виртуализация. Виды виртуализации платформ зависят от того, насколько полно осуществляется симуляция аппаратного обеспечения. До сих пор нет единого соглашения в терминах в сфере виртуализации, поэтому некоторые из приведенных ниже видов виртуализации могут отличаться от тех, что предоставляют другие источники. В приложении № 1 приводится пример различия терминов в определении видов виртуализации.

Аппаратная виртуализация реализуется за счёт так называемого гипервизора (Hypervisor) – специализированного программного обеспечения, которое само является в некотором роде операционной системой. В литературе также часто используется термин монитор или же менеджер виртуальных машин (Virtual Machine Monitor/Manager, сокращенно VMM). Слово «гипервизор» произошло от обозначения программного обеспечения, работающего «под супервизором» («supervisor» - старое название операционной системы). Это своего рода "программная прослойка" или "программный слой", поскольку именно гипервизор обеспечивает взаимодействие операционных систем и аппаратного обеспечения (в частности, процессора). Таким образом, гостевые системы используют не

ресурсы хост системы, а напрямую аппаратные ресурсы компьютера. Гипервизор управляет виртуальными машинами, распределяет ресурсы, обеспечивает их независимость и, в некоторых случаях, взаимодействие.

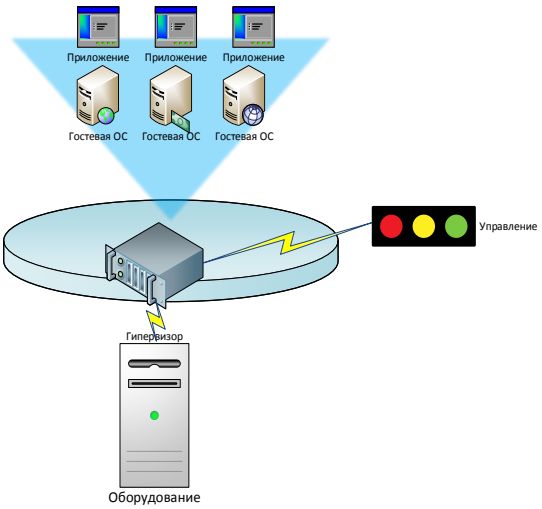


Рис. 2 Схема работы аппаратной виртуализации.

Однако не все действия совершает гипервизор, потому что гостевые системы по-прежнему должны устанавливаться в хост системе, через специализированное программное обеспечение, поддерживающее технологию аппаратной виртуализации.

В целях необходимости поддержки аппаратной виртуализации возникла необходимость изменить архитектуру процессоров за счет введения дополнительных инструкций для предоставления прямого доступа к ресурсам процессора из гостевых систем. Этот набор дополнительных инструкций носит название Virtual Machine Extensions (VMX). VMX предоставляет следующие инструкции: VMPTRLD, VMPTRST, VMCLEAR, VMREAD, VMREAD, VMWRITE, VMCALL, VMLAUNCH, VMRESUME, VMXON и VMXOFF.

Процессор с поддержкой виртуализации может работать в двух режимах root operation и non-root operation. В режиме root operation работает VMM.

Чтобы перевести процессор в режим виртуализации, платформа виртуализации должна вызвать инструкцию VMXON и передать управление

гипервизору, который запускает виртуальную гостевую систему инструкцией VMLAUNCH и VMRESUME (точки входа в виртуальную машину). Virtual Machine Monitor может выйти из режима виртуализации процессора, вызвав инструкцию VMXOFF.

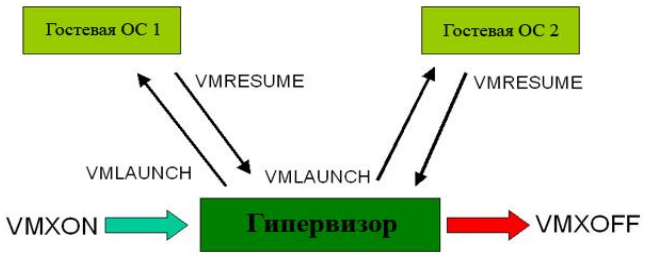


Рис. 3. Процедура запуска виртуальных машин

Таким образом, каждая из гостевых операционных систем запускается и работает независимо от других и является изолированной с точки зрения аппаратных ресурсов и безопасности.

На сегодняшний день существуют две технологии аппаратной виртуализации, представленные двумя крупнейшими производителями процессоров **Intel** и **Advanced Micro Devices** (**AMD**).

Технология **Intel Virtualization Technology** (**Intel VT**) требует поддержки не только со стороны процессора, но также чипсета и BIOS материнской платы. Принцип работы следующий: пользователь запускает программу виртуализации, которая в свою очередь активирует специальный режим работы процессора. Далее всю работу по корректному обслуживанию виртуальной машины берет на себя VMM.

AMD предложила своим пользователям собственную технологию **AMD Virtualization** (**AMD-V**), базирующуюся на другой фирменной технологии **Direct Connect**. Сама виртуализация построена таким образом, что VMM полагает все запущенные на компьютере ОС виртуальными:

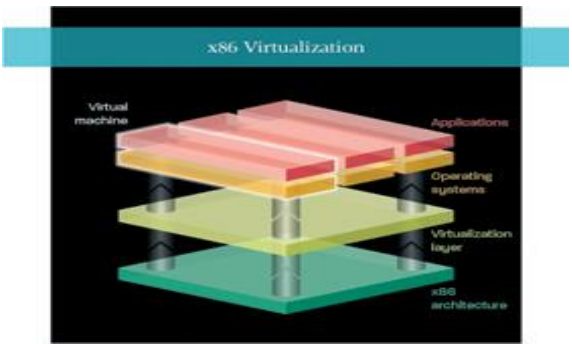


Рис. 4.Схема виртуализации AMD-V

При создании виртуальной машины процессор переходит в так называемый гостевой режим, после чего VMM, в отличии от технологии Intel, уже практически не принимает участие в работе системы.

Следует отметить, что, не смотря на всю внешнюю схожесть, эти технологии не совместимы между собой. Таким образом, использовать аппаратную виртуализацию **IVT** за счет программ поддерживающих исключительно технологию **AMD-V**, и наоборот, не возможно.

К преимуществам аппаратной виртуализации можно отнести возможно получение изолированных гостевых систем, управляемых гипервизором напрямую. Такой подход может обеспечить простоту реализации платформы виртуализации и увеличить надежность платформы с несколькими одновременно запущенными гостевыми системами, при этом нет потерь производительности на обслуживание хостовой системы. Такая модель позволит приблизить производительность гостевых систем к реальным и сократить затраты производительности на поддержание хостовой платформы. Таким образом обеспечивается высокая эффективность, а также отпадает потребность вносить изменения в операционную систему.

Единственный недостаток - необходимость использования специальных процессоров.

К наиболее известным продуктам аппаратной виртуализации относится качественный, функциональный, и при этом совершенно бесплатный XEN, в

след за которым поддержку данной технологии получили продукты и других компаний таких как KVM.

Программная виртуализация имитирует работу виртуального аппаратного обеспечения, то есть компьютера, состоящего из процессора, ОЗУ, жесткого диска, сетевой карты и т. д. Гостевые системы «считают», что виртуальное аппаратное обеспечение является реальным. Для того, чтобы такая система функционировала, работающая на хозяине программа виртуализации должна отслеживать код гостя и заменять определенные команды другими фрагментами кода. Эту задачу выполняет гипервизор (VMM). Такая программа-гипервизор также отвечает за события, связанные с хранением информации и управлением процессами.

Иначе говоря, не требуется никакого специализированного аппаратного обеспечения, как в случае с аппаратной виртуализацией. Пользователь просто устанавливает одну из программ виртуализации, создаёт в ней виртуальные машины и запускает на них гостевые ОС. При этом, разумеется, используются лишь ресурсы потребляемые хост-системой.

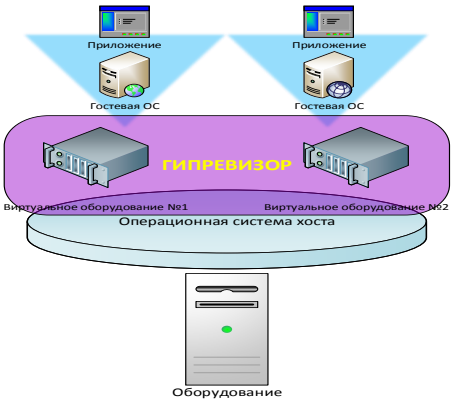


Рис. 5. Схема работы программной виртуализации.

К преимуществам программной виртуализации можно отнести функционирование практически любой гостевой операционной системы. При этом в операционную систему не требуется вносить никаких изменений.

К недостаткам относится сложность и трудоёмкость реализации такой платформы, присутствие потери производительности. Безопасность

виртуальных машин также находится под угрозой, поскольку получение контроля на хостовой операционной системой автоматически означает получение контроля над всеми гостевыми системами.

К наиболее известным продуктам программной виртуализации относится такие продукты, как: VMware, QEMU, Parallels, VirtualBox, Microsoft Virtual PC.

Паравиртуализация – виртуализация, при которой производится 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 гипервизора. Именно этот вариант является сегодня наиболее актуальным направлением развития серверных технологий виртуализации. То есть этот способ похож на полную виртуализацию, так как здесь тоже используется гипервизор, но код виртуализации интегрируется в саму операционную систему.

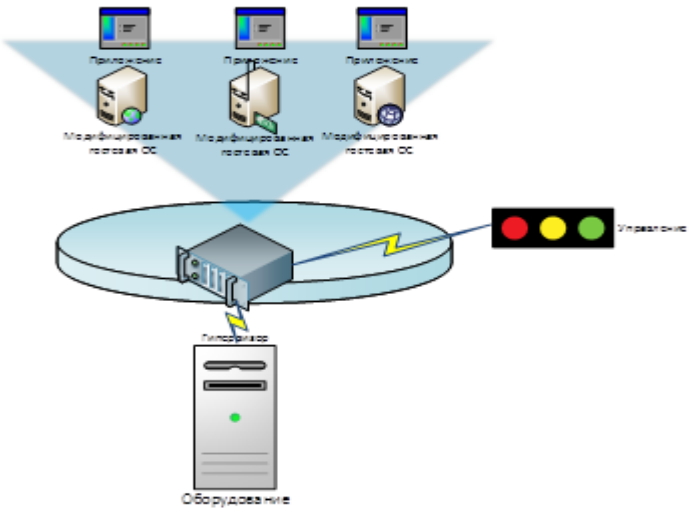


Рис 6 Схема работы паравиртуализации.

Преимущества данной технологии заключаются в отсутствии потребности в хостовой ОС – виртуальные машины, устанавливаются

фактически на "голое железо", а аппаратные ресурсы используются эффективно. То есть обеспечивается производительность, близкая к производительности невиртуализированной системы.

Недостатками являются сложности реализации подхода и необходимости создания специализированной ОС-гипервизора путём модификации гостевой операционной системы

К наиболее известным продуктам паравиртуализации относится: VMware ESX Server, Xen (и решениях других поставщиков на базе этой технологии), Microsoft Hyper-V.

Виртуализация на уровне операционной системы– это вид виртуализации, который подразумевает использование одного ядра хостовой ОС для создания независимых параллельно работающих операционных сред. Для гостевого ПО создается только собственное сетевое и аппаратное окружение. При использовании данного метода настоящие виртуальные машины не применяются. Вместо этого при таком подходе машины используют общее ядро и фрагменты файловой системы хозяина. Такая система виртуализации обеспечивает изоляцию между хозяином и гостями в целях исключения каких бы то ни было проблем с безопасностью.

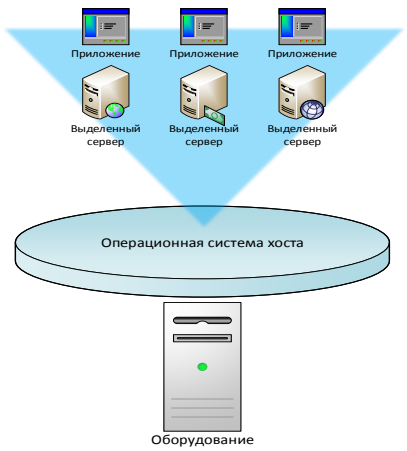


Рис. 7. Схема работы виртуализации на уровне ОС

К достоинствам виртуализация на уровне операционной системы можно отнести высокую эффективность использования аппаратных ресурсов, низкие накладные технические расходы (сбережение ресурсов: ОЗУ, дискового

пространства и т. д.), отличную управляемость, минимизацию расходов на приобретение лицензий.

К недостаткам относиться реализация только однородных вычислительных сред, то есть может применяться только тогда, когда хозяин и гости используют в точности одну и ту же операционную систему и совершенно одинаковую версию ядра. ОСдолжна быть модифицирована соответствующим образом.

К наиболее известным продуктам виртуализациии на уровне операционной системы относится: Virtuozzo (для Linux и Windows), OpenVZ (бесплатный вариант Virtuozzo) и Solaris Containers, Linux-VServer

Учитывая выше изложенное, повышенный интерес к компьютерным технологиям виртуализации в настоящее время не случаен. Вычислительная мощь нынешних процессоров быстро растет, и вопрос сейчас ставится в эффективности её использования.

Технологии виртуализации становятся одним из ключевых компонентов в самых новых и будущих процессорах Intel и AMD, и в операционных системах.

Использование виртуализации поможет повысить эффективность надёжность и отказоустойчивость экономических информационных систем. При этом необходимо учитывать специфичность АЭИС при выборе вышерассмотренных видов виртуализации серверов. Для этого следует детально рассмотреть различные характеристики видов виртуализации.

Во-первых, необходимо учитывать поддержку различных ОС в качестве гостевых систем, а также с обеспечением возможности работы приложений в виртуальных средах. При выборе продукта виртуализации нужно также иметь в виду широкий набор технических характеристик: уровень потери

производительности приложений в результате появления нового операционного слоя, необходимость дополнительных вычислительных ресурсов для работы механизма виртуализации, спектр поддерживаемой периферии.

Во-вторых, при виртуализации АЭИС необходимо обратить особое внимание на решение задач управления такими системами, как: преобразование физических сред в виртуальные и наоборот, восстановление системы в случае отказа, перенос виртуальных сред с одного компьютера на другой, развертывание и администрирование ПО, обеспечение безопасности и т. д.

В-третьих, важны стоимостные показатели используемой инфраструктуры виртуализации. При этом следует иметь в виду, что здесь в структуре расходов главной может быть не столько цена самих средств виртуализации, сколько возможность экономии на приобретении лицензий для базовых ОС или бизнес-приложений.

Учитывая, что на крупных хозяйственных субъектах используют различные ОС, рассмотрение виртуализации на уровне операционной системы не считается приемлемой, поскольку такая технология использует общее ядро, то ОС виртуальных машин работают на однородном ядре.

В отношении паравиртуализации нужно отметить ее неадекватность к современным массовым вычислениям. Паравиртуализация не дает значительного роста производительности работы гостевых ОС. Хотя по сравнению с программной виртуализацией или применением чистого hypervisor у паравиртуализации прирост продуктивности больший. Но не настолько, чтобы расходовать массу сил и времени на создание подобных

версий ОС. Паравиртуализация неуместна согласно закону Мура [1] и усовершенствований архитектуры современных процессоров, так уже сейчас более 80% доступных серверных процессоров Intel и AMD имеют встроенные средства виртуализации – VT (Virtualization Technology) и SVM (Secure Virtual Machine) соответственно. Такие решения разработаны для оптимизации производительности hypervisor-виртуализации и делают паравиртуализацию практически бесперспективной.

Теперь ставится вопрос эффективности использования аппаратной или программной виртуализации в АЭИС. Компания VMware провела исследование собственной программы виртуализации в сравнении с аппаратными технологиями виртуализации компании Intel на процессоре 3.8 GHz Intel Pentium 4 672 с отключенной технологией Hyper-Threading в 2006 году[2].

Один из экспериментов проводился с помощью систем тестов SPECint2000 и SPECjbb2005, являющихся стандартом де-факто для оценки производительности компьютерных систем. В качестве гостевой системы использовалась ОС Red Hat Enterprise Linux 3, управляемая программным и аппаратным гипервизором. Ожидалось, что аппаратная виртуализация даст коэффициент производительности около ста процентов в отношении программного (нативного) запуска операционной системы. Однако результаты оказались весьма неожиданными: в то время как программный гипервизор без использования аппаратных техник виртуализации давал 4% потерь производительности в отношении нативного запуска, аппаратный гипервизор, в целом, терял 5% производительности. Результаты этого теста приведены на рисунке ниже:

[1] Производительность процессоров постоянно увеличивается, емкость и скорость работы основной памяти также возрастают, причем с одновременным снижением стоимости
[2] Документ «A Comparison of Software and Hardware Techniques for x86 Virtualization»

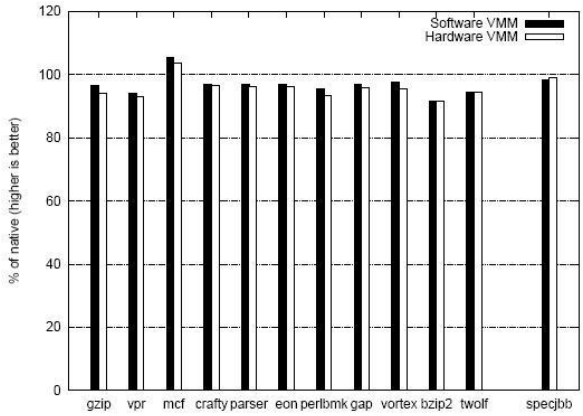


Рис. 8. Сравнение производительности программного и аппаратного гипервизоров

Тем не менее, не стоит воспринимать результаты производительности, как единственно верные. Объективная оценка производительности различных аппаратных и программных платформ для виртуализации является нетривиальной задачей, упомянутая рабочая группа в составе SPEC работает над созданием набора стандартных методов для оценки таких систем. На сегодня можно отметить, что средства виртуализации от AMD являются технически более совершенными, нежели реализованные Intel.

Однако, как было отмечено, программная виртуализация может подвергнуть угрозе безопасности виртуальных серверов, что следует учитывать при использовании в АЭИС, но этот минус не даёт веских причин отказаться от такой технологии.

Поддержка технологий аппаратной виртуализации в процессорах открывает широкие перспективы по использованию виртуальных машин в качестве надежных, защищенных и гибких инструментов для повышения эффективности виртуальных инфраструктур. Использование аппаратной виртуализации в перспективе должно уменьшить потери производительности при запуске нескольких виртуальных машин на одном физическом сервере. Безусловно, аппаратная виртуализация повысит защищенность виртуальных систем в корпоративных средах.

Таким образом, необходимо отметить в отношении аппаратной виртуализации следующее:

1. Потери производительности на виртуализацию незначительны.
2. Большинство технологий работают на операционных системах семейства Linux.
3. Большинство технологий доступны бесплатно и являются открытыми проектами.

Все это делает данные технологии легкодоступными для использования в различных отраслях экономики и снижает затраты на использование АЭИС.

Виртуализация рабочих мест

Наряду с виртуализацией серверов существует еще одно решение по виртуализации АЭИС – виртуализация рабочих мест. Виртуализация рабочих мест или десктопов (Virtual Desktop Infrastructure – VDI) – это технология создания рабочих столов в виртуальной среде. Данное решение даёт возможность пользователю запустить его персональную рабочую среду с сервера виртуализации (терминального сервера). Подключение к серверу осуществляется посредством либо тонкого клиента, установленного на его рабочем месте, вместо традиционного ПК, либо мобильно с ПК (ноутбука) с помощью специального агента.

Иными словами, с помощью технологии виртуализации рабочих мест сотрудник, имея любое устройство с доступом в сеть – смартфон, планшетный компьютер, тонкий клиент, – может получить доступ к персональному рабочему столу и корпоративным информационным ресурсам. Внедрение VDI позволяет упростить создание и администрирование рабочих мест пользователей, обеспечить гибкость своей IT-инфраструктуры.

Надо отметить, что VDI – это необходимость, которая позволяет существенно сократить расходы на эксплуатацию IT-сферы в рамках

отдельного предприятия, организации. Косвенно, это подтверждается и тем фактом, что все больше компаний разных сфер экономики, банков, заводов и т.д., так или иначе, используют виртуализацию в своей работе.

В настоящее время сложилось два основных вида построения вычислительных систем по типу архитектуры: централизованный и распределенный. Распределенная архитектура дополнительно включает двух- и трехзвенную клиент-серверную архитектуру. Принципиальным различием между ними является то, что при распределенной архитектуре большая часть вычислений проходит на «клиенте», а при централизованной все вычисления выполняются на центральном сервере. Система, основанная на терминалах, представляет собой центральную вычислительную площадку, к которой подсоединяются терминальные клиенты. Причем клиенты могут быть как стационарными, так и мобильными, а подключаться не только через LAN, но и через WAN. На центральной вычислительной площадке находится терминальный сервер, он же, как правило, и является сервером приложений, который может быть связан с сервером баз данных. На площадке также может находиться резервный терминальный сервер, обеспечивающий повышенную отказоустойчивость и высокую готовность системы в целом. При централизованной архитектуре особо актуально применение технологии «тонкий клиент».

В связи с усложнением инфраструктуры предприятий и организаций, стремительно растет количество необходимых прикладных программ и приложений, постоянно возрастает уровень требований к мощностям вычислительных ресурсов.

Поэтому целями и задачами использования терминальной технологии на базе использования тонких клиентов является:

- Снижение временных расходов на администрирование;
- Повышение безопасности — снижение риска инсайдерских взломов;

- Снижение затрат на программное и аппаратное обеспечения;
- Снижение расхода электроэнергии;
- Увеличение отказоустойчивости оборудования.

При этом тонкие клиенты должны позволять не только решать технические проблемы, но и побороть сложившиеся стереотипы.

Ниже приводятся специфические термины, используемые относительно виртуализации рабочих мест.

Терминальный комплекс – многомашинная ассоциация, предназначенная для *организации массового доступа* удаленных и локальных пользователей к ресурсам некоторой вычислительной системы.

Терминальный режим работы — организация сетевой работы информационной системы (ИС) посредством размещения всех пользовательских приложений и данных на центральном сервере (серверах), доступ к которым осуществляется с машин-терминалов, изготовленных в упрощённом исполнении и, как следствие, более дешёвых, занимающих минимум места, бесшумных и практически не требующих обслуживания. При работе пользователя с сервером терминалов приложение выполняется на сервере, а по сети передаются только события клавиатуры, мыши и отображения на экране.

Тонкий клиент (англ. *thin client*) в компьютерных технологиях — компьютер или программа-клиент в сетях с клиент-серверной или терминальной архитектурой, который переносит все или большую часть задач по обработке информации на сервер.

Иначе говоря, под термином «тонкий клиент» подразумевается достаточно широкий с точки зрения системной архитектуры ряд устройств и программ, которые объединяются общим свойством: возможность работы в терминальном режиме. Таким образом, для работы тонкого клиента

необходим терминальный сервер. Этим тонкий клиент отличается от толстого клиента, который, напротив, производит обработку информации независимо от сервера, используя последний в основном лишь для хранения данных.

Аппаратный тонкий клиент (например, Windows- и Linux-терминалы) — специализированное устройство, принципиально отличное от ПК. Он не имеет жёсткого диска, использует специализированную локальную ОС (одна из задач которой организовать сессию с терминальным сервером для работы пользователя), не имеет в своём составе подвижных деталей, выполняется в специализированных корпусах с полностью пассивным охлаждением. Для расширения функциональности тонкого клиента прибегают к его «утолщению», например, добавляют возможности автономной работы, сохраняя главное отличие — работу в сессии с терминальным сервером. Когда в клиенте появляются подвижные детали (жёсткие диски), появляются возможности автономной работы, он перестаёт быть тонким клиентом в чистом виде, а становится универсальным клиентом.

Тонкий клиент в большинстве случаев обладает минимальной аппаратной конфигурацией, вместо жёсткого диска для загрузки локальной специализированной ОС используется DOM (DiskOnModule) (модуль с разъёмом IDE, флэш-памятью и микросхемой, реализующей логику обычного жёсткого диска — в BIOS определяется как обычный жёсткий диск, только размер его обычно в 2-3 раза меньше). В некоторых конфигурациях системы тонкий клиент загружает операционную систему по сети с сервера, используя протоколы PXE, BOOTP, DHCP, TFTP и Remote Installation Services (RIS).

Иными словами, тонкий клиент представляет собой системный блок, у которого обычно нет жесткого диска, и присутствует только минимальный набор железа, нужный для запуска операционной системы тонного клиента (далее просто тонкого клиента). К системному блоку подключены питание,

мышь, клавиатура, монитор, сетевой кабель. Кроме стандартного набора к тонкому клиенту могут быть подключены другие устройства, при условии, что он сможет их распознать и передать терминальному серверу.

Терминальный сервер, *сервер терминалов*, *сервер виртуализации* (англ. *terminal server*) — сервер, предоставляющий клиентам вычислительные ресурсы (процессорное время, память, дисковое пространство) для решения задач. Технически терминальный сервер представляет собой очень мощный компьютер (либо кластер), соединенный по сети с терминальными клиентами - которые, как правило, представляют собой маломощные или устаревшие рабочие станции, либо специализированные решения для доступа к терминальному серверу. Терминальный сервер служит для удалённого обслуживания пользователя с предоставлением рабочего стола.

С технической точки зрения построение сети выглядит, как показано на рисунке 9.

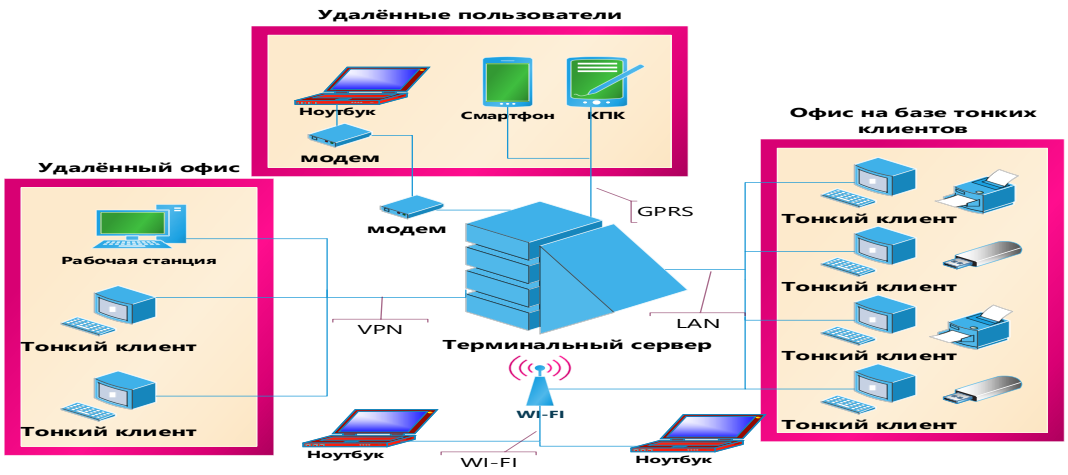


Рис. 9. Варианты решений на основе терминальной сети.

Далее загружается образ ОС тонкого клиента, и управление передаётся ей. Затем ОС тонкого клиента при загрузке ещё раз получает настройки сети и IP адрес TFTP-сервера и пытается загрузить конфигурационный файл. Если такой файл есть, то он загружается и применяется, если нет - применяются настройки по умолчанию.

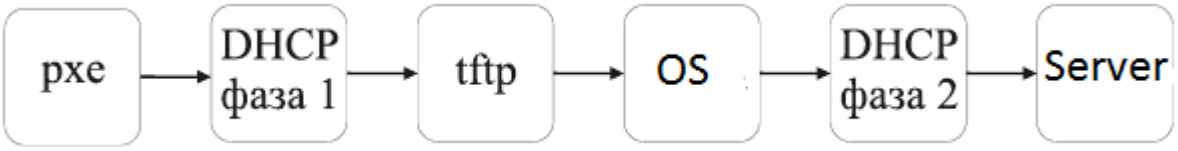


Рис. 10. Схема работы тонкого клиента.

Сам же, терминальный режим подразумевает, что на рабочем месте пользователя есть только «фреймы» т.е. изображение результата обработки информации. Все операции по ее обработке и хранению осуществляют сервера приложений. Доступ клиентов к приложениям осуществляется через терминальный сервер с использованием одного из известных протоколов терминального доступа.

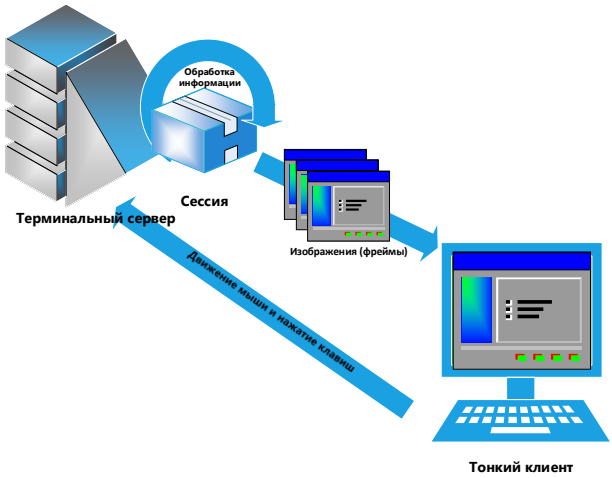


Рис. 11. Принцип работы терминальной технологии.

Ключевым элементом для организации рабочего места пользователя есть тонкий клиент. Иными словами, тонкий клиент после установления связи с терминальным сервером пересылает на последний вводимые данные (нажатия клавиш, перемещения мыши) и, возможно, предоставляет доступ к локальным ресурсам. Терминальный сервер предоставляет среду для работы (терминальная сессия), в которой исполняются приложения пользователя. Результат работы сервера передается на клиента, как правило, это изображение для монитора и звук (при его наличии).

Преимуществами использования терминальной технологии на базе тонких клиентов вместо обычных персональных компьютеров следующие:

- Снижение начальных затрат на приобретение, вследствие минимальных требований к конфигурации.
- Унификация — все клиенты имеют одинаковый набор программного обеспечения.
- Взаимозаменяемость оборудования.
- Простота реализации задач — нет необходимости настраивать каждый компьютер по отдельности, так как осуществляется централизованное управление информационным процессом. Все настройки для управления тонкими клиентами системный администратор выполняет централизованно на сервере.
- Экономия времени системного администратора, обслуживающего абсолютно одинаковые компьютеры, вероятность поломок которых сведена к минимуму, а все программы установлены на сервере.
- Масштабируемость — созданный единожды образ системы для работы всей группы пользователей позволяет поддерживать легко масштабируемую сеть. Можно установить столько ПК, сколько требуется, при этом добавление новых рабочих мест требует минимальных усилий.
- Безопасность и отказоустойчивость. Терминал, загружаясь, получает операционную систему «от производителя», настройка которой осуществляется только отделом информационной поддержки. Все модификации операционной системы и прикладного ПО никак не влияют ни на других пользователей, ни на образ, хранящийся на сервере. Вся пользовательская информация хранится на сервере на RAID-массиве и регулярно резервируется, что увеличивает отказоустойчивость.
- Защита от утечек информации — нет локальных носителей — нет возможности делать копии документов на съемные носители информации.
- Сверхнизкое энергопотребление. «Холодный» процессор, отсутствие вентиляторов и прочих моторчиков, жестких дисков и приводов, флэш-

память с мизерным потреблением – благодаря всему этому тонкий клиент потребляет чрезвычайно мало электроэнергии по сравнению с обычным ПК.

- Терминалы практически не подвержены моральному старению, и срок их службы в 2—3 раза больше, чем у персональных компьютеров.

Преимущества использования технологии «Тонких Клиентов» по сравнению со стандартными персональными ПК приведены в приложении №1.

К недостаткам можно отнести следующее:

- Концентрация всей функциональности в рамках одного (нескольких) серверов — выход из строя любого элемента между приложением и клиентами (сервер, коммутаторы, СКС) приводит к простою многих пользователей.
- Усиливаются негативные последствия ошибок конфигурации и работы ПО (последствия ошибок сказываются не на отдельных пользователях, а на всех пользователях сервера сразу же).
- Проблемы с лицензированием (некоторое ПО не предусматривает ситуации работы нескольких пользователей на одном компьютере или требует использования более дорогих версий).
- Неприменимы ресурсоемкие приложения для работы с графикой и трехмерным моделированием, такие как Photoshop, AutoCAD, 3D Studio Max.

Исходя из вышеизложенного терминальные комплексы являются полезной технологией позволяющей повысить гибкость ИКТ - инфраструктуры организаций, предприятий, фирм, уменьшить затраты на программное и аппаратное обеспечение. Недостатки присущие данному виду аппаратного и программного обеспечения, такие как сниженная эргономичность тонких клиентов может быть компенсирована правильным распределением данной техники среди сотрудников организации, продуманным применением ПО на серверных и клиентских частях терминальных комплексов.

Терминальный доступ обеспечит эффективность, если будет применяться по назначению. Снижение расходов (в том числе экономия на лицензиях), повышение уровня безопасности, уменьшение трудозатрат на обслуживание техники – таковы основные предпосылки применения этой технологии в АЭИС.

Виртуализация сети

Исторически телекоммуникационные технологии, и локальные сети развивались своими независимыми путями. Поэтому одна и та же проблема создания виртуальных соединений (виртуальных сетей) была фактически решена принципиально разными способами. Поэтому (с некоторой долей условности), можно выделить два пути:

Локальный. Строится на базе коммутируемого Ethernet с использованием виртуальных сетей (VLAN). Разделение происходит на уровне коммутатора, который имеет возможность выделять на канальном уровне одного или нескольких пользователей в группу по некоторым признакам (порту или MAC-адресу). То есть если есть несколько свичей в локальной сети с поддержкой стандарта 802.1q можно построить между ними VLAN. Применяется только в LAN.

Телекоммуникационный. Предполагает создание виртуальных каналов (туннелей) "поверх" транспортного протокола (обычно IP или Ethernet). Узел-клиент, используя свои учетные данные, устанавливает соединение "точка-точка" с сервером доступа, и уже через этот вновь образованный канал осуществляет передачу/прием данных. При этом как процедура авторизации, так и информационный обмен может быть зашифрован весь, либо частично (только заголовок и пароль авторизации). Применяется в WAN для связи между публичными IP, можно применять в LAN.

Виртуальные локальные сети. Виртуальной сетью VLAN (Virtual Local-Area Network) называется группа узлов сети, трафик которой, в том

числе и широковещательный, на канальном уровне полностью изолирован от других узлов сети. Таким образом, технология виртуальных сетей позволяет преодолеть недостатки отсутствия барьеров на пути широковещательного трафика, заложенного в алгоритме работы сетевого моста, который реализован в коммутаторе.

Иначе говоря, кроме своего основного назначения - повышения пропускной способности связей в сети - коммутатор позволяет локализовать потоки информации в сети, а также контролировать эти потоки и управлять ими, используя пользовательские фильтры. Однако, пользовательский фильтр может запретить передачи кадров только по конкретным адресам, а широковещательный трафик он передает всем сегментам сети. А применение технологии виртуализации локальной сети делает невозможной передачу кадров между разными виртуальными сегментами на основании адреса канального уровня независимо от типа адреса - уникального, группового или широковещательного.

В то же время внутри виртуальной сети кадры передаются по технологии коммутации, то есть только на тот порт, который связан с адресом назначения кадра. По аналогии с доменом коллизий, который образуется повторителями сетей Ethernet, можно скачать, что виртуальная сеть образует домен широковещательного трафика (broadcast domain).

Назначение технологии виртуальных сетей состоит в облегчении процесса создания независимых сетей, которые затем должны связываться с помощью протоколов сетевого уровня. При использовании технологии виртуальных сетей в коммутаторах одновременно решаются две задачи:

- повышение производительности в каждой из виртуальных сетей, так как коммутатор передает кадры в такой сети только узлу назначения;
- изоляция сетей друг от друга для управления правами доступа пользователей и создания защитных барьеров на пути широковещательных штормов.

Для связи виртуальных сетей в интерсеть требуется привлечение сетевого уровня. Он может быть реализован в отдельном маршрутизаторе, а может работать и в составе программного обеспечения коммутатора.

Как уже говорилось, виртуальная сеть представляет собой коммутируемую сеть, в которой выполнено логическое сегментирование по исполняемым функциям, используемым приложениям или по принадлежности пользователей к определенному отделу, вне зависимости от физического расположения их компьютеров. Каждый порт коммутатора может быть включен в виртуальную сеть. Все порты, включенные в одну виртуальную сеть, принимают широковещательные сообщения в ее пределах, в то время как порты, в нее не включенные, этих сообщений не принимают.

Существует несколько способов построения виртуальных сетей:

Группировка портов - это тип виртуальных локальных сетей (ВЛС) определяющий членство каждой ВЛС на основе номера подключенного порта.

VLAN на основе группировки MAC-адресов - это второй способ, который используется для образования виртуальных сетей. При существовании в сети большого количества узлов этот способ требует выполнения большого количества ручных операций от администратора. Однако, он оказывается более гибким при построении виртуальных сетей на основе нескольких коммутаторов, чем способ группирования портов.

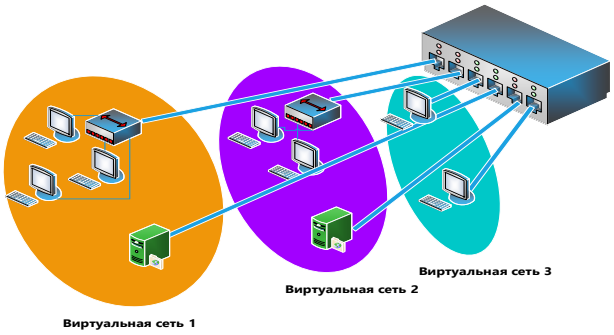


Рис. 12. Пример порт ориентированной ВЛС

При использовании меток в дополнительном поле кадра (*частные протоколы и спецификации IEEE 802.1 Q/p*) действует второй уровень сетевой модели. В каждый кадр вставляется тег ID идентифицирующий их членство в определенной VLAN. Эту технологию используют что бы создать виртуальные сети (VLAN) охватывающие множество коммутаторов.

Новый стандарт IEEE 802.1Q определяет изменения в структуре кадра Ethernet, позволяющие передавать информацию о VLAN по сети. Стандарт IEEE 802.1p специфицирует метод указания приоритета кадра, основанный на использовании новых полей, определенных в стандарте IEEE 802.1Q. К кадру Ethernet добавлены два байта. Эти 16 бит содержат информацию по принадлежности кадра Ethernet к VLAN и о его приоритете. Говоря точнее, тремя битами кодируется до восьми уровней приоритета, 12 бит позволяют различать трафик до 4096 VLAN, а один бит зарезервирован для обозначения кадров сетей других типов (Token Ring, FDDI), передаваемых по магистрали Ethernet. Надо сказать, что добавление двух байтов к максимальному размеру кадра Ethernet ведет к возникновению проблем в работе многих коммутаторов, обрабатывающих кадры Ethernet аппаратно. Чтобы избежать их, группы по стандартизации предложили сократить на два байта максимальный размер полезной нагрузки в кадре.

При *спецификации LANE для ATM-комутаторов* используются поля для маркировки принадлежности кадра виртуальной сети, которые принадлежат не кадрам канальных протоколов, а ячейкам технологии ATM[3] или пакетам сетевого уровня.

Спецификация LANE вводит такое понятие как эмулируемая локальная сеть - ELAN. Это понятие имеет много общего с понятием виртуальной сети:

- ELAN строится в сети, состоящей из коммутаторов (коммутаторов ATM);

[3] ATM (англ. Asynchronous Transfer Mode — асинхронный способ передачи данных) — сетевая высокопроизводительная технология коммутации и мультиплексирования, основанная на передаче данных в виде ячеек (cell) фиксированного размера (53 байта), из которых 5 байтов используется под заголовок.

- связь между узлами одной и той же ELAN осуществляется на основе MAC-адресов без привлечения сетевого протокола;
- трафик, генерируемый каким-либо узлом определенной ELAN, даже широковещательный, не выходит за пределы данной ELAN.

Кадры различных ELAN не смешиваются друг с другом внутри сети коммутаторов ATM, так как они передаются по различным виртуальным соединениям и номер виртуального соединения VPI/VCI является тем же ярлыком, который помечает кадр определенной VLAN в стандарте 802.1Q и аналогичных фирменных решениях.

Если VLAN строятся в смешанной сети, где имеются не только коммутаторы ATM, то "чистые" коммутаторы локальных сетей, не имеющие ATM-интерфейсов, должны использовать для создания виртуальной сети один из выше перечисленных методов, а пограничные коммутаторы, имеющие наряду с традиционными еще и ATM-интерфейсы, должны отображать номера VLAN на номера ELAN при передаче кадров через сеть ATM.

При использовании *сетевого протокола* коммутаторы должны для образования виртуальной сети понимать какой-либо сетевой протокол. Такие коммутаторы называют коммутаторами 3-го уровня, так как они совмещают функции коммутации и маршрутизации. Каждая виртуальная сеть получает определенный сетевой адрес - как правило, IP или IPX.

Принадлежность конечного узла к той или иной виртуальной сети в этом случае задается традиционным способом - с помощью задания сетевого адреса. Порты коммутатора также получают сетевые адреса, причем могут поддерживаться нестандартные для классических маршрутизаторов ситуации, когда один порт может иметь несколько сетевых адресов, если через него проходит трафик нескольких виртуальных сетей, либо несколько портов

имеют один и тот же адрес сети, если они обслуживают одну и ту же виртуальную сеть.

При передаче кадров в пределах одной и той же виртуальной сети коммутаторы 3-го уровня работают как классические коммутаторы 2-го уровня, а при необходимости передачи кадра из одной виртуальной сети в другую - как маршрутизаторы. Решение о маршрутизации обычно принимается традиционным способом - его делает конечный узел, когда видит на основании сетевых адресов источника и назначения, что кадр нужно отослать в другую сеть.

К преимуществам VLAN можно отнести главную функцию - создание виртуальных рабочих групп, основанных на общих функциях пользователей и общих ресурсах, в доступе к которым они нуждаются. При помощи реализации VLAN пользователи каждого департамента могут быть логически описаны и сгруппированы в различные рабочие группы с различными доступными ресурсами сети.

При использовании VLAN технологии большая сеть с большим широковещательным трафиком сегментируется на множество широковещательных доменов с несколькими рабочими станциями на один широковещательный домен. Следовательно частота (плотность) широковещания будет уменьшена. *Производительность каждой подсети возрастает*, потому что все сетевые устройства сети меньше отвлекается от передачи реальных данных при приеме широковещательного трафика

При использовании технологии VLAN, пользователи сети одной рабочей группы или отдела меньше ограничены их физическим местонахождением. Эта свобода зависит от возможностей применяемых Ethernet коммутаторов. В случае применения VLAN, пользователи сети одной рабочей группы или отдела могут находится на разных этажах и даже в разных

зданиях и при этом относиться к одной виртуальной сети. Тем самым обеспечивается разрушение традиционных концепций границ сети.

Многие управляемые коммутаторы позволяют одному коммутируемому порту иметь членство в нескольких VLAN. Благодаря этой возможности сервера могут предоставлять доступ рабочим станциям во всех виртуальных сетях. С другой стороны, доступ к серверам одного отдела, подключенных к портам с членством в одной VLAN возможен только в пределах соответствующей VLAN. Тем самым решается проблема безопасности и разделение доступа к сетевым ресурсам.

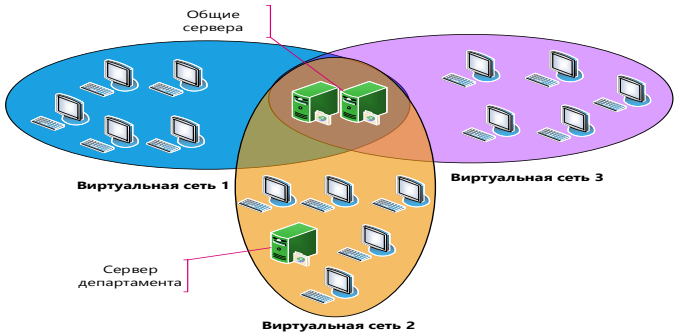


Рис. 13. Разделение доступа к сетевым ресурсам с помощью VLAN

При перемещении рабочих мест персонала из различных отделов в пределах организации, или изменения физического местоположения конкретного подразделения не требуется реконструкция соединений на существующих коммутаторах Ethernet. Так как идентификаторы ID членства VLAN будут перенесены вместе с рабочими станциями сети (применение VLAN (IEEE 802.1Q)), то стоимость перемещения включает только физическое перемещение рабочих мест персонала. Что обеспечивает *уменьшение затрат при перемещения персонала.*

К недостаткам можно отнести следующее:

- Конфигурирование VLAN в сложных сетях требует применения специализированных протоколов (GVRP) или существенного объёма ручной работы.
- При использовании протокола ISL требуется абонентское оборудование, понимающее этот протокол (поддерживается малым количеством пользователей).
- Использование IEEE 802.1Q требует использования коммутаторов, поддерживающих (как минимум) стандарт 802.3ab, стандартное оборудование 802.3u может уничтожать часть фреймов как нарушающие стандарт.
- В случае статической конфигурации оконечное оборудование теряет функциональность plug-n-play (так как порты коммутатора становятся не взаимозаменяемыми).

Исходя из вышеизложенного, нужно отметить следующее:

VLAN, базирующиеся на номере порта позволяют определить конкретный порт в VLAN. Порты могут быть определены индивидуально, по группам, по целым рядам и даже в разных коммутаторах через транспортный протокол. Это наиболее простой и часто используемый метод определения VLAN, когда рабочие станции используют протокол динамической настройки TCP/IP (DHCP).

VLAN, базирующиеся на MAC адресах позволяет пользователям находиться в той же VLAN, даже если пользователь перемещается с одного места на другое. Этот метод требует, чтобы администратор определил MAC адрес каждой рабочей станции и затем внес эту информацию в коммутатор. Этот метод может вызвать большие трудности при поиске неисправностей, если пользователь изменил MAC адрес. Любые изменения в конфигурации должны быть согласованы с сетевым администратором, что может вызывать административные задержки.

Виртуальные сети, базирующиеся на сетевых адресах, позволяют пользователям находиться в той же VLAN, даже когда пользователь перемещается с одного места на другое. Этот метод перемещает VLAN, связывая ее с сетевым адресом Уровня 3 рабочей станции для каждого коммутатора, к которому пользователь подключен. Этот метод может быть очень полезным в ситуации, когда важна безопасность и когда доступ контролируется списками доступа в маршрутизаторах. Поэтому пользователь "безопасной" VLAN может переехать в другое здание, но остаться подключенным к тем же устройствам потому, что у него остался тот же сетевой адрес. Сеть, построенная на сетевых адресах, может потребовать комплексного подхода при поиске неисправностей.

Виртуальная частная сеть VPN (virtual private network) – это обобщённое название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети. Слово «частный» в данном контексте означает, что передача данных между удаленными пользователями корпоративной сети компании осуществляется в зашифрованном виде, что позволяет говорить о создании безопасного канала связи — «туннеля». Иными словами, VPN - это объединение локальных сетей или отдельных машин, подключенных к сети общего пользования в единую сеть, обеспечивающую секретность и целостность передаваемой по ней информации. VPN использует в качестве среды для передачи данных в существующую коммуникационную инфраструктуру, например сеть Интернет. Виртуальные частные сети применяются для создания безопасных и надежных каналов, связывающих локальные сети и обеспечивающих доступ к ним пользователей, постоянно меняющих свое географическое местоположение.

Основными задачами, решаемых VPN является:

- обеспечение высокой безопасности передачи данных;

- создание единого информационного пространства организации, имеющей территориально-распределенные офисы, филиалы, склады и т.п.;
- организация внутрикорпоративной телефонной связи с единым номерным планом и возможностью набора короткого номера;
- получение удаленного доступа к общим информационным ресурсам и использование единой системы документооборота;
- получение доступа к распределенным информационным ресурсам объединенной сети;
- организация систем информационных киосков и банкоматов;
- обеспечение мобильности сотрудников.

Виртуальные частные сети подразделяются на два типа: пользовательские VPN и узловые VPN.

Пользовательские VPN – это сети, построенные между отдельной пользовательской системой и узлом или сетью организации. Пользователь подключается к интернету через телефонное подключение к локальному поставщику услуг, через канал DSL или кабельный модем и инициирует VPN-соединение с узлом организации через интернет.

Рис. 14. Конфигурация пользовательской VPN

К преимуществу пользовательских VPN можно отнести возможность удалённых сотрудников осуществлять доступ к внутренним системам в любое время без аренды дорогостоящих выделенных каналов, что способствует финансовой экономии.

К недостаткам пользовательских VPN можно отнести увеличение степени риска, связанного с безопасностью, и проблемами реализации такие как: сложности управления пользователями в домене и негативное влияние трансляции сетевых адресов (NAT) на организацию пользовательской VPN соответственно.

Узловые виртуальные частные сети используются для подключения к удаленным узлам без применения дорогостоящих выделенных каналов или для соединения двух различных организаций, между которыми необходима связь для осуществления информационного обмена. VPN соединяет один межсетевой экран или пограничный маршрутизатор с другим аналогичным устройством.

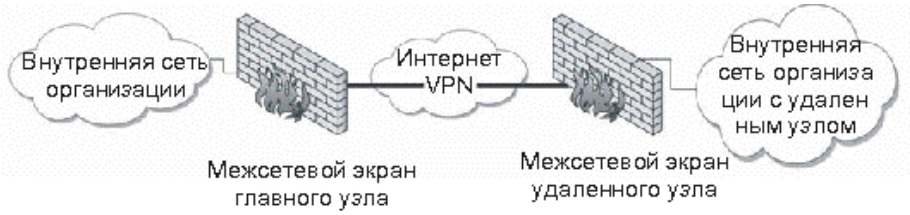


Рис. 15. Межузловое соединение VPN, проходящее через интернет

Основное преимущество узловой VPN - это экономичность.

К недостаткам можно отнести возможность возникновения конфликтов, связанных с адресацией при соединении нескольких организаций.

Выделяется три типа VPN-построителей: аппаратные системы, программные системы и веб-системы.

Аппаратные системы VPN базируются на аппаратной платформе, используемой в качестве VPN-сервера, на которой выполняется программное обеспечение производителя, а также, возможно, некоторое специальное программное обеспечение, предназначенное для улучшения возможностей шифрования. Для построения VPN на системе удаленного пользователя необходимо наличие соответствующего программного обеспечения.

Программные VPN работают на компьютерных системах общего назначения, и могут быть установлены на выделенной для VPN системе либо

совместно с другим программным обеспечением, таким как межсетевой экран. При загрузке программного обеспечения, для поддержки VPN, необходимо обеспечить достаточную мощность аппаратной платформы.

Веб-системы решают главный недостаток большинства пользовательских систем VPN – потребность в установке программного обеспечения на систему-клиент. В качестве VPN-клиентов стали использовать веб-браузеры, т.е. пользователь с помощью браузера подключается к VPN через SSL. SSL обеспечивает шифрование трафика, а подтверждение подлинности пользователя выполняется с помощью средств аутентификации, встроенных в систему.

Таким образом, технология VPN отвечает основополагающим критериям сохранности информации: целостность, конфиденциальность, авторизованный доступ. При правильном выборе VPN обеспечивается масштабирование, то есть использование VPN не создаст проблем роста и поможет сохранить сделанные инвестиции в случае расширения бизнеса. В сравнении с выделенными линиями и сетями на основе Frame Relay виртуальные частные сети не менее надежны в плане защиты информации, однако в 5-10, а иногда и в 20 раз дешевле.

Однако, при использовании VPN происходит падение производительности сети, связанное с криптографической обработкой трафика, проходящего через VPN-устройство. Многими специалистами отмечается, что VPN сети нагружают интернет-сеть на 20%, что делает невозможность использования такой технологии в сетях с низкой скоростью.

Выводы по главе 1

Обычно ресурсы АЭИС не всегда загружаются полностью. Как уже было сказано большинство серверов используют не больше 15% своих максимальных мощностей, а их виртуализация при этом может служить очень мощным привлекательным усовершенствованием. С помощью гипервизоров появляется возможность установки множества ОС и различных приложений в

пределах одного сервера, что, несомненно, повышает загрузку ресурсов сервера, которые будут потребляться совместно.

Использование виртуализации рабочих мест делает сеть важным ресурсом функционирования АЭИС. Поэтому базовые технологии виртуализации сети позволяют большому количеству пользователей проводить работу в пределах единой сетевой инфраструктуры независимо от их физического расположения. Таким образом, применяемые на практике сетевые ресурсы постоянно эксплуатируются с достаточно высокой загрузочной мощностью.

Современное развитие экономики прежде всего требует масштабирование емкостных характеристик сети. С помощью виртуализации развивается новое поколение сетевых решений.

Очевидно, что факторы, которые занимаются стимуляцией виртуализации сети, совершенно не схожи с факторами, которые влияют на виртуализацию серверов. Современная тенденция развития АЭИС направленна больше на ускорение возможности присоединения сервисов и приложений, а так же на упрощение процессов относящихся к широкомасштабному выделению ресурсов. Сети, ориентация которых направленна на приложения, дают большие преимущества, как в сфере эксплуатации сетевых ресурсов, так и со стороны экономии расходов.

Глава II. Разработка инфраструктуры виртуализации автоматизированных экономических информационных систем.

1. Выбор средств виртуализации серверов

Как было уже сказано, аппаратная виртуализация обеспечивает производительность, сравнимую с производительностью невиртуализованной машины. Таким образом, аппаратная виртуализация даёт возможность практического использования в качестве средства виртуализации серверов для экономических информационных систем. К таким средствам виртуализации относятся: KVM и XEN.

Архитектура KVM

KVM (Kernel-based Virtual Machine) — это программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86, которая поддерживает аппаратную виртуализацию на базе Intel VT (Virtualization Technology) либо AMD SVM (Secure Virtual Machine). Распространяется на основе открытых и свободных лицензий и активно поддерживается и развивается усилиями сообщества свободных разработчиков и лидеров отрасли - HP, Intel, Red Hat, IBM. В KVM обеспечивается:

1. Производительность виртуальных машин близкая к производительности при непосредственном исполнении на железе;
2. Возможность живой миграции работающих виртуальных машин между хостами;
3. Поддержка до 160 логических ЦП (центральных процессоров) и до 2 ТБ оперативной памяти на узел с возможностью горячего добавления (hotplug) процессоров;
4. Полная поддержка платформ x86/32, x86/32 с PAE, x86/64, IA64, а так же серверных и настольных операционных систем семейства Windows

5. Интеграция гипервизора в стандартное ядро Linux - эта технология в дальнейшем будет активно поддерживаться и развиваться разработчиками.

Как уже упоминалось выше, технология виртуализации **KVM** (Kernel-based Virtual Machine) — это относительно новая программная технология, основанная на операционных системах Linux, которая позволяет запускать на x86-совместимых процессорах виртуальные машины с различными типами операционных систем. В отличие от других технологий виртуализации, поддержка KVM впервые была интегрирована в стандартное ядро Linux. Поэтому сразу после установки KVM можно запустить операционную систему в пространстве пользователей. Каждая гостевая ОСпредставляет собой отдельный процесс базовой операционной системы (или гипервизора). На рисунке 2 показана схема виртуализации с KVM. В основе лежит аппаратная платформа, которая способна к виртуализации (в настоящее время это Intel VT или AMD-SVM процессор). На «голой» аппаратуре работает гипервизор (ядро Linux с модулем KVM). Этот гипервизор выглядит точно также как стандартное ядро Linux, на котором можно запускать другие приложения. Но это ядро поддерживает также гостевые ОС, загруженные с помощью утилиты kvm. В конечном счете гостевая ОС поддерживает те же самые приложения, что и базовая операционная система. KVM в режиме полной виртуализации, превращает ядро Linux в гипервизор, которое поддерживает аппаратную виртуализацию.

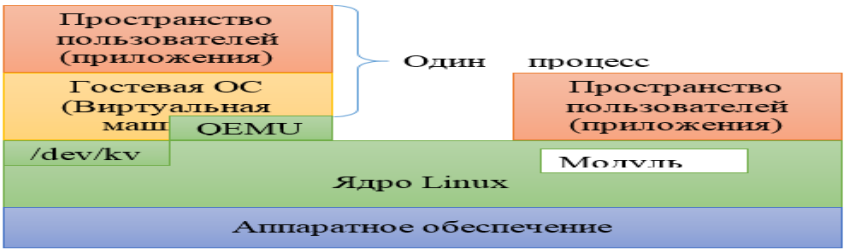


Рис. 16. Компоненты виртуализации с KVM

Как уже упоминалось, KVM представляет собой один из составных элементов системы виртуализации. С помощью KVM виртуализируется

память. Другой элемент виртуализации – это процессор, который призван виртуализировать самого себя для множества операционных систем. И последний элемент в системе виртуализации - это QEMU. Слегка модифицированный процесс QEMU виртуализирует систему ввода/вывода, копия которого выполняется с каждым процессом гостевой операционной системы.

Таким образом, KVM вводит новый режим процессов в Linux в существующее ядро и пользовательские режимы. Новый режим называется *guest* и, как подсказывает его имя, используется для выполнения кодов гостевой операционной системы (или, по крайней мере, некоторых из них). Причём, режим ядра представляет собой привилегированный режим выполнения кодов, в то время как пользовательский режим не является привилегированным (программы, работающие вне ядра). Поэтому режимы выполнения определяются для различных целей, в зависимости от того, что выполняется и с какой целью. Режим guest существует для выполнения кодов гостевой операционной системы, но только для кодов, которые не являются вводом/выводом. В рамках режима guest существуют два стандартных режима, для того чтобы гостевая ОС работала в режиме guest, но поддерживала стандартные режимы: режим ядра и пользовательский режим, для своего ядра и приложений пространства пользователей. Пользовательский режим гостевой операционной системы существует для того, чтобы выполнять операции ввода/вывода, которые управляются независимо.

Выполнение операции ввода/вывода с гостевой операционной системы обеспечивается QEMU. QEMU – это виртуализационная платформа, которая позволяет виртуализировать все оборудование PC, включая диски, графические адаптеры, сетевые устройства. Любые запросы ввода/вывода, которые делает гостевая операционная система,

перехватываются и направляются в пользовательский режим для эмулирования с помощью процесса QEMU.

KVM обеспечивает виртуализацию памяти с помощью /dev/kvm. Каждая гостевая ОС имеет свое собственное адресное пространство, которое устанавливается, когда создается гостевая система. Физическая память, которая назначается для гостевой операционной системы, является в действительности виртуальной памятью процесса. Набор теневых таблиц поддерживается для преобразования с гостевых физических адресов в реальные физические адреса. Процессор также поддерживает процесс преобразования памяти, передавая управление гипервизору (базовому ядру), когда имеется обращение к нераспределенному адресу памяти.

Архитектура XEN

Xen — это монитор виртуальных машин (VMM, Virtual Machine Monitor) или гипервизор (hypervisor) с поддержкой паравиртуализации (para-virtualization) для процессоров x86 архитектуры, распространяющийся с открытым исходным кодом (opensource). Xen может организовать совместное безопасное исполнение нескольких виртуальных машин на одной физической системе с производительностью близкой к непосредственной (native).

Xen обладает функциональностью ПО корпоративного уровня; в нём, в частности, обеспечивается:

1. Производительность виртуальных машин близкая к производительности при непосредственном исполнении на железе;
2. Возможность живой миграции работающих виртуальных машин между хостами;
3. Поддержка до 32 виртуальных процессоров на одну гостевую машину с возможностью горячего добавления (hotplug) процессоров;
4. Поддержка платформ x86/32, x86/32 с PAE, x86/64, IA64, а также частичная поддержка платформ ARM и PPC;

5. Поддержка аппаратной виртуализации для запуска немодифицированных операционных систем (включая Microsoft Windows);
6. Отличная поддержка оборудования (поддерживаются практически все драйверы устройств Linux).

Xen часто сравнивают с разнообразными мониторами виртуальных машин, средствами виртуализации операционной системы, эмуляторами и даже слоями совместимости.

Citrix XenServer относится к программному обеспечению для виртуализации, устанавливаемому непосредственно на «железо» (bar-metal solutions), например, VMware ESX или Microsoft Hyper-V, в отличие от решений hosted solutions, которые устанавливаются на полноценную операционную систему, например VMware Workstation или Microsoft Virtual PC. Таким образом, Citrix XenServer практически не расхо дует ресурсы физического сервера на нужды собственной операционной системы и использует порядка 4-6% от его общей производительности. Схематично архитектура Citrix XenServer представлена на рис. 10. Приведу краткие пояснения:

- **Hardware** – это, собственно, вычислительные ресурсы, т.е. сервер.
- **Hypervisor (Xen Hypervisor)** – это программное обе спечение, которое устанавливается непосредственно на физическое «железо», т.е. на сервер, и образует так называемый уровень абстракции (Abstraction layer), который обеспечивает виртуализацию вычислительных ресурсов (Virtualized Hardware) и позволяет запускать на одном физическом сервере несколько виртуальных, эффективно развязывая их, а также приложения внутри этих виртуальных машин. Hypervisor управляет оперативной памятью и процессорами (RAM/CPU).
- **Control Domain** – это виртуальная машина Linux с наивысшим приоритетом использования вычислительных ресурсов. Эта машина управляет остальным

оборудованием, таким как сетевые адаптеры, устройства хранения данных, в том числе и локальные и т.д. (Drivers), кроме оперативной памяти и процессора. Так как используются обычные драйверы для Linux, поддерживается обширный перечень устройств (список всех протестированных устройств можно посмотреть здесь – http://hcl.xensource.com). Control Domain содержит пакет инструментов (Xen Tool Stack) для управления Citrix XenServer.

- **Linux** – это гостевая виртуальная машина Linux с поддержкой паравиртуализации. Доступ к устройствам хранения данных и к сетевым интерфейсам такая виртуальная машина получает через Control Domain, а к процессорам и оперативной памяти посредством Xen Hypervisor.
- **Windows** – гостевая виртуальная машина Windows. Доступ к устройствам хранения данных и к сетевым интерфейсам такая виртуальная машина, так же как и Linux, получает через Control Domain, а к процессорам и оперативной памяти посредством Xen Hypervisor, используя возможности аппаратной технологии виртуализации процессоров Intel VT и AMD-V, что позволяет увеличить производительность.

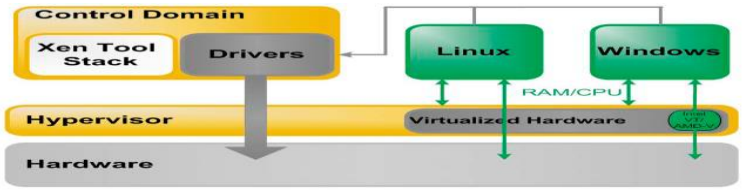


Рис. 17. Архитектура Citrix XenServer

Анализ характеристик производительности средств аппаратной виртуализации XEN и KVM

Тестирование проводилось на двух одинаковых серверах Supermicro со следующими техническими характеристиками:

- Процессор - Intel Xeon E3-1220 (четыре ядра, 3.10Ghz)
- Память - 24GB Kingston DDR3 RAM

- Жёсткие диски 4 диска Western Digital RE-3 160GB объединенные в массив RAID 10

Все тесты проводились на дистрибутиве Linux Fedora 20 (с включенным SELinux) для физических и виртуальных машин.

- Ядро: 3.14.8
- Для KVM: QEMU-KVM 1.6.2
- Для Xen: Xen 4.3.2
- Файловая система: XFS с конфигурацией по умолчанию

Виртуальные машины были созданы с использованием virt-manager, с настройками по умолчанию доступными для KVM и Xen.

В качестве виртуальных дисков использовались необработанные образы, было выделено 8 Гб оперативной памяти, а так же 4 виртуальных процессора. Для гостевых машин Xen оптимизированных паравиртуализированных драйверы PVHVM для обеспечения лучшей производительности диска и сети.

Целью теста служило сравнение производительности виртуальных машин с реальными машинами. Отклонение в производительности между KVM и реальной машиной без виртуализации составляло менее 0,51%.

Результаты теста показали снижение производительности KVM по сравнению с невиртуализированной (реальной) машиной на 1,5% практически во всех тестах. Только два испытания выходят за рамки этой статистики. Один из таких тестов был 7-Zip, где производительность KVM упала на 2,79%. А при другом тесте - PostMark, который имитировал загруженный почтовый сервер, как ни странно, производительность KVM была на 4,11% быстрее, чем на реальном оборудовании (приложение 26).

Как видно из таблицы в приложении 26, значения отклонений производительности XEN от реальной машины по всем видам тестов резко меняются по сравнению с KVM. В среднем Xen показал себя на 2,5% медленнее чем реальная машина, но остальные показатели были гораздо меньше. Тест PostMark был в Xen 14.41% медленнее, чем реальная машина. Лучший результат теста CPU по критерию расчёта выравнивания MAFFT дал KVM, тогда как XEN оказался на втором месте. Основываясь на результаты проведённых тестов, можно утверждать, что производительность KVM почти всегда ниже 2% от производительности физической машины. Xen же медленнее в 2,5% реальной машины в трех из десяти испытаний, но часто имели отклонение 5-7%.

Следует обратить внимание на то, что показатели KVM лучше XEN, но в практическом использовании мало ощутимы. Поэтому можно утверждать, что XEN и KVM приблизительно дают наивысшую производительность по сравнению с другими видами виртуализации.

2. Расчёт оптимальных требований к аппаратному обеспечению для виртуализации серверов

Виртуализация инфраструктуры является сложным процессом, так как необходимо обеспечить приемлемую производительность десятков, а иногда и сотен различных виртуальных серверов с множеством запущенных приложений, дающих различную нагрузку на аппаратную часть.

Поэтому необходимо правильно рассчитать следующие требования к:

1. Оптимальной производительности аппаратного обеспечения для виртуальных серверов;
2. Обеспечению резерва аппаратного обеспечения для быстрой замены или масштабированию инфраструктуры виртуализации;
3. Обеспечению отказоустойчивости и обхода единой точки отказа.

Требования к количеству ядер процессоров на сервере виртуализации можно описать следующей формулой:

$$\boldsymbol{totalkernels} = ((\boldsymbol{kernelscurr} + \sum_{\mathbf{1}}^{\boldsymbol{n}} \boldsymbol{kernelspl}) \times \mathbf{2}) + (((\boldsymbol{kernelscurr} + \sum_{\mathbf{1}}^{\boldsymbol{n}} \boldsymbol{kernelspl}) \times \mathbf{2}) \times \mathbf{0.30}) \quad (1)$$

где:

1. **total**`kernels` - Сумма всех требований к количеству ядер процессоров платформы.
2. `kernels`**curr** - Требования к количеству ядер, исходя из текущей загрузки физических серверов, которых предполагается виртуализировать.
3. `kernels`**pl** - Требования "с нуля" к предполагаемым серверам (если такие имеются).
4. **n** – Количество предполагаемых виртуальных серверов.
5. $(\boldsymbol{kernelscurr} + \sum_{1}^{n} \boldsymbol{kernelspl}) \times \mathbf{2}$ - 50 % резервирование производительности для обеспечения отказоустойчивости и высокой доступности за счёт кластеризации.
6. $(\boldsymbol{kernelscurr} + \sum_{1}^{n} \boldsymbol{kernelspl}) \times \mathbf{2}) \times \mathbf{0.30})$ - 30 % резерв производительности, в случае возникновения необходимости увеличения количества виртуальных серверов, при сохранении полной отказоустойчивости.

Чтобы оценить текущую ИТ-инфраструктуру для различных технологических проектов по виртуализации, необходимо замерить параметр CPU Utilization Counter. CPU Utilization Counter – это усреднённая утилизация процессора за интервал времени. На каждом отрезке, на котором не выполняется Idle Thread (простой), процессор считается занятым какой-то реальной нагрузкой. Этот счётчик – сумма показателей утилизации ЦПУ

пользователем, системой и во время простоя (Idle + User + System utilization, названия могут отличаться на разных платформах) Учитывая, что на большинстве платформ есть отдельный счётчик ЦПУ простоя, можно использовать следующую формулу для расчета потребления ЦПУ: CPU Consumption = 100 - Idle CPU (%).

Требования " с нуля" к планируемым сервисам описаны в документации, предоставляемой производителем, где представлены рассчитанные и проверенные конфигурации. Кроме того, при внедрении любого крупного решения, после выхода его на предполагаемую мощность, необходимо провести нагрузочное тестирование для уточнения пороговых значений.

При использовании отказоустойчивой кластеризации необходимо резервировать 50% доступной производительности на каждом узле кластера на случай сбоя другого узла, чтобы мигрировавшие на него виртуальные машины могли запуститься и работать. Аналогично стоит подходить к резервированию компонентов сетевой инфраструктуры, инфраструктуры обмена данными между серверами и систем хранения данных (SAN или NAS).

Количество реально используемых ядер для нужд предполагаемых виртуальных серверов рассчитывается по следующей формуле:

$$\mathbf{kernels curr} = \sum_{1}^{m} (\boldsymbol{Phy.CoresNumber} \times (\boldsymbol{CpuUtilizationPercent}/\mathbf{100})) \quad (2)$$

где:

1. **kernelscurr** - суммарно используемое серверами количества виртуальных ядер.
2. **Phy.CoresNumber** - количество физических ядер процессора на серверах, которые необходимо виртуализировать.
3. **CpuUtilizationPercent** - суммарный процент утилизации процессоров физических серверов.

4. **m** – Количество физических серверов, подлежащих виртуализации.

Требования к количеству ядер, исходя из текущей загрузки серверов определяется суммой количества реально используемых ядер для нужд предполагаемых виртуальных серверов и количества ядер по требованиям выбранного средства виртуализации.

Требования к размеру оперативной памяти на сервере виртуализации можно описать следующей формулой:

$$\mathbf{total}\boldsymbol{RAM} = ((\boldsymbol{RAM}\mathbf{curr} + \sum_{\mathbf{1}}^{\boldsymbol{n}} \boldsymbol{RAM}\mathbf{pl}) \times \mathbf{2}) + (((\boldsymbol{RAM}\mathbf{curr} + \sum_{\mathbf{1}}^{\boldsymbol{n}} \boldsymbol{RAMpl}) \times \mathbf{2}) \times \mathbf{0.30}) \quad (3)$$

где:

1. **total**RAM – требования к общему размеру оперативной памяти на сервере виртуализации.
2. RAM**curr** - Требования к размеру оперативной памяти, исходя из текущей потребности серверов, которых необходимо виртуализировать.
3. RAM**pl** - Требования " с нуля" к предполагаемым сервисам (если такие имеются).
4. **n** – Количество предполагаемых виртуальных серверов.
5. $(\boldsymbol{RAM}\mathbf{curr} + \sum_{\mathbf{1}}^{\boldsymbol{n}} \boldsymbol{RAM}\mathbf{pl}) \times \mathbf{2}$ - 50 % резервирование производительности для обеспечения отказоустойчивости и высокой доступности за счёт кластеризации.
6. $(\boldsymbol{RAM}\mathbf{curr} + \sum_{\mathbf{1}}^{\boldsymbol{n}} \boldsymbol{RAMpl}) \times \mathbf{2}) \times \mathbf{0.30}$ - 30 % резерв, в случае возникновения необходимости увеличения количества виртуальных серверов, при сохранении полной отказоустойчивости.

При расчёте требований к размеру оперативной памяти следует учитывать, что:

1. Первый гигабайт оперативной памяти, выделенный для каждой виртуальной машины, получает 32 мегабайта административной нагрузки.
2. Каждый последующий гигабайт оперативной памяти, выделенный для виртуальной машины, получает 8 мегабайт административной нагрузки.

Таким образом, расчёт требований к размеру оперативной памяти, исходя из текущей потребности серверов будет представлен следующей формулой:

$$\boldsymbol{RAMcurr} = (\boldsymbol{m} \times \mathbf{32} + \sum_{\mathbf{1}}^{\boldsymbol{m}} (\boldsymbol{RAM} - \mathbf{1}) \times \mathbf{8}) + \boldsymbol{RAMServer} \quad (4)$$

где:

1. **RAMcurr** - Требования к размеру оперативной памяти, исходя из текущей потребности серверов.
2. **m** – Количество виртуальных машин.
3. **RAM** – Доступная оперативная память в каждой виртуальной машине.
4. **RAMServer** – Резервируемая оперативная память для серверной операционной системы.

Причём, обобщённо распределение памяти выглядит следующим образом:

1. Память зарезервированная для гипервизора и родительского раздела.
2. Память зарезервированная под административную нагрузку, возникающую при выделении ОЗУ виртуальным машинам.
3. Доступный сегмент памяти для виртуальных машин.

Требования к размеру дискового пространства пула виртуальных серверов на сервере виртуализации можно описать следующей формулой:

$$\boldsymbol{totalDisk} = ((\boldsymbol{Diskcurr} + \boldsymbol{Diskpl}) \times \mathbf{2}) + (((\boldsymbol{Diskcurr} + \boldsymbol{Diskpl}) \times \mathbf{2}) \times \mathbf{0.30}) \quad (5)$$

где:

1. **totalDisk** – Требования к общему размеру дискового пространства пула виртуальных серверов на сервере виртуализации.

2. **Diskcurr** - Требования к размеру дискового пространства, исходя из текущей потребности физических серверов, которых предполагается виртуализировать.
3. **Diskpl** - Требования "с нуля" к предполагаемым сервисам (если такие имеются).
4. **((Diskcurr + Diskpl)*2)** - 50 % резервирование для обеспечения отказоустойчивости за счёт кластеризации. В этом случае появляется возможность сохранения работоспособности сервисов в случае отказа 1 узла кластера.
5. **(((Diskcurr + Diskpl)*2) *0.30)** - 30 % резерв, в случае возникновения необходимости увеличения количества виртуальных серверов, при сохранении полной отказоустойчивости.

Требования к размеру дискового пространства, исходя из текущей потребности серверов можно рассчитать по следующей формуле:

$$\boldsymbol{Diskcurr} = \sum_{\mathbf{1}}^{\boldsymbol{m}} \left((\boldsymbol{VMDiskSize} + \boldsymbol{VMRam} \times \mathbf{2}) \times \mathbf{1.1} \right) \quad (6)$$

где:

1. **Diskcurr** - Требования к размеру дискового пространства, исходя из текущей потребности серверов.
2. **VMDiskSize** - Выделяемое место дискового пространства для каждого виртуального сервера.
3. $\boldsymbol{VMRam} \times \mathbf{2}$ - Коррекция размера диска при использовании подкачки (SWAP)
4. 1.1 - резерв 10% дискового пространства.
5. **m** – Количество виртуальных машин.

Таким образом, требования к компонентам аппаратной платформы виртуальных серверов описывается общей математической формулой:

(7)

$$\mathbf{Reqtotal} = ((\mathbf{Reqcurr} + \sum_{\mathbf{1}}^{\boldsymbol{n}} \mathbf{Reqpl}) \times \mathbf{2}) + (((\mathbf{Reqcurr} + \sum_{\mathbf{1}}^{\boldsymbol{n}} \mathbf{Reqpl}) \times \mathbf{2}) \times \mathbf{0.30})$$

где:

1. **Reqtotal** - Сумма всех требований к компоненту аппаратной платформы.
2. **Reqcurr** - Требования для виртуализации, исходя из текущей загрузки серверов.
3. **Reqpl** - Требования к " с нуля" планируемым сервисам.
4. **n** – Количество предполагаемых виртуальных серверов.
5. $(\mathbf{Reqcurr} + \sum_{\mathbf{1}}^{\boldsymbol{n}} \mathbf{Reqpl}) \times \mathbf{2}$ - 50 % резервирование производительности для обеспечения отказоустойчивости и высокой доступности за счёт кластеризации.
6. $((\mathbf{Reqcurr} + \sum_{\mathbf{1}}^{\boldsymbol{n}} \mathbf{Reqpl}) \times \mathbf{2}) \times \mathbf{0.30}$ - 30 % резерв производительности в сервера виртуализации, в случае возникновения необходимости увеличения кол-ва виртуальных серверов, при сохранении полной отказоустойчивости.

3. Расчёт оптимальных требований к аппаратному обеспечению терминальных серверов

Нагрузка на терминальный сервер зависит от двух факторов:

1. Количества одновременно подключающихся пользователей, которых можно разделить на две категории:
   a) Обычные пользователи - выполняют одновременно только одно приложение. Их работа чаще всего связана с набором текста (обработка жалоб, прием заказов или служба работы с покупателями). Они создают среднюю нагрузку на сервер.
   b) Опытные пользователи - используют одновременно несколько приложений, переключаясь между ними. В основном, это администраторы, менеджеры, аналитики. Они создают повышенную нагрузку на сервер.

2. Количества программ, которые эти пользователи запускают. В этом случае следует использовать усреднённые значения.

Основной параметр, от которого зависит производительность терминального сервера, является объём оперативной памяти. Причём, чем больше свободной оперативной памяти установлено на терминальном сервере, тем большее количество пользователей он может обслужить.

При расчете объема оперативной памяти для терминальных серверов следует учитывать:

1. На каждом терминальном сервере для системных нужд необходимо иметь не менее 256 Мб оперативной памяти.
2. Требования к объему памяти для каждого пользователя (средняя – 15 Mb, повышенная — 20 Mb).
3. Дополнительные требования приложений (2-4 Mb для каждого приложения).

Расчёт объёма необходимой оперативной памяти терминального сервера производится по следующей формуле:

$$\boldsymbol{RAM} = (\boldsymbol{N} \times \boldsymbol{avgusers} \times \mathbf{15}) + (\boldsymbol{N} \times \boldsymbol{raisuser} \times \mathbf{20}) + (\boldsymbol{N} \times \boldsymbol{X} \times \mathbf{4}) + \mathbf{256}) + [(\boldsymbol{N} \times \boldsymbol{avguse} \quad (8) \quad - (\boldsymbol{N} \times \boldsymbol{raisuser} \times \mathbf{20}) + (\boldsymbol{N} \times \boldsymbol{X} \times \mathbf{4}) + \mathbf{256}] \times \mathbf{0,30}$$

где:

1. **RAM** - Объёма необходимой оперативной памяти терминального сервера.
2. **N** – Количество пользователей.
3. **X** – Количество запущенных приложений пользователями.
4. **avgusers** - Процент пользователей со средней нагрузкой.
5. **raisuser** - Процент пользователей с повышенной нагрузкой.
6. **[N × avgusers ×15) + (N × raisuser × 20) + (N × X × 4) + 256] ×0,30** – так как главным ограничительным фактором для одновременной работы

пользователей является объем оперативной памяти терминального сервера, поэтому оперативную память лучше устанавливать с запасом.

Следует отметить, что однозначного соответствия между количеством подключенных тонких клиентов и мощностью процессора терминального сервера нет. Требуемая мощность процессора зависит от количества одновременно работающих пользователей, общего числа запущенных пользователями приложений и степени их «тяжести». Один и тот-же терминальный сервер с легкостью обслужит 100 пользователей, работающих с офисными приложениями, но уже с приложениями бухгалтерского учёта, которые использую большую базу данных, смогут работать не более 10 пользователей. В итоге для подбора процессоров используют документацию к приложениям, с которыми предполагается работать пользователям. «Легкие» приложения - это классические офисные приложения, Интернет-браузеры и почтовые клиенты. К классу средней «тяжести» можно отнести приложения класса бухгалтерского учёта с не очень большими базами данных (1-2 Гб). К тяжелым приложениям относятся статистические пакеты, приложения OLAP и ERP, которые обрабатывают большие объемы данных. Таким образом, оценив примерный объем задач, который будет выполняться на терминальном сервере, можно определить мощность процессора.

4. Проектирование инфраструктуры виртуализации автоматизированной экономической системы

В предыдущих пунктах данной главы был обоснован выбор платформы виртуализации и представлены рекомендации по расчёту требований к аппаратному обеспечению. Далее следует рассмотреть такой комплекс инфраструктурных решений, которые смогут обеспечить бесперебойную работу виртуализированной среды и позволят усилить эффект от внедрения виртуализации.

Комплекс инфраструктурных решений включает в себя:

1. вычислительную платформу (чаще всего на базе серверов-лезвий);
2. отказоустойчивую сеть хранения данных (SAN);
3. дисковые массивы для хранения данных виртуальных машин и работающих на них приложений;
4. сетевую инфраструктуру передачи данных.

В качестве вычислительной серверной платформы для внедрения решений по виртуализации рекомендуется использовать блейд-серверы на базе процессоров Intel, Xeon и AMD Opteron (рис. 18). Выбор в пользу серверов-лезвий обусловлен тем фактом, что они призваны решать те же задачи, что и виртуализация: повышение эффективности и надежности, снижение энергопотребления, сокращение совокупной стоимости владения.

Рис. 18. Виртуализация на базе блэйд-сервера

Преимущества блейд-систем:

- обеспечение беспрецедентно высокой плотности размещения вычислительных ресурсов, что позволяет экономить дорогостоящее пространство в дата-центре;
- используются уменьшенное число источников питания и вентиляторов, а также интеллектуальные средства управления энергопотреблением, что позволяет сократить энергопотребление более чем на 30 % по сравнению со стоечными серверами аналогичной конфигурации;

- обеспечивается с целью повышения надежности резервирование по схеме N+1, что более экономично по сравнению с полным дублированием, которое используется в традиционных стоечных серверах, когда дублируются компоненты (блоки питания, сетевые интерфейсы и т.д.) каждого сервера.

Отказоустойчивая сеть хранения данных. Задачей централизованного сетевого хранилища является консолидация данных БД, CRM, финансовой системы, совместно используемых файлов, а также системы электронной почты и документооборота предприятий.

Таким образом, консолидация данных – это один из способов повышения производительности, надежности и управляемости информационной системы предприятия. Благодаря переносу данных с внутренних дисков серверов на внешнюю систему хранения, они становятся доступными сразу нескольким серверам, что позволяет реализовывать различные схемы повышения производительности и отказоустойчивости. Кроме консолидации данных, на систему хранения возлагается все больше дополнительных функций.

Современные системы хранения данных обладают развитыми интеллектуальными возможностями и предусматривают множество функций управления данными, что позволяет создавать «мгновенные снимки», оптимизировать работу с данными в зависимости от их типа.

Поэтому отказоустойчивую сеть хранения данных (SAN) эффективно использовать на предприятиях малого и среднего бизнеса с ограниченным бюджетом, но жесткими требованиями к хранению данных, так как достигается баланс между соотношением цены и производительности, отказоустойчивостью, возможностями копирования данных средствами самой системы хранения. Отказоустойчивая сеть хранения данных (SAN) позволяет решать самые разнообразные задачи. На предприятиях эти массивы часто используются для предоставления всем подразделениям различных ИТ-

сервисов: поддержки систем электронной почты, бухгалтерских систем, работы с файлами и т.д.

Для предприятий со штатом 50-250 сотрудников и с малочисленным или отсутствующим ИТ-персоналом оптимально использовать дисковые массивы без SAN, включающих два-три сервера и дисковый массив. Предприятиям же, насчитывающих более 250 сотрудников (производство, банки, аутсорсеры и госсектор) рекомендуется применять эти средства хранения данных в инфраструктуре головных и удаленных офисов, при построении решений резервного копирования и катастрофоустойчивых (DR) конфигураций благодаря поддержке в дисковых массивах удаленной асинхронной репликации данных и доставке «снимков» данных на удаленную систему и последующего их архивирования.

Ещё одной задачей централизованного сетевого хранилища является поддержка виртуальных сред. Современные дисковые массивы обеспечивают высокую производительность, поэтому система может поддерживать больше виртуальных машин и способствует уменьшению времени отклика в приложениях. Благодаря виртуализации на уровне сетей хранения данных (виртуальный диск), появилась возможность повысить производительность за счёт объединения в группы дисков в RAID-группу и распределить данные по всем HDD в массиве.

Таким образом, небольшим предприятиям, решающим задачи по виртуализации серверного оборудования и консолидации хранения, нужны надежные, но в то же время недорогие и простые в обслуживании системы хранения данных.

Таким решением являются дисковые массивы, которые смогут обеспечить высокую производительность, надежность и простоту управления в рамках ограниченного бюджета на информационно коммуникационные технологии.

В случае модернизации оборудования достаточно заменить старые контроллеры на новые, не меняя при этом диски, установленные в систему. После процедуры замены, данные будут автоматически распознаны и готовы к работе. Такой подход существенно экономит бюджет предприятия на развитие и модернизацию инфраструктуры хранения данных, защищает сделанные в нее инвестиции.

В свою очередь, возможности SAN прекрасно дополняют «портативность» виртуальных машин, достигаемую за счет инкапсуляции ОС и прикладного ПО, благодаря которой возможно быстро переносить работающую ВМ с одного физического сервера на другой, обеспечивая тем самым высокую доступность приложений и сервисов, автоматическое перераспределение нагрузки, повышение эффективности использования ресурсов.

Использование SAN в качестве среды хранения данных виртуальных машин позволяет:

- хранить данные на отказоустойчивых дисковых массивах и использовать множественные пути доступа, исключающих наличие единой точки отказа;
- использовать функции средств виртуализации, сводящих к минимуму время плановых и внеплановых перерывов предоставления сервисов;
- использовать функции для балансировки нагрузки;
- максимально эффективно использовать дисковые ресурсы массивов, особенно при использовании Thin Provisioning;
- обеспечивать отличную масштабируемость и защиту инвестиций.

Выбирая систему хранения для виртуализированной среды, стоит обратить внимание на поддержку массивом иерархической модели хранения (Hierarchical Storage Management, HSM). Поддержка HSM позволит разносить образы виртуальных машин по разным уровням хранения в соответствии с

требованиями к производительности и надежности, оптимизируя тем самым стоимость хранения данных.

С учетом того, что рост объема образа виртуальной машины зачастую трудно спрогнозировать, особое значение приобретает использование «интеллектуального снабжения» – Thin Provisioning. Эта технология позволяет администратору выделять каждой виртуальной машине значительный объем дискового пространства, легко покрывающий как нынешние, так и будущие ее потребности. При этом благодаря Thin Provisioning реально выделяется лишь небольшое количество ресурсов, четко соответствующее потребностям в настоящий момент времени. Thin Provisioning повышает эффективность использования дисковых ресурсов с обычных 30-50% до 80-90%, что отлично сочетается с основной целью виртуализации – достижением максимальной эффективности использования существующих ресурсов.

5. Особенности проектирования сетей и систем хранения данных. Проектируя SAN-сеть для виртуализированной инфраструктуры необходимо придерживаться тех же правил, что и при построении SAN для традиционных решений. Однако следует помнить, что запуск на одном физическом хосте нескольких виртуальных машин, особенно с интенсивным дисковым вводом\выводом, серьезно увеличивает загрузку FC-интерфейса. Следовательно, необходимо обеспечивать достаточное количество хост-адаптеров шин (HBA) на сервере виртуализации.

По той же самой причине – сильной загруженности FC-интерфейсов, следует уделять особое внимание и архитектуре SAN. Следует помнить, что многие FC-коммутаторы не способны обеспечить заявленную пропускную способность портов из-за переподписки (oversubscription). В традиционных решениях это не играет большой роли, т.к. в большинстве случаев сервер не загружает 4-гигабитный интерфейс и на половину. В случае же фермы

виртуальных серверов, генерирующих значительную нагрузку на дисковую подсистему, FC-коммутатор может стать узким местом. Для обеспечения отказоустойчивости решения необходимо проектировать SAN-сеть с множественными путями доступа.

Сетевая структура передачи данных. Целью создания структурированной локальной вычислительной сети (ЛВС) предприятия является обеспечение более высокой степени защищенности информации от несанкционированного доступа по сравнению с традиционными (распределенными) сетями, повышение производительности сети и организация более эффективного управления компонентами сети и информационными потоками.

Создание такой ЛВС стало возможным благодаря использованию во всех коммутационных узлах интеллектуальных Ethernet коммутаторов, работающих на канальном (2-м) уровне модели OSI. Кроме того, они должны поддерживать передачу маркированных (tegging) пакетов (стандарт IEEE 802.1q), а также ассиметричные виртуальные локальные сети. Как было уже сказано, виртуальная локальная сеть (VLAN) это сегмент общей сети, который представляет собой объединение портов различных коммутаторов в логическую группу, образующую безопасный автономный широковещательный домен. Основной целью разбиения сети на несколько VLAN является ограничение распространения широковещательных пакетов, поскольку их передача между различными VLAN невозможна. Поэтому исключается развитие широковещательных штормов, которые существенно снижают производительность сети. Наряду с этим использование VLAN дает целый ряд дополнительных преимуществ.

В рамках рассматриваемой концепции построения ЛВС не предусматривается применение маршрутизации/коммутации пакетов на сетевом (3-м) уровне. Такой подход обусловлен изначальным ключевым

требованием обеспечения максимальной защищенности информационных каналов. Поэтому используются имеющиеся технические возможности по изолированию трафика групп пользователей, начиная с возможного низкого уровня, то есть со 2- уровня модели OSI. Кроме того, концепция организации сети описанная ниже, единственно возможная в тех случаях, когда выполнятся передача информации посредством пакетов, не обладающих способностью к маршрутизации (например, NetBIOS или протоколы специального назначения), но требуется высокая степень защищенность сети и изолированность информационных потоков пользователей.

Тем ни менее, данная методика построения сети не исключает применение маршрутизации, которая может рассматриваться как следующий уровень управления трафиком.

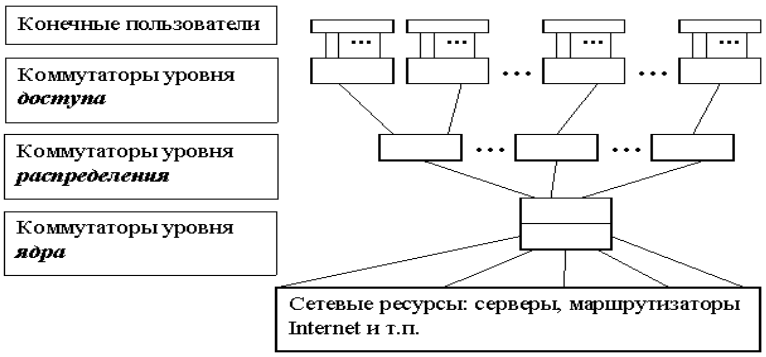


Рис. 19 Функциональная схема структурирования ЛВС

В качестве практического примера рассмотрим схему структурированной ЛВС некоторого предприятия, представленную на рис.19. В ее основу положена древовидная топология.

Коммутаторы уровня ядра устанавливаются в аппаратной (серверной), где также, как правило, располагаются все сетевые ресурсы: серверы (терминальные серверы, файл-серверы, серверы приложений, почтовые серверы и т.п.), маршрутизатор доступа в Internet и т.п. Эти коммутаторы формируют единую производительную информационную магистраль предприятия.

Коммутаторы уровня распределения обычно располагаются в коммутационных помещениях и/или шкафах, установленных на разных этажах или в разных секциях здания. Там же, как правило, располагаются и ***коммутаторы уровня доступа***, к которым подходят линии от персональных компьютеров (ПК) и тонких клиентов пользователей, сетевых принтеров и иных конечных устройств. Коммутаторы уровня доступа обычно служат для увеличения количества портов, требуемых для подключения ПК пользователей.

Коммутаторы «ядра» удобно объединить в *стек*. Стек коммутаторов ядра соединяется с коммутаторами уровня распределения, причем обычно в этом случае используется агрегирование каналов или, иначе, объединение портов в транк, состоящий из нескольких (2 – 8) кабельных линий. Это позволяет, с одной стороны, расширить полосу пропускания соединения, а с другой стороны, обеспечивает повышенную надежность соединения за счет дублирования каналов.

Для того, чтобы иметь возможность обеспечить доступ к различным комбинациям сетевых ресурсов пользователям в зависимости от их прав доступа, организационной принадлежности и политик безопасности без прокладки дополнительных кабельных линий следует применить ***асимметричные VLAN***, построенные на основе меток в дополнительном поле пакета – стандарт IEEE 802.1q. При этом должна быть исключена возможность трафика между компьютерами различных подразделений предприятия.

Экспериментальное исследование вариантов использования асимметричных VLAN. Рассмотрим решение данной задачи на примере простейшей сети, изображенной на рис.20 имеются три сервера S1, S2 и S3, которые подключены к коммутатору «ядра» SW1, который в свою очередь соединен с коммутатором уровня распределения SW2. Для каждого сервера на базе портов 1, 2 и 3 коммутатора SW1 создается отдельный VLAN – соответственно VS1 (включает порт 1), VS2 (включает порт 2) и VS3 (включает порт 3). Создается еще 2 VLAN - VS12 (включает порт 4) и VS23 (включает порт 5), и с помощью настроек асимметричных VLAN формируются пути для трафика, показанные на рисунке стрелками (сплошные линии). В этом случае ПК PC12, который подключен к порту 4 коммутатора SW1, входящему в VLAN VS12, будут доступны данные на серверах S1 и S2, а ПК PC23, который подключен к порту 5 коммутатора SW1, входящему в VLAN VS23, будут доступны данные на серверах S2 и S3. В то же время трафик между VS12 и VS23 будет запрещен (перечеркнутая стрелка); следовательно, информация на S3 и PC23 будет недоступна для PC12, и информация на S1 и PC12 будет недоступна для PC23, а ресурсы сервера S3 будут доступны как PC12, так и PC23.

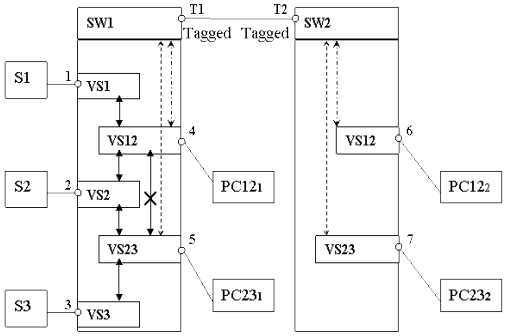


Рис. 20. Использование асимметричных VLAN для создания раздельных ресурсов

Поскольку коммутатор SW1 является коммутатором «ядра», то компьютеры пользователей PC12 и PC23 желательно подключать к другому коммутатору – коммутатору уровня распределения, например SW2, или коммутатору уровня доступа (на рисунке не показан). В этом случае

необходимо распространить действие VS12 и VS23 на коммутатор SW2. Для этого используется такое свойство коммутаторов, как поддержка VLAN на основе меток в дополнительном поле пакета – стандарт IEEE 802.1q.

Казалось бы, можно решить эту задачу следующим образом. На коммутаторе SW2 создать 2 аналогичных VLAN - VS12 иVS23 и соединить эти коммутаторы, например, портами T1 и T2. Отметить эти порты как Tagged (маркирующие) и включить их в состав VS12 и VS23 на обоих коммутаторах. Предполагаемый трафик обозначен на рисунке стрелками из пунктирных линий. Однако, такое решение оказывается неверным, поскольку поддержка асимметричных VLAN ограничена автономными коммутаторами, а была попытка распространить действие асимметричных VLAN на 2 коммутатора. Верное решение приведено на рис.21.

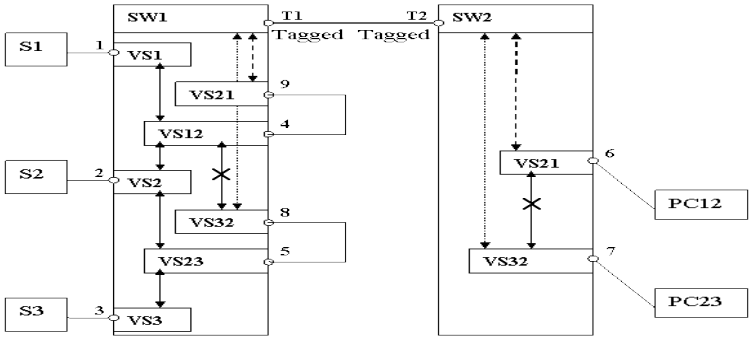


Рисунок 21. Организация передачи сетевых ресурсов на уровень распределения - вариант I

На коммутаторах SW1 и SW2 создаются по 2 дополнительных VLAN – VS21 на портах 6 и 9 и VS32 на портах 7 и 8 соответствующих коммутаторов. Соединяются эти коммутаторы посредством маркирующих (Tagged) портов T1 и T2. Порты 5 и 8, а также 4 и 9 коммутатора SW1 соединяются перемычками. В результате организуется связь между портами различных коммутаторов, принадлежащих одноименным VLAN. Такую организацию виртуальных сетей в дальнейшем будет называться ***распределенной (мостовой)*** VLAN. В этом случае, трафик будет осуществляться так, как показано на рисунке стрелками. Тогда ПК PC12

будет иметь доступ к ресурсам серверов S1 и S2, а ПК PC23 будет иметь доступ к ресурсам серверов S2 и S3; в то же время трафик между PC12 и PC23 оказывается невозможен (перечеркнутая стрелка).

Для увеличения полосы пропускания соединения коммутаторов SW1 и SW2 следует применить агрегирование портов. Коммутатор уровня распределения, например SW2, желательно использовать для создания «распределенных» VLAN, имеющих на этом коммутаторе всего один порт, а для увеличения числа портов, предназначенных для включения в эти VLAN конечных пользователей, следует использовать коммутаторы уровня доступа, подключаемые, например, к портам 6 и 7 коммутатора SW2. Для увеличения полосы пропускания соединения коммутаторов уровня распределения и коммутаторов уровня доступа можно также использовать агрегирование портов.

В ряде случаев возникает необходимость передавать каждый сетевой ресурс на уровень распределения и уже там организовать доступ к этому ресурсу, исключая трафик между его потребителями. Например, необходимо предоставить доступ к Internet ПК подразделений П1 и П2 предприятия, исключив возможность трафика между компьютерами этих подразделений. На рис.22 приведен простейший вариант такой сети.

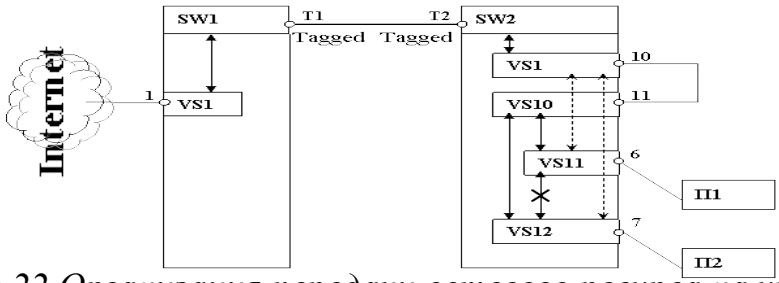


Рисунок 22.Организация передачи сетевого ресурса на уровень распределения - вариант II.

Создаётся «распределенная» VLAN VS1 на обоих коммутаторах описанным выше способом. На коммутаторе SW2 создаётся две пользовательских VLAN подразделений – VS11 и VS12 на портах соответственно 6 и 7.

Как было отмечено выше, поддержка асимметричных VLAN ограничена автономными коммутаторами. Поэтому, если непосредственно включить порты 6 и 7 в состав VS1 в качестве асимметричных VLAN, то трафик, обозначенный стрелками из пунктирных линий, не будет выполняться, поскольку в этом случае асимметричный VS1 будет располагаться на нескольких коммутаторах. Для реализации поставленной задачи на коммутаторе SW2 создается дополнительный VLAN VS10, в состав которого включаются порты 6 и 7, одновременно принадлежащие виртуальным сетям VS11 и VS12, а в состав VS11 и VS12 включается порт 11, принадлежащий VS10; иными словами, создаётся асимметричный VLAN. Порты 10 и 11 коммутатора SW2 соединяются перемычкой. В результате будет соблюдаться требование, ограничивающее действие асимметричного VLAN одним коммутатором, и в то же время будет организован совместный доступ компьютеров подразделений П1 и П2 к Internet при запрете трафика между ПК подразделений (перечеркнутая стрелка).

Оба рассмотренных способа организации передачи сетевых ресурсов с уровня ядра на уровень распределения имеют свои достоинства и недостатки. Первый способ позволяет создавать необходимые комбинации сетевых ресурсов в одном месте и передавать их только на те коммутаторы уровня распределения, где они требуются. В этом случае существенно сокращается трафик между коммутаторами ядра и коммутаторами уровня распределения. Кроме того, данный способ не требует установки на уровне распределения коммутаторов, поддерживающих асимметричные VLAN; достаточно коммутатора, поддерживающего обычный стандарт IEEE 802.1q. К недостаткам такого способа следует отнести необходимость установки на уровне ядра коммутаторов с большим количеством портов или создания стека коммутаторов ядра. Также изменение конфигурации коммутаторов ядра в

процессе эксплуатации, что обычно требуется при изменении топологии сети, может повлиять на работу всех пользователей.

Второй способ удобен тем, что на всех коммутаторах уровня распределения можно иметь сразу все сетевые ресурсы, изначально подключенные к коммутаторам ядра, и по мере необходимости создавать из них требуемые комбинации лишь для групп пользователей, подключенных к конкретному коммутатору уровня распределения. В этом случае переконфигурирование коммутатора может затронуть лишь подключенных к нему пользователей. К недостаткам такого способа следует отнести повышенный трафик между коммутаторами ядра и коммутаторами уровня распределения, поскольку доступ к сетевым ресурсам распространяется сразу на все коммутаторы уровня распределения. Кроме того, все коммутаторы уровня распределения должны поддерживать асимметричные VLAN. Но основной недостаток второго способа заключается в том, что он не позволяет простым образом обеспечивать связь между ПК, подключенными к различным коммутаторам уровня распределения, например, в случае, когда ПК одного подразделения предприятия располагаются на разных этажах и физически подключены к различным коммутаторам уровня распределения.

Опыт построения большой распределенной ЛВС показывает, что при проектировании такой сети следует предусмотреть возможность использования сразу обоих способов организации передачи сетевых ресурсов на уровень распределения, т.е. на уровне ядра следует создавать стек из коммутаторов, поддерживающих асимметричные VLAN, и на уровне распределения использовать также коммутаторы, поддерживающие асимметричные VLAN. Кроме того, для обеспечения повышенной защищенности системы управления настройками коммутаторов ЛВС от несанкционированного доступа следует создать на всех коммутаторах сети отдельную распределенную VLAN и настроить коммутаторы так, чтобы

управлять настройками всех коммутаторов сети было возможно лишь со станции управления сетью, подключенной только в эту VLAN.

Предложенный вариант построения локальной вычислительной сети обладает следующими свойствами: структурность, универсальность и избыточность.

Основными достоинствами рассмотренного варианта построения ЛВС являются:

- Обеспечение высокой степени защищенности информации от несанкционированного доступа за счет создания для каждого подразделения предприятия (или отдельного пользователя) виртуальных локальных сетей (VLAN), ограничивающих трафик в пределах отдельной VLAN.
- Возможность размещения всех основных вычислительных ресурсов (серверов) в одной аппаратной (серверной) позволяет обеспечить требуемый уровень их защищенности от внешних воздействий и удобство обслуживания.
- Возможность предоставления доступа к любым из имеющихся сетевых ресурсов (серверов, систем хранения информации Internet и т.п.) на каждом коммутаторе уровня распределения без проведения работ по прокладке дополнительных кабельных линий.
- Структурная гибкость сети, позволяющая быстро менять строение сети, наращивая или подстраивая ее под изменяющуюся структуру предприятия без проведения работ по прокладке дополнительных кабельных линий.
- Масштабируемость сети, что дает возможность легко наращивать вычислительные ресурсы сети простым подключением дополнительных серверов и других сетевых элементов к стеку коммутаторов «ядра».
- Возможность подключения локальных средств архивизации в любой удобной точке сети, что позволяет расположить устройства архивизации как с учетом минимизации нагрузки на сеть, так и в месте, наиболее защищенном от пожара, затопления и т.п.

- Невысокая стоимость решения и оптимальное соотношение показателя цена/качество, позволяет при малом бюджете развертывать гибкую, высокозащищенную информационную инфраструктуру.

Дальнейшее развитие рассмотренной архитектуры и методологии должно предусматривать наращивание защищенности информационной инфраструктуры. Вполне очевидно, что для этого требуется наличие дублирующих маршрутов в сети, т. е. сеть рассматривается как граф, причем его связность не должна нарушаться при недоступности какого-либо ребра (при отказе информационного канала) или узла.

Исходя из вышеизложенного, общая инфраструктура виртуализации предприятия будет иметь вид, который представлен на рисунке 23.

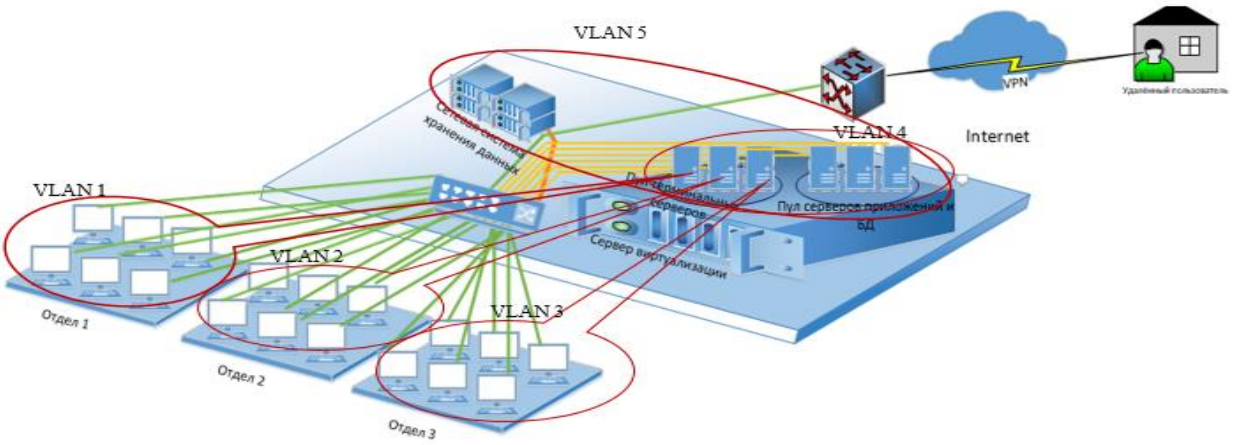


Рисунок 22. Общая инфраструктура виртуализации предприятия.

На сервере виртуализации развёртываются пулы терминальных серверов и серверов приложений, баз данных. Образы виртуальных машин и сами базы данных хранятся на дисковых массивах (SAN). Для каждого отдела создаётся виртуальная машина терминального сервера. Каждый тонкий клиент отдела объединён в соответствующую виртуальную сеть VLAN. Причём, терминальные сервера и сервера приложений в свою очередь объедены в соответствующую VLAN в зависимости от прав доступа отделов.

Сетевая система хранения данных так же выполнена на виртуальной сети и объединена с виртуальными серверами. Удалённый пользователь, подключаясь через VPN сеть, имеет доступ к своему виртуальному рабочему месту и к соответствующим сервисам.

Выводы по главе 2

Таким образом, основными общими причинами использования технологий виртуализации являются:

- **Консолидация серверов и оптимизация инфраструктуры**. С помощью виртуализации можно достичь значительно более эффективного использования ресурсов, поскольку она обеспечивает объединение стандартных ресурсов инфраструктуры в единый пул и преодолевает ограничения устаревшей модели "одно приложение на сервер".
- **Сокращение расходов на физическую инфраструктуру**. Виртуализация позволяет сократить количество серверов и связанного с ними ИТ-оборудования в информационном центре. В результате этого потребности в обслуживании оборудования, электропитании и охлаждении материальных ресурсов сокращаются, и на ИТ-инфраструктуру затрачивается гораздо меньше средств.
- **Повышение гибкости и скорости реагирования системы.** Виртуализация предлагает новый метод управления ИТ-инфраструктурой и помогает ИТ-администраторам затрачивать меньше времени на выполнение повторяющихся заданий (например, на инициацию, настройку, отслеживание и техническое обслуживание).
- **Повышение доступности приложений и обеспечение непрерывности работы предприятия.** Благодаря надежной системе резервного копирования и миграции виртуальных сред целиком без перерывов в обслуживании можно сократить периоды планового простоя и обеспечить быстрое восстановление системы в критических ситуациях.

Использование технологий виртуализации в поставляемых решениях позволит сэкономить до 45% от стоимости оборудования (имеется в виду стоимость x86 серверов) [27]. При этом не учитываются и другие преимущества (помимо консолидации), которые предоставляет виртуализация.

Во внутренней ИТ-инфраструктуре предприятия рекомендуется комплексно внедрять следующие технологии виртуализации:

- серверную виртуализацию;
- виртуализацию рабочих станций;
- виртуализацию систем хранения данных;
- виртуализацию сети.

Глава III. Создание инфраструктуры виртуализации автоматизированных экономических информационных систем.

1. Реализация инфраструктуры виртуализации серверов на базе гипервизора XEN

Настройка домена паравиртуализации Xen-Linux

Процедура подготовки и запуска домена паравиртуализации с Linux в системе виртуализации Xen состоит в следующем. В первую очередь необходимо установить Xen.

```
# yum-y install xen
```

В процессе установки в /boot/grub/menu.lst добавляется блок загрузки ядра с поддержкой Xen. (См. Приложение 2.)

```
# vim /boot/grub/menu.lst
```

Нужно сделать так, чтобы именно ядро Xen загружалось в процессе загрузки сервера. Для этого надо поменять опцию "default=1 на "default=0". Таким образом, будет загружаться ядро перечисленное первым в файле загрузчика Grub (наше ядро с поддержкой Xen). (См. Приложение 3.)

Затем сервер перезагружается

```
# shutdown -r now
```

После загрузки сервера необходимо проверить какое ядро загрузилось в самой ОС. (См. Приложение 4.)

```
#uname -r
```

На этом установка Xen закончена и необходимо создать новую виртуальную машину. (См. Приложение 5)

```
[root@localhost ~]# virt-install – nographic –prompt
```

Основные параметры команды virt-install представлены в приложении таблице 2. После того как будут заданы все эти опции, начнется стандартный процесс инсталляции. (См. Приложение. 6). Чтобы выйти из интерфейса виртуальной машины, можно ввести Ctrl+], а чтобы вернуться - "xm console <имя виртуальной машины>. Чтобы виртуальная машина автоматически

загружалась при перезагрузке сервера, необходимо создать символьную ссылку:

```
ln -s /etc/xen/virt1 /etc/xen/auto/
```

Посмотреть производительность можно командой. (См. Приложение 7).

```
# xen top
```

Настройка домена с аппаратной поддержкой виртуализации (HVM) Xen-Windows

Выполнять *Windows в паравиртуальном режиме,* т.е. без использования аппаратных архитектурных расширений виртуализации, на сегодняшний день нельзя.

Начиная с версии Xen 3.2.0, при наличии в системе аппаратной поддержки виртуализации ввода/вывода Intel VT-d (не путать с виртуализацией процессора VT-x!) существует возможность выполнять монопольное выделение PCI-устройства домену Xen. Раньше это было возможно для паравиртуальных доменов, но было невозможно для HVM-доменов, а именно в таком исполняется Windows.

При выполнении проброс PCI-устройства Windows работает с ним напрямую, на полной скорости, и используют собственные драйвера. Это позволяет обойти проблемы с производительностью, которые есть при эмуляции устройств, а также задействовать все возможности устройства, о которых знает драйвер. Процедура подготовки и запуск домена с Windows в системе виртуализации Xen на платформе с аппаратной поддержкой виртуализации (HVM) состоит в следующем. В первую очередь, для установки Windows, как и любой другой не портированной на Xen системы, необходима поддержка центральным процессором технологии виртуализации Intel®Virtualization Technology (VT) или Pacifica (AMD). Убедиться в наличии поддержки VMX можно командой (См. Приложение 8).:

```
# xm info | grep caps
```

hvm-3.0-x86_32 говорит о том, что XEN успешно обнаружил процессор, который поддерживает технологии Intel VT или AMD-V.

Затем создаётся образ диска Xen:

```
# mkdir -p /virt
# cd /virt
# dd if=/dev/zero of= windows.img bs=1M count=4096
```

И ISO образ установочного диска Windows, поместив диск в CDROM и выполнив команду:

```
#dd if=/dev/cdrom of= /iso/windows.iso
```

В ходе инсталляции Windows доступ к виртуальной машине будет осуществляться через VNC-сервер, встроенный в Xen. Поэтому необходимо установить vnc клиент.

```
# yum install  vncviewer
```

Далее следует создать конфигурационный файл виртуальной машины. (См. Приложение 9).

```
# vim /etc/xen/windows
```

Основные параметры конфигурационного файла виртуальной машины, работающей под управлением гипервизора XEN приведены в приложении таблице 3. Следующий шаг – запуск виртуальной машины и установка Windows.

```
# xm create windows
```

Сразу после создания домена запуститься vncviewer и можно установить Windows. После того как программа установки Windows отформатирует диск и скопирует на него необходимые файлы, выполняется перезагрузка. В соответствии с файлом конфигурации, виртуальная машина при перезагрузке будет завершена, и представится возможность отредактировать параметр boot='c' в файле конфигурации виртуальной машины (См. Приложение 10), после чего виртуальную машину можно будет запускать вновь и продолжить установку Windows (Приложение 27).

```
# xm create windows
```

2. Реализация инфраструктуры виртуализации серверов на базе гипервизора KVM

Настройка пула KVM-Windows

Для создания виртуальной машины Windows, в командной строке используется команда virt-install (См. Приложение 11). Основные параметры команды virt-install гипервизора KVM представлены в приложении таблице 4. Конфигурационный файл созданной машины будет находиться в каталоге /etc/libvirt/qemu/.

После выполнения команды установки виртуальной машины, необходимо подключиться по vnc к хост-серверу для продолжения установки ОС. При этом порт vnc-сервера увеличится на 1. При удалении виртуальной машины порт освобождается и затем выдаётся новой виртуальной машине. Чтобы виртуальная машина стартовала при загрузке хостовой ОС, необходимо выполнить следующую команду:

```
# virsh autostart VM_windows
```

Настройка пула KVM-Linux

Для создания виртуальной машины Linux, в командной строке так же используется команда virt-install (См. Приложение 12). Если процессор не поддерживает аппаратную виртуализацию в --virt-type, то ставится значение qemu (программная виртуализация - медленная) (См. Приложение 13). Параметры virtio дают возможность использовать специальные драйвера для сети и диска, тем самым позволяют улучшить производительность виртуальных машин под управлением KVM. Данные драйвера реализуют «паравиртуализацию». В этом режиме работа устройства не полностью эмулируется гипервизором. Драйвер устройства в виртуальной машине знает о том, что он работает не с настоящим устройством, и взаимодействует с

гипервизором, что обеспечивает большую производительность. Параметр – – extra-args позволяет задать дополнительные аргументы, которые установщик передаст ядру гостевой ОС при загрузке с образов ядра и initrd. Конфигурационый файл созданной машины находится в каталоге /etc/libvirt/qemu/.

В целях управления и создания виртуальных машин возможно использовать графическую утилиту virt-manager. В графическом окне virt-manager можно:

- Создавать гостевые виртуальные машины.
- Копировать уже созданные машины.
- Останавливать или запускать имеющиеся машины.
- Менять параметры конфигурации виртуальных машин (кол-во RAM, CPU и т.д.).
- Производить миграцию с одного гипервизора на другой.
- Осуществлять мониторинг текущей работы виртуальной машины.
- Осуществлять управление гостевой машиной.

Мониторинг производительности и управление работой виртуальной машины показан в приложении 28.

3. Реализация инфраструктуры виртуализации рабочих станций

Файловый сервер будет реализован на базе ОС Linux, а терминальный – Windows и Linux.

Поэтому для реализации терминальной технологии придётся использовать 2 службы и 1 пакет.

- DHCP – предоставляет клиентам сетевые реквизиты
- TFTP – простой способ предоставить доступ к файлам по сети
- Thinstation – пакет образа тонкого клиента

Дистрибутив Thinstation разработан специально для создания тонких клиентов и оснащен всеми необходимыми приложениями, обеспечивающими подключение к сервисам по основным протоколам удаленной работы: Citrix ICA, Microsoft RDP, VNC, NX NoMachine, 2X ThinClient, VMWare View Open client, X11, Telnet, SSH. Систему можно загружать по сети с помощью Etherboot/PXE или внешнего носителя (FDD/CD/HDD/CF/USB-flash). Все настройки производятся централизованно при помощи конфигурационных файлов, что упрощает управление терминалами.

Thinstation - мини-дистрибутив Linux, позволяющий превратить старые компьютеры в полноценные бездисковые тонкие клиенты.

Версия Thinstation 2.2.2, в отличие от Thinstation 5.1 менее требовательна к техническому обеспечению, поэтому не стоит забывать про неё при использовании старых машин в качестве тонких клиентов.

Настройка файлового сервера

Установка и настройка DHCP

Сначала необходимо установить DHCPD и добавить его в автозагрузку.

```
# yum -y install dhcp
# chkconfi g dhcpd on
# vim  /etc/dhcpd.conf
```

Конфигурационный файл DHCP сервера представлен в приложении 14.

Далее производится запуск DHCPD или его перезагрузка, если он был запущен

```
# service dhcpd restart
```

Установка и настройка TFTP

Затем устанавливается пакет tftp-server из репозитория

```
# yum -y install tftp-server
```

Теперь необходимо включить tftp в конфигурации xinetd, для этого в файле /etc/xinetd.d/tftp следует поменять “disable = yes” на “disable = no” и включить xinetd

```
# service xinetd start
```

Следует также, проверить, что порт tftp-сервера прослушивается (tftp работает на порту 69) (См. Приложение 15).

```
# netstat -nlp | grep :69
```

Настройки тонких клиентов

Для начала, необходимо распаковать архив с образом тонкого клиента в /var/lib/tftpboot таким образом, чтобы pxe-загрузчик был доступен по пути “/var/lib/tftpboot/visteh-0.1/pxelinux.0″. Далее создается общий файл настроек для тонких клиентов:

```
# vim  /var/lib/tftpboot/thinstation.conf.network
```

Содержимое файла настроек для тонких клиентов представлен в приложении листинге 43.

Настройка разрешения экрана в Thinstation Linux. В простейшем виде разрешение задаётся параметром SCREEN_RESOLUTION:

```
SCREEN_RESOLUTION="1280x1024"
```

Однако, часто этого недостаточно. Следующим шагом может быть указание режима работы монитора и частоты развёртки. Делается это инструкциями X_MONITOR_MODELINE, SCREEN_HORIZSYNC и SCREEN_VERTREFRESH. Если SCREEN_HORIZSYNC и SCREEN_VERTREFRESH можно найти в руководстве по монитору или в интернете.

При использовании видеокарт Intel с широкоформатным разрешением экрана вам, возможно, понадобится указать видеокарте необходимый VGA-режим. Делается это с помощью утилиты 915resolution, которая включена в образ Thinstation Linux. Пример:

Включается 915 resolution

```
INTELWIDESCREEN_ENABLED=On
```

Параметры, передаваемые утилите при запуске.

- 49 - номер режима
- 1920 1080 - количество точек по горизонтали и вертикали соответственно
- 16 - битность изображения

```
INTELWIDESCREEN_OPTIONS=”49 1920 1080 16″
```

Для более полной информации по параметрам следует набрать команду в консоли тонкого клиента

```
# 915resolution -h
```

Настройка сервера терминалов в Windows Server 2003 R2

Настройка сервера терминалов в Windows Server 2003 R2 Необходимо зайти: «Start» — «Administrative Tools» - «Terminal Service Configuration»**.**

Откроется окно «Terminal Service Configuration». Затем, надо два раза щёлкнуть по папке «Corrections» в правой части окна.

Далее, заходят в свойства RDP-TCP путём нажатия правой кнопки по иконке RDP-TCP и выбора пункта контекстного меню «Property»

На вкладке «General» необходимо изменить значение «Encryption level» на «High»

На вкладке «Sessions» изменяется значение «I do session limit» на «15 minutes» и активируется опция «Override user settings»

На вкладке «Remote Control» ставится переключатель в положение «Interact with the session»

Все, на этом установка сервера терминалов закончена.

Настройка сервера терминалов в Linux

Настройка VNC сервера

VNC расшифровывается, как Virtual Network Computing - это система удалённого доступа к рабочему столу компьютера, использующая протокол RFB (Remote FrameBuffer).

Сначала устанавливается VNC сервер.

```
# yum install vnc-server
```

Далее создаётся пользователь.

```
# useradd user
# passwd user
```

Затем правится конфигурационный файл /etc/sysconfig/vncservers:

```
VNCSERVERS="1:user 2:user2"
VNCSERVERARGS[1]="-geometry 800x600 -nohttpd"
VNCSERVERARGS[2]="-geometry 800x600 -nohttpd"
```

Где -nohttpd отключает web страницу с java-аплетом на порту 580*

После чего необходимо переключиться на ранее созданного пользователя:

```
# su user
# su user
```

и создать пароль для подключения к VNC серверу

```
# vncpasswd
```

Далее редактируется файл запуска графической оболочки.

```
~/.vnc/xstartup
```

В нём необходимо раскомментировать следующие строки:

```
unset SESSION_MANAGER
```

```
exec /etc/X11/xinit/xinitrc
```

Листинг файла представлен в приложении (листинг 44).

Теперь VNC сервер загружается и добавляется в автозагрузку.

```
# service vncserver start
# chkconfig vncserver on
```

Подключение к серверу осуществляется следующим образом: имя-хоста:1

Настройка XRDP сервера

XRDP - это сервер для Unix-систем, реализованный поверх VNC, и предоставляющий доступ к рабочему столу терминального сервера по протоколу RDP. Установка XRDP осуществляется следующей командой:

```
# rpm –ivh /opt/distrib/xrdp-0.4.0-1.el5.rf.i386.rpm
```

Далее для корректной работы сервера необходимо переместить библиотеки в нужный каталог.

```
# cp /usr/lib/xrdp/lib* /lib/
```

Затем настраивается поддержка различных графических оболочек (См. Приложение.16).

```
# vim /usr/lib/xrdp/startwm.sh
```

После чего приводится в надлежащий вид файл запуска XRDP сервера как показано в приложении листинге 46.

```
# vim /usr/share/doc/xrdp-0.4.0/xrdp_control.sh
```

Далее следует запустить XRDP сервер.

```
# /usr/share/doc/xrdp-0.4.0/xrdp_control.sh start
```

Добавление сервера в автозагрузку производится следующим образом:

```
# ln –s /usr/share/doc/xrdp-0.4.0/xrdp_control.sh
/etc/rc5.d/S94xrdp
# ln –s /usr/share/doc/xrdp-0.4.0/xrdp_control.sh
/etc/rc5.d/K11xrdp
```

На этом настройка терминального сервера закончена.

Выводы по главе 3

Таким образом, в рамках настоящей работы был разработан вычислительный комплекс, обеспечивающий необходимую производительность для запуска виртуальных машин на различных платформах с возможностью дальнейшего роста. В результате создано отказоустойчивое эффективное решение по виртуализации IT-инфраструктуры предприятия на базе комплекса open source программных продуктов, развернуто на высокопроизводительном оборудовании.

С точки зрения экономической эффективности, данное решение позволит национальным компаниям и предприятиям существенно повысить эффективность работы и производительность автоматизированных экономических информационных систем, надежность хранения и передачи данных. Внедрение виртуализации позволит значительно сократить затраты на ИТ-инфраструктуру и сэкономить электроэнергию. С технической стороны, существенно уменьшить время обслуживания системы и сроки ввода новых сервисов, упростить и сделать более удобным управление виртуальной инфраструктурой.

Заключение

В заключении следует отметить, что национальные предприятия и компании в настоящее время сталкиваются с целым рядом проблем при проектировании, развёртывании и модернизации своих АЭИС. При этом основной проблемой является сокращение расходов на информационно-коммуникационную инфраструктуру. В этих условиях на первый план выходят вопросы экономии, грамотного использования ресурсов, оптимизации вычислительных процессов. Разработка и реализация долгосрочных стратегий и планов по развитию вычислительных комплексов предприятий и компаний значительно усложняются. Поэтому всё больше они стараются сосредоточиться на направлениях, не требующих значительных капитальных вложений или способных в краткосрочной перспективе вернуть инвестиции. Одним из таких направлений является виртуализация. Данная технология с точки зрения большинства лидеров ИТ рынка, позволяет существенно сократить совокупную стоимость владения программными решениями. При внедрении новых технологий очень сложно просчитать отдачу, прибыльность.

Краткие выводы по результатам диссертационного исследования заключается в следующем:

1. Технология полной виртуализации отвечает требованиям новых запросов предприятий. Виртуализация на уровне операционной системы будет поддерживаться и будет востребованной для клиентов с меньшими запросами и приложений, которые можно адаптировать под возможности данной технологии. Например Oracle адаптирует свои решения под гипервизор Xen.

2. Анализируя результаты сравнительных тестов, можно прийти к выводу, что Linux KVM является самыми быстрыми и самыми надежными средством виртуализации. Причём, следует учитывать, что на практике виртуализация

Xen будет функционировать приблизительно на таком же уровне как KVM для многих вычислительных рабочих нагрузок.

3. При анализе работы гипервизоров выяснилось, что программная виртуализация не подходит для автоматизированных экономических информационных систем. В частности не подходит по требованиям безопасности и быстродействия. Для высоконагруженных проектов лучше всего подходит KVM и XEN, поскольку ресурсы, выделенные конкурирующим виртуальным машинам, хорошо распределялись при больших нагрузок.

4. На предприятиях чаще всего используется клиент-серверная технология т.е. информация не обрабатывается на ПК пользователя и он не нуждается в высокой производительности. Поэтому считается оптимальным использовать в этих целях терминальные решения. Виртуализация рабочих станций решает задачу эффективного администрирования ИТ-инфраструктуры из единого центра, а также снижения затрат на обслуживание высокопроизводительных компьютеров. Среди преимуществ виртуализации рабочих станций следует также отметить следующее:

- Снижение затрат на ПО для пользователей. Особенно это важно для крупных компаний, содержащих обширный парк техники. После проведения виртуализации рабочих столов не понадобится приобретать ОС, почтовые приложения, антивирусные программы и др.

- Использование более дешевой техники вместо дорогостоящих рабочих станций. Производительность рабочего компьютера пользователя должна быть достаточной лишь для трансляции изображения, передаваемого с сервера. Поэтому дорогие ПК могут быть успешно заменены на небольшие станции или морально устаревшие ПК, отлаженные для работы в терминальном режиме.

- Возможность удаленного доступа. Терминальная схема работы позволяет подключаться к рабочему месту с любого компьютера с выходом в Интернет. Это удобно для тех сотрудников, которые находятся в командировке. Данные при этом передаются в зашифрованном виде, что обеспечивает высокий уровень защищенности.

- Отсутствие сбоев при работе. Высокопроизводительный сервер, обладающий высокой мощностью, может поддерживать множество процессов одновременно. Рабочая станция же может быть даже выключена (например, в случае перебоев с электроэнергией), но потери данных при этом не произойдет.

- Простота эксплуатации. С точки зрения пользователя ИТ-инфраструктура не будет отличаться от классической. К рабочей станции можно подключать внешние устройства и использовать с нее принтер и мн. др.

Следует отметить, что в ходе подготовки данной работы нами была создана инфраструктура виртуализации на базе тонких клиентов и с использованием средств виртуализации серверов KVM и XEN. На основе полученных данных нами были разработаны расчёты оптимальных требований к аппаратному обеспечению для виртуализации серверов и терминальных серверов.

Список литературы

Нормативно-правовые документы:

1. Доклад Президента Республики Узбекистан И.А. Каримова на заседании Кабинета Министров, посвященном итогам социально-экономического развития в 2014 году и важнейшим приоритетным направлениям экономической программы на 2015 год. Народное слово 2015 год.
2. Постановление Президента Республики Узбекистан от 21.03.2012 г «О мерах по дальнейшему внедрению и развитию современных информационно-коммуникационных технологий» Собрание законодательства Республики Узбекистан, 2012 г., № 13
3. Закон Республики Узбекистан «Об информатизации», Lex.uz
4. Постановление Кабинета Министров Республики Узбекистан «Об утверждении положения о Министерстве по развитию информационных технологий и коммуникаций Республики Узбекистан» (Собрание законодательства Республики Узбекистан, 2015 г., № 15, ст. 178)
5. Постановление Президента Республики Узбекистан «О мерах по дальнейшему развитию Национальной информационно–коммуникационной системы Республики Узбекистан» от 27.06.2013г. № ПП–1989.

Учебники и учебные пособия:

6. Chris Wolf, Erick M. Halter . Virtualization: From the Desktop to the Enterprise. Apress, 2005.
7. Браун С.Виртуальные частные сети. Лори.[RUS,504с.,2001]
8. Запечников С. Основы построения виртуальных частных сетей.Телеком.[RUS,249с.,2003].
9. Захватов М.Построение виртуальных частных сетей (VPN) на базе технологии MPLS.Cisco.[RUS,52с.,,2004]
10. Фортенбери Т.Проектирование виртуальных частных сетей в среде Windows 2000.Вильямс.[RUS,320с.,2002]

11. Чефранова А.Технология построения виртуальных защищенных сетей ViPNet версии 3.0
12. Митч Таллок - Решения Майкрософт для виртуализации: от настольного компьютера до центра обработки данных(Understanding Microsoft Virtualization Solutions from the Desktop to the Datacentre). Корпорация Майкрософт, 2011
13. Дмитрий Тихович - Технологии виртуализации VMware: динамическая ИТ-инфраструктура уже сегодня. VMware Россия и СНГ 01,2008
14. Создание всесторонней комплексной стратегии виртуализации. Корпорация Майкрософт, 2007
15. Наталия Елманова, Сергей Пахомов — Виртуальные машины 2007.
16. G'ulomov S.S., Xodiev B.Yu., Begalov B.A. va boshq. Informatika va axborot texnologiyalari. Darslik. – T.: Fan, 2010. – 742 bet.
17. Исаев Г. Н. Информационные системы в экономике: учебник.- М.: ИНФРА. 2009.-462 стр.
18. Титоренко Г.А. Автоматизированные информационные технологии в экономике: Учебник. – М.: Компьютер, ЮНИТИ, 2007. – 400 стр.
19. Банк В.Р., Зверев В.С. Информационные системы в экономике. Учебник. - М.:Экономист, 2005. - 477стр.
20. Уткин Б.Б., Балдин К.В. Информационные системы и технологии в экономике. Учебник. М.: ЮНИТИ-Данс, 2005.
21. Ғуломов С.С., Алимов Р.Х., Лутфуллаев Х.С. ва бошқ. Ахборот тизимлари ва технологиялари: ОЎЮ талабалари учун дарслик. - Т.: "Шарқ", 2000. – 592 бет.

Периодические издания, сборники и отчеты:

22. Журнал "Information Security/ Информационная безопасность" #6, 2009
23. АТМ и альтернативы на магистрали глобальной сети, LAN/Журнал сетевых решений №7, 1999.

24. Trusted Enterprice Manager 2.0. Управление корпоративной сетью Windows NT, LAN/Журнал сетевых решений №5.
25. Журнал «Системный администратор» №№ 65,71,72,79,80,84-87,92-94, 101,102,106,107,109
26. Жебрак С.М. Повышение надёжности и эффективности автоматизированных экономических информационных систем путём виртуализации серверов и рабочих мест. Сборник статей VII международной научной конференции «Приоритетные направления в области науки и технологии в XXI веке» 2014 г.
27. Назаров А.И., Жебрак С.М. Применение технологии «Тонкий клиент» в системе образования. Сборник статей международной научной конференции «Ахборот ва телекоммуникация технологиялари муаммолари». 2015 г.
28. Жебрак С.М., Файзиев М.М. Расчёт требований к аппаратному обеспечению виртуализации серверов в автоматизированных экономических системах. Сборник статей международной научной конференции «Радиотехника, телекоммуникация ва ахборот технологиялари муаммолари ва келажак ривожи» 2015 г.

Интернет сайты:

29. Технический центр виртуализации Microsoft: http://technet.microsoft.com/ru-ru/virtualization
30. Виртуальный Linux. Обзор методов виртуализации, архитектур и реализаций. [Электронный ресурс]. – Режим доступа: http://www.ibm.com/developerworks/ru/library/l-linuxvirt/index.html
31. Виртуализация ОС [Электронный ресурс]. – Режим доступа: http://www.parallels.com/ru/products/virtuozzo/os/
32. Сергей Озеров, Александр Карабуто «Технологии виртуализации: вчера, сегодня, завтра» http://citforum.ru/operating_systems/virtualization/part2.shtml

33. Keith Adams, Ole Agesen A Comparison of Software and Hardware Techniques for x86 Virtualization
http://www.vmware.com/pdf/asplos235_adams.pdf
34. AMD Virtualization Solutions http://enterprise.amd.com/us-en/AMD-Business/Business-Solutions/Consolidation/Virtualization.aspx
35. Intel Virtualization Technology
http://developer.intel.com/technology/virtualization/index.htm
36. Debunking Blue Pill
myth http://www.virtualization.info/2006/08/debunking-blue-pill-myth.html

Приложение 1

Описание фактора	Тонкие клиенты	Персональные компьютеры
Администрирование	Централизованное, с помощью ПО, поставляемого бесплатно в комплекте; простая диагностика проблем	Децентрализованное, требует дополнительных средств управления, занимает больше времени и ресурсов; сложная диагностика отказов ввиду более сложной конструкции
Безопасность данных и компьютера	Очень высокая, т.к. приложения исполняются на сервере, невозможность их изменения пользователями, кража устройства не приводит к потере данных; простой backup (на сервере)	Низкая, ввиду обилия дополнительных программ, сложности их взаимодействия и взаимного влияния
Создание нового рабочего места	10 минут	4 часа
Создание дополнительного узла резервирования	Дополнительный сервер 2 часа на развертывание	Невозможно
Рабочее место	Географически не привязано	Фиксированное
Ошибки пользователя	Ограничены теми приложениями, с которыми работают пользователи	Высокий уровень обусловлен множеством и сложностью установленных приложений
Цикл обновления парка (бюджет)	8-10 лет	3-4 года
Надежность и ремонтопригодность	Более высокая ввиду отсутствия вращающихся частей, а также более комфортного термального режима работы компонент; отстутствует	Менее высокая (наиболее часто отказывают вентиляторы, HDD , блоки питания). Растет с повышением температуры. Требуется склад запчастей для ремонта.

	необходимость поддержки склада запчастей	
Уровень шума	0 dB	Около 25 dB
Энергопотребление (экономия оплаты электроэнергии и необходимость использования UPS)	12-30 Вт	70 -350 Вт
Габариты	¼ площади ПК <1/15 объема ПК	
Необходимость и стоимость проведения upgrade по мере наращивания программного обеспечения	Низкая, т.к. upgrade производится на сервере	Высокая, требует upgrade всего парка ПК
Легкость внедрения	Установка и подключение: 10 минут	От 30 минут до 2-3 часов
Общая стоимость владения за время эксплуатации TCO (5 лет)	На 60-70% меньше, чем у ПК (в отдельных случаях до 80% снижения общих затрат)	

Приложение 2

```
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-164.el5xen)
        root (hd0,0)
        kernel /xen.gz-2.6.18-164.el5
        module /vmlinuz-2.6.18-164.el5xen ro root=/dev/VolGroup00/LogVol00 rhgb quiet
        module /initrd-2.6.18-164.el5xen.img
title CentOS (2.6.18-164.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-164.el5 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
        initrd /initrd-2.6.18-164.el5.img
```

Приложение 3

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-164.el5xen)
        root (hd0,0)
        kernel /xen.gz-2.6.18-164.el5
        module /vmlinuz-2.6.18-164.el5xen ro root=/dev/VolGroup00/LogVol00 rhgb quiet
        module /initrd-2.6.18-164.el5xen.img
title CentOS (2.6.18-164.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-164.el5 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
        initrd /initrd-2.6.18-164.el5.img
```

Приложение 4

```
[root@localhost ~]# uname -r
2.6.18-164.el5xen
```

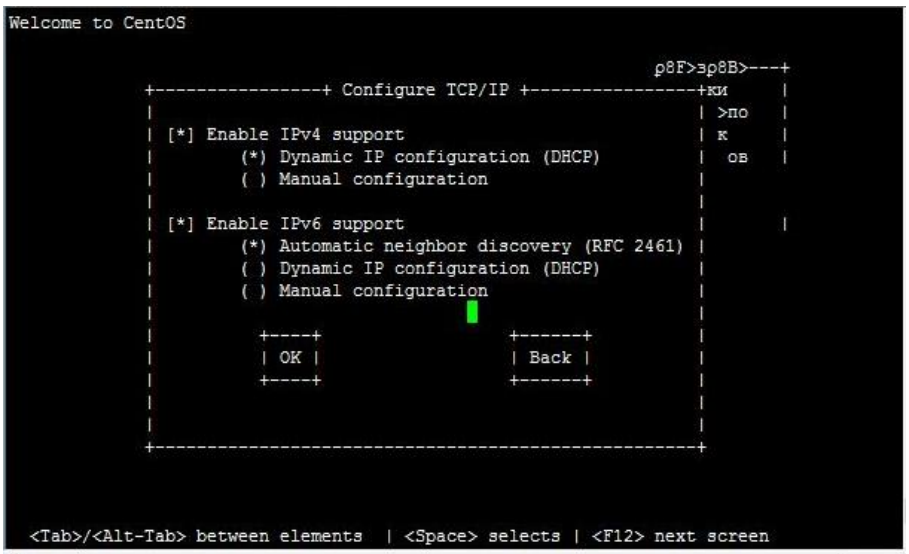
Приложение 5

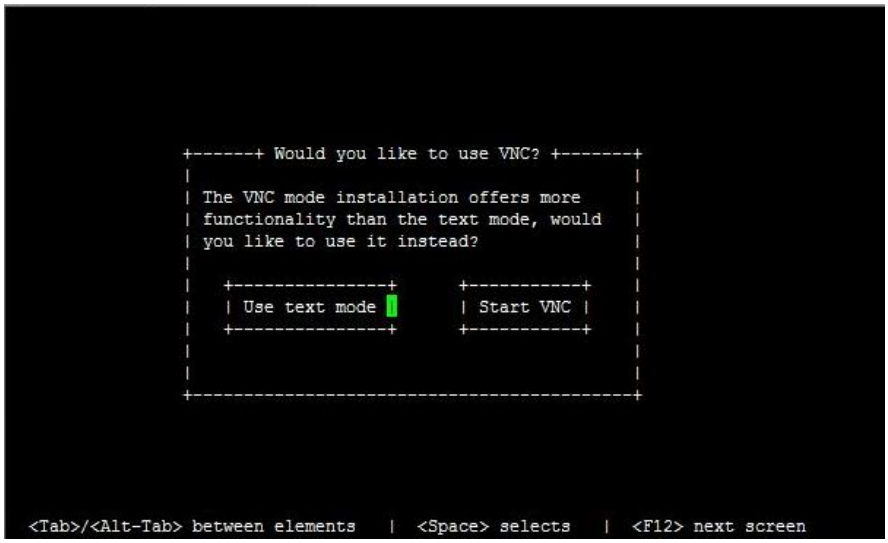
```
What is the name of your virtual machine? virt1
 How much RAM should be allocated (in megabytes)? 120
 What would you like to use as the disk (file path)? /vm/virt1.img
 How large would you like the disk (/vm/virt1.img) to be (in gigabytes)? 4
 What is the install URL? /cd
```

Приложение 6

```
[root@localhost ~]# virt-install --nographics --bridge=BRIDGE --prompt
What is the name of your virtual machine? virt1
 How much RAM should be allocated (in megabytes)? 120
 What would you like to use as the disk (file path)? /vm/virt1.img
 How large would you like the disk (/vm/virt1.img) to be (in gigabytes)? 4
 What is the install URL? /cd


Starting install...
Retrieving file .treeinfo...                          |  413 B     00:00
Retrieving file vmlinuz...                            | 2.1 MB     00:01
Retrieving file initrd.img...                         | 6.6 MB     00:03
Creating storage file...                              | 4.0 GB     00:00
```

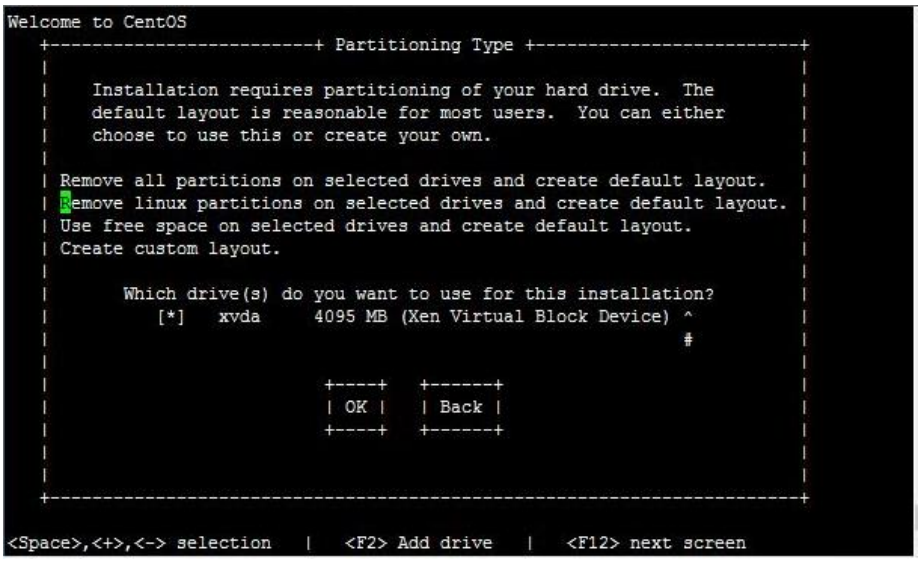
```
Welcome to CentOS

                +---------+ Choose a Language +---------+
                |                                       |
                | What language would you like to use   |
                | during the installation process?      |
                |                                       |
                |       Northern Sotho          ^       |
                |       Oriya                   :       |
                |       Polish                  :       |
                |       Portuguese              :       |
                |       Portuguese(Brazilian)   #       |
                |       Punjabi                 :       |
                |       Russian                 :       |
                |       Serbian                 v       |
                |                                       |
                |               +----+                  |
                |               | OK |                  |
                |               +----+                  |
                |                                       |
                |                                       |
                +---------------------------------------+

  <Tab>/<Alt-Tab> between elements  | <Space> selects | <F12> next screen
```

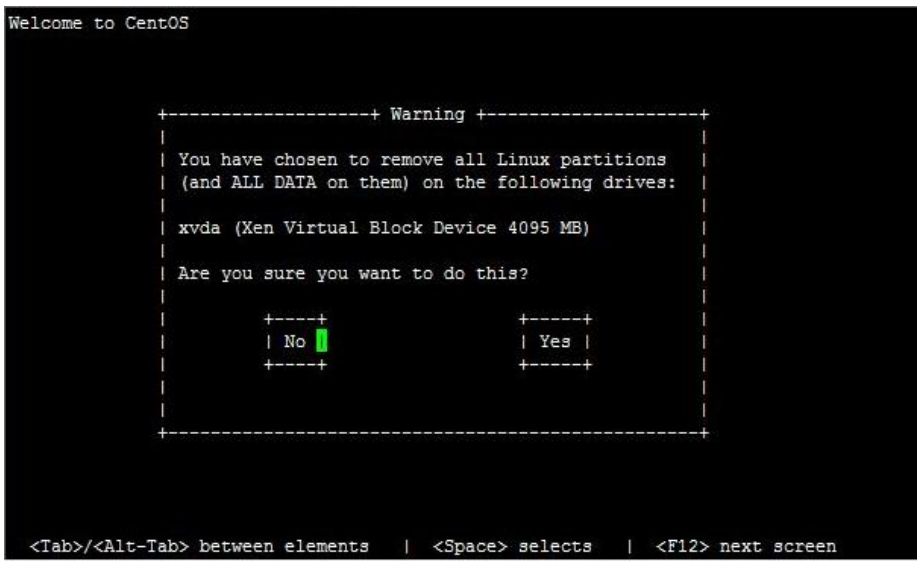

Welcome to CentOS
+----------------+ Configure TCP/IP +----------------+
| [*] Enable IPv4 support |
| (*) Dynamic IP configuration (DHCP) |
| ( ) Manual configuration |
| [*] Enable IPv6 support |
| (*) Automatic neighbor discovery (RFC 2461) |
| ( ) Dynamic IP configuration (DHCP) |
| ( ) Manual configuration |
| OK | | Back |
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen


+------+ Would you like to use VNC? +-------+
| The VNC mode installation offers more |
| functionality than the text mode, would |
| you like to use it instead? |
| Use text mode | | Start VNC |
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

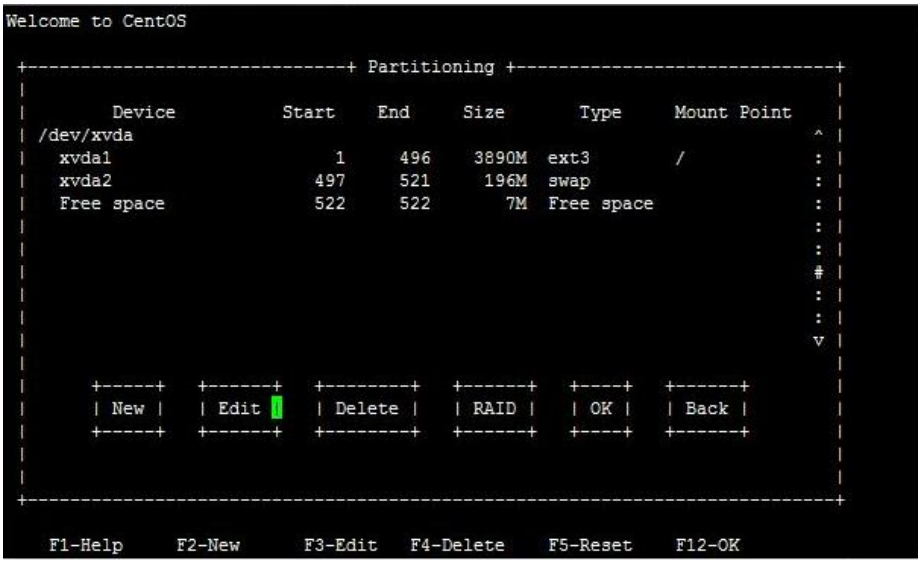

Welcome to CentOS
+------------------+ CentOS +-------------------+
| Welcome to CentOS! |
| OK |
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

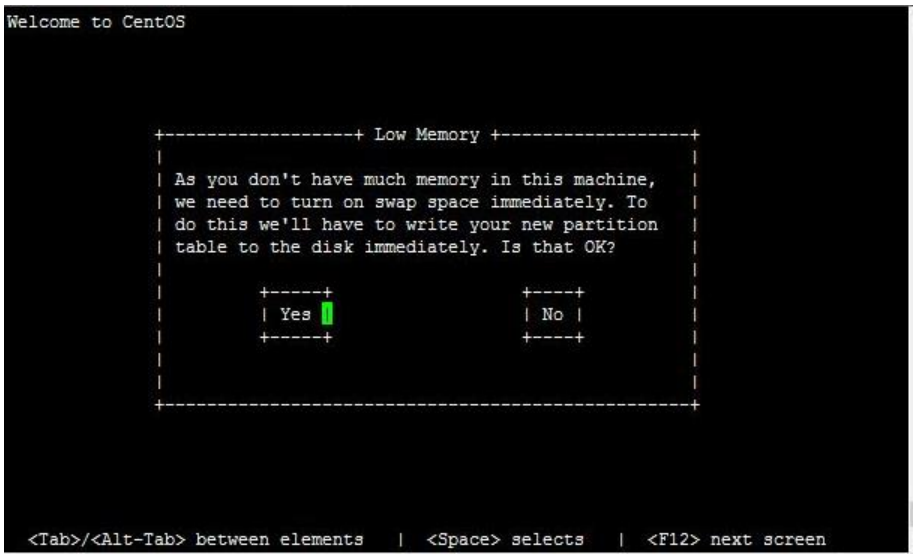

Welcome to CentOS
CentOS
Welcome to CentOS!
OK
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

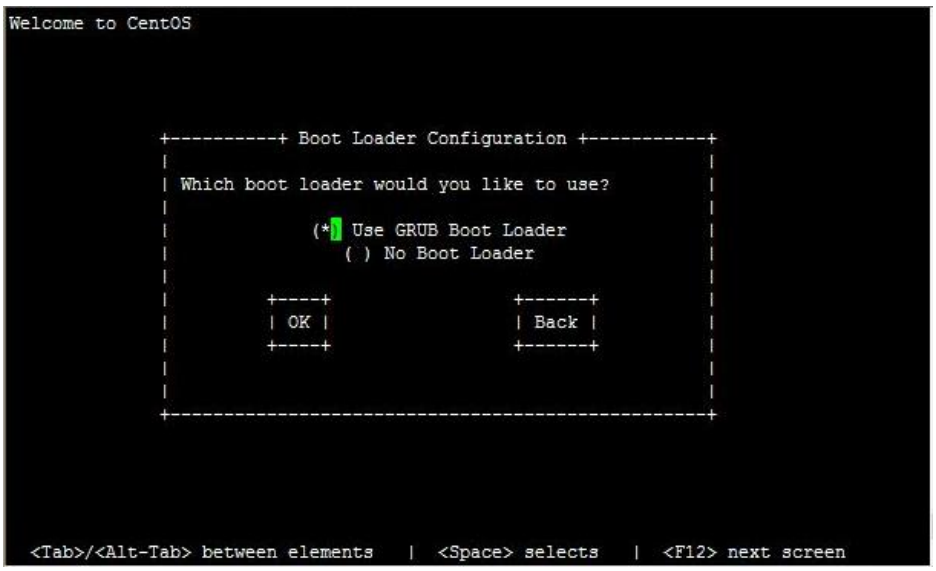

Welcome to CentOS
Partitioning Type
Installation requires partitioning of your hard drive. The default layout is reasonable for most users. You can either choose to use this or create your own.
Remove all partitions on selected drives and create default layout.
Remove linux partitions on selected drives and create default layout.
Use free space on selected drives and create default layout.
Create custom layout.
Which drive(s) do you want to use for this installation?
[*] xvda 4095 MB (Xen Virtual Block Device)
OK
Back
<Space>,<+>,<-> selection | <F2> Add drive | <F12> next screen

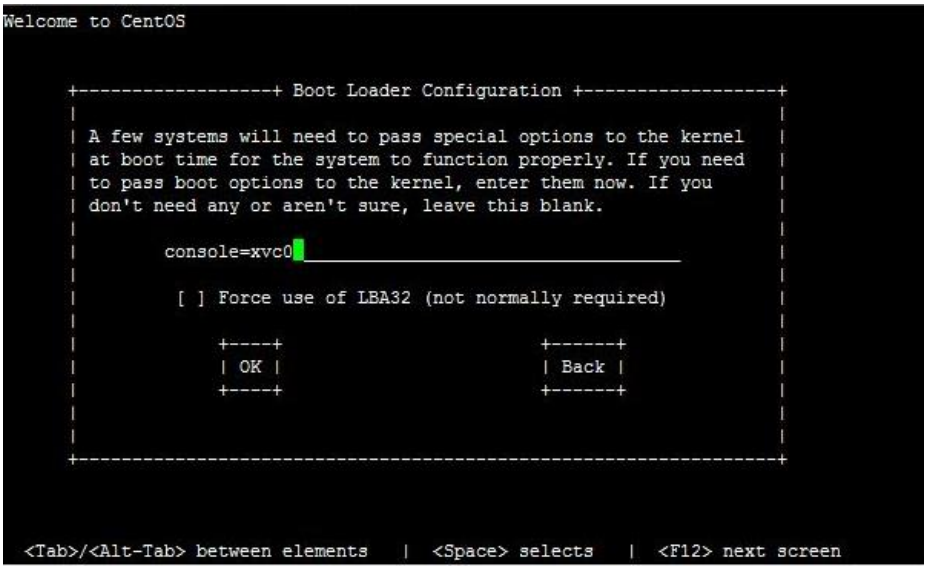

Welcome to CentOS
Warning
You have chosen to remove all Linux partitions (and ALL DATA on them) on the following drives:
xvda (Xen Virtual Block Device 4095 MB)
Are you sure you want to do this?
No
Yes
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen


Welcome to CentOS
Review Partition Layout
Review and modify partitioning layout?
Yes
No
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

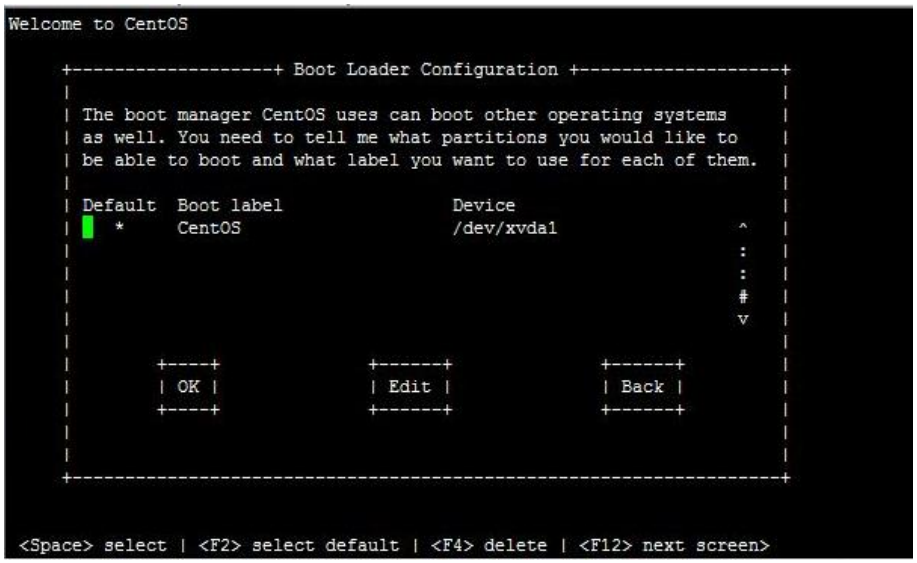

Welcome to CentOS
Partitioning
Device Start End Size Type Mount Point
/dev/xvda
xvda1 1 496 3890M ext3 /
xvda2 497 521 196M swap
Free space 522 522 7M Free space
New
Edit
Delete
RAID
OK
Back
F1-Help F2-New F3-Edit F4-Delete F5-Reset F12-OK

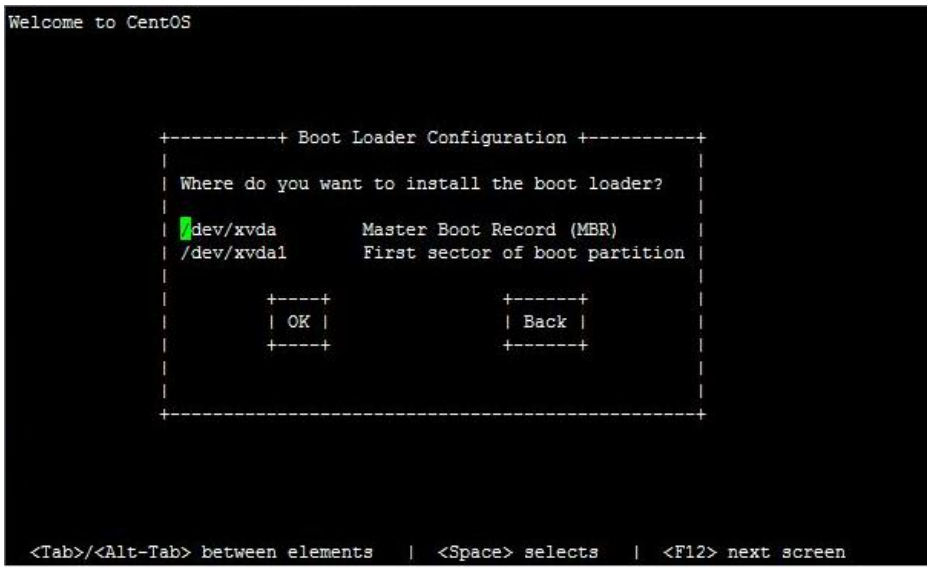

Welcome to CentOS
Low Memory
As you don't have much memory in this machine,
we need to turn on swap space immediately. To
do this we'll have to write your new partition
table to the disk immediately. Is that OK?
Yes
No
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen


Welcome to CentOS
Boot Loader Configuration
Which boot loader would you like to use?
(*) Use GRUB Boot Loader
( ) No Boot Loader
OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

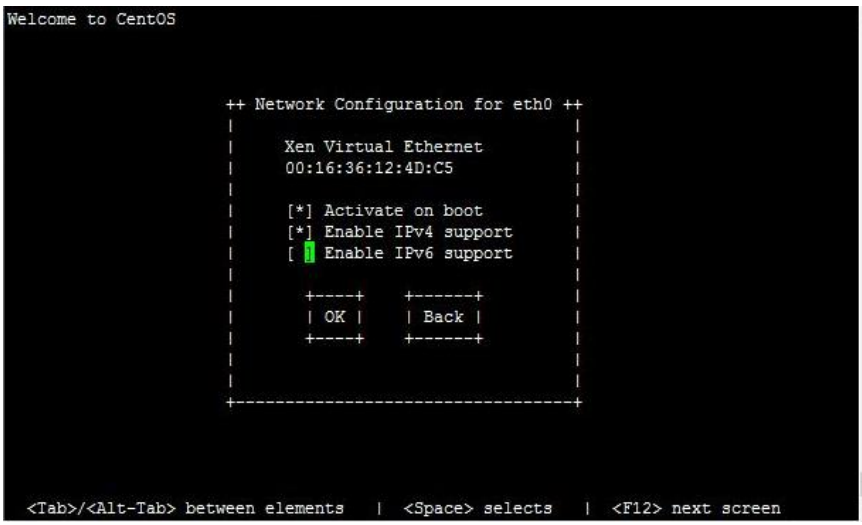

Welcome to CentOS
Boot Loader Configuration
A few systems will need to pass special options to the kernel at boot time for the system to function properly. If you need to pass boot options to the kernel, enter them now. If you don't need any or aren't sure, leave this blank.
console=xvc0
[ ] Force use of LBA32 (not normally required)
OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

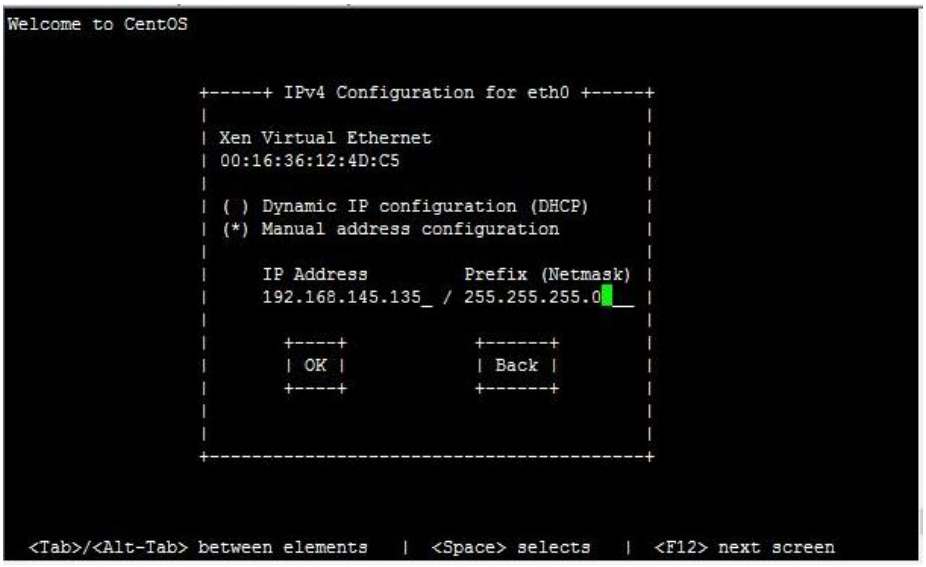

Welcome to CentOS
Boot Loader Configuration
A boot loader password prevents users from passing arbitrary options to the kernel. For highest security, we recommend setting a password, but this is not necessary for more casual users.
[ ] Use a GRUB Password
Boot Loader Password:
Confirm:
OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

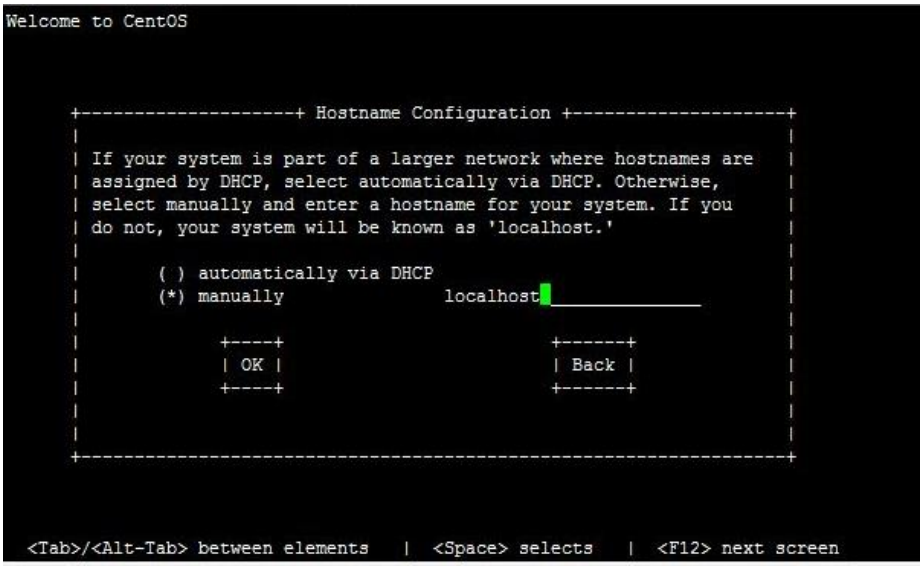

Welcome to CentOS
Boot Loader Configuration
The boot manager CentOS uses can boot other operating systems as well. You need to tell me what partitions you would like to be able to boot and what label you want to use for each of them.
Default
Boot label
Device
*
CentOS
/dev/xvda1
OK
Edit
Back
<Space> select | <F2> select default | <F4> delete | <F12> next screen>

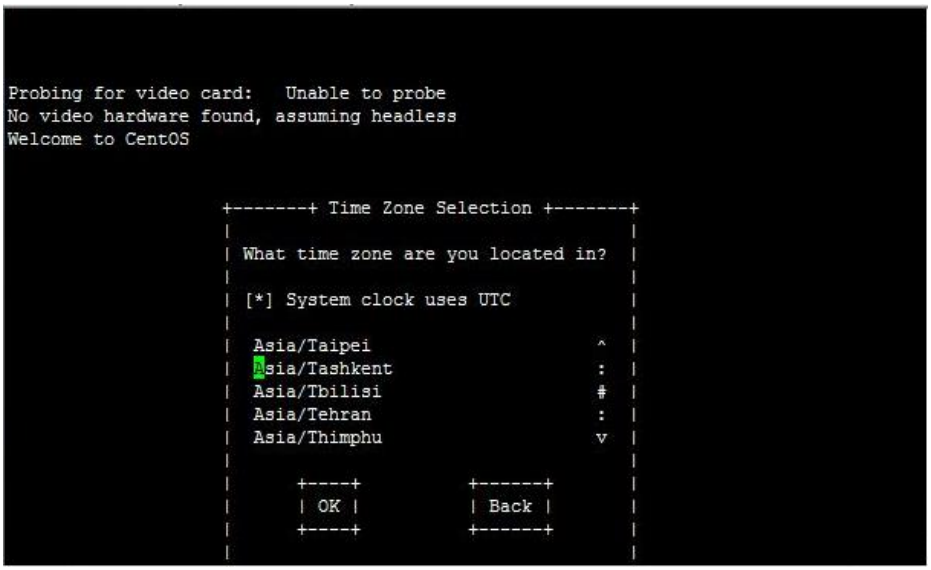

Welcome to CentOS
Boot Loader Configuration
Where do you want to install the boot loader?
/dev/xvda
Master Boot Record (MBR)
/dev/xvda1
First sector of boot partition
OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

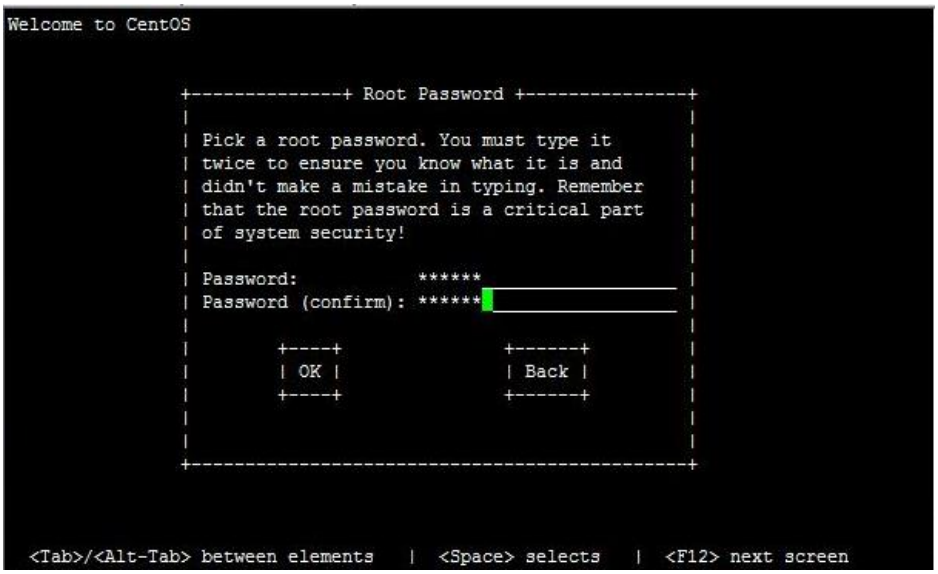

Welcome to CentOS
Configure Network Interface
Would you like to configure the eth0 network interface in your system?
Yes
No
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

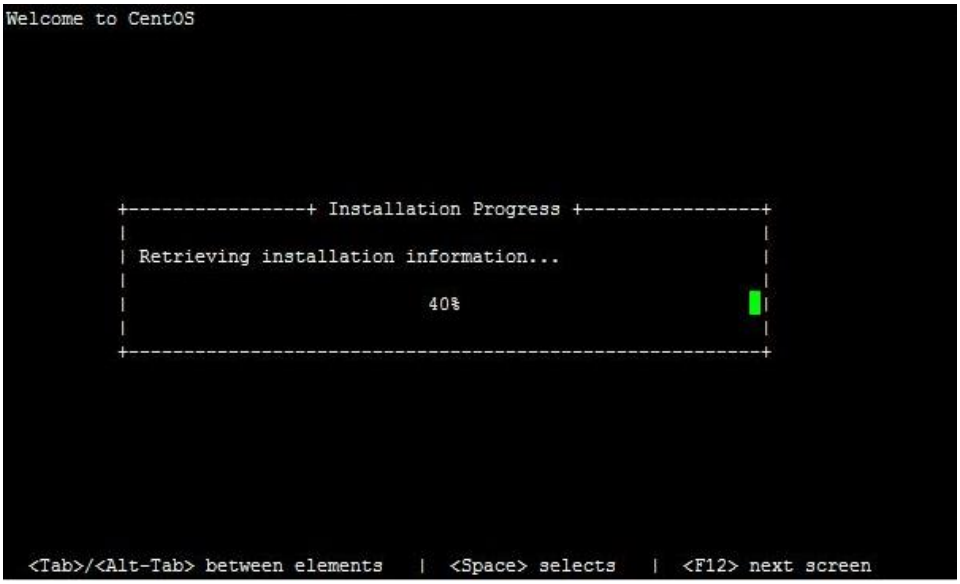

Welcome to CentOS
++ Network Configuration for eth0 ++
Xen Virtual Ethernet
00:16:36:12:4D:C5
[*] Activate on boot
[*] Enable IPv4 support
[ ] Enable IPv6 support
OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

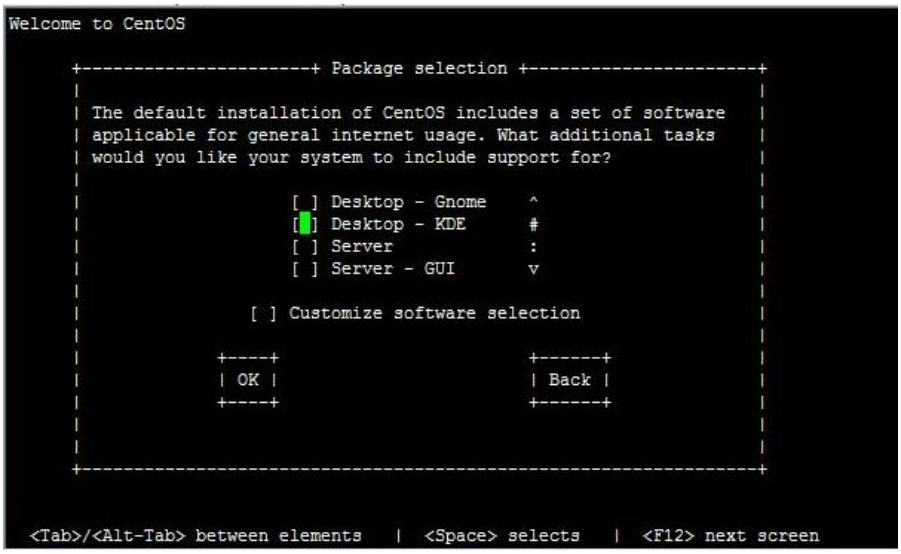

Welcome to CentOS
+-----+ IPv4 Configuration for eth0 +-----+
Xen Virtual Ethernet
00:16:36:12:4D:C5
( ) Dynamic IP configuration (DHCP)
(*) Manual address configuration
IP Address
Prefix (Netmask)
192.168.145.135_ / 255.255.255.0
OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen


Welcome to CentOS
+--------------------+ Hostname Configuration +--------------------+
If your system is part of a larger network where hostnames are
assigned by DHCP, select automatically via DHCP. Otherwise,
select manually and enter a hostname for your system. If you
do not, your system will be known as 'localhost.'
( ) automatically via DHCP
(*) manually
localhost
OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

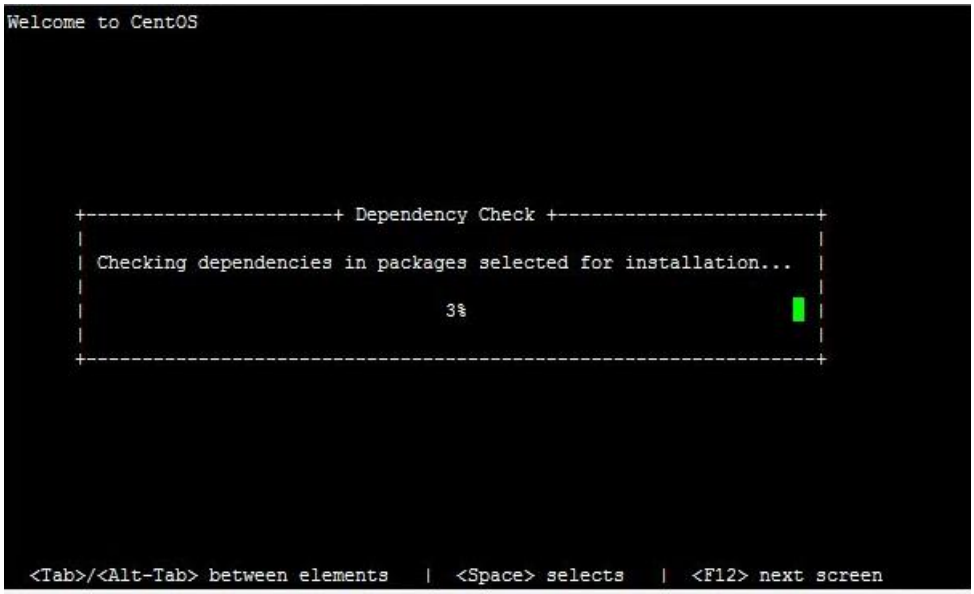

Probing for video card: Unable to probe
No video hardware found, assuming headless
Welcome to CentOS
Time Zone Selection
What time zone are you located in?
[*] System clock uses UTC
Asia/Taipei
Asia/Tashkent
Asia/Tbilisi
Asia/Tehran
Asia/Thimphu
OK
Back

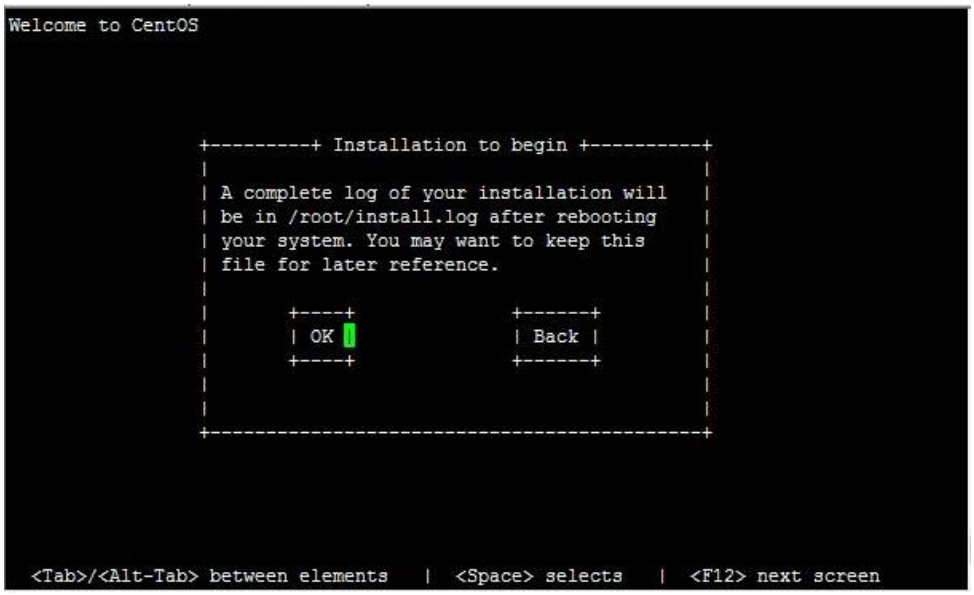

Welcome to CentOS
Root Password
Pick a root password. You must type it twice to ensure you know what it is and didn't make a mistake in typing. Remember that the root password is a critical part of system security!
Password: ******
Password (confirm): ******
OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen


Welcome to CentOS
Installation Progress
Retrieving installation information...
40%
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

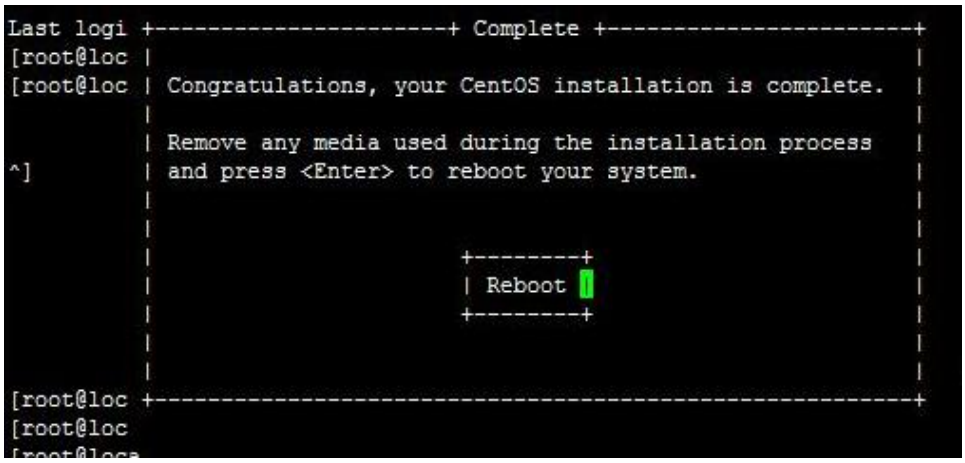

Welcome to CentOS
Package selection
The default installation of CentOS includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?
[ ] Desktop - Gnome
[ ] Desktop - KDE
[ ] Server
[ ] Server - GUI
[ ] Customize software selection
OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen


Welcome to CentOS
Dependency Check
Checking dependencies in packages selected for installation...
3%
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

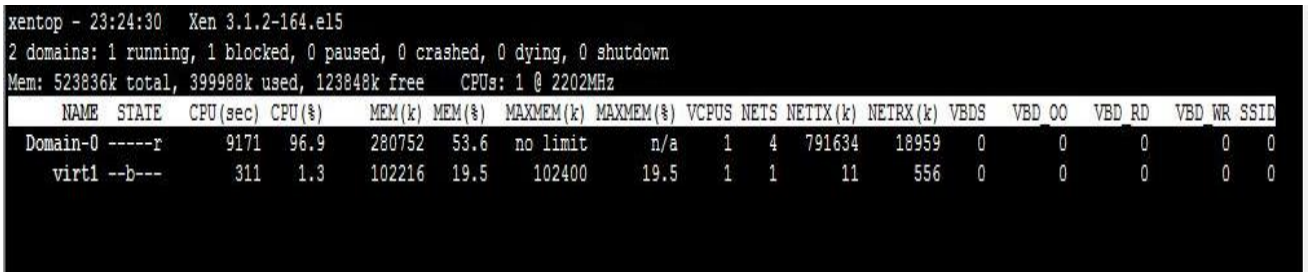

Welcome to CentOS
Dependency Check
Checking dependencies in packages selected for installation...
3%
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen


Welcome to CentOS
+---------+ Installation to begin +----------+
| A complete log of your installation will
| be in /root/install.log after rebooting
| your system. You may want to keep this
| file for later reference.
| OK
| Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen


Welcome to CentOS
+------------------+ Package Installation +------------------+
Name :
Size :
Summary:
+---------+ Install Starting +----------+
| Starting install process. This may
| take several minutes...
Time
Total
Comple
Remaini
0%
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen


Last logi +----------+ Complete +----------+
[root@loc
[root@loc | Congratulations, your CentOS installation is complete.
| Remove any media used during the installation process
^] | and press <Enter> to reboot your system.
| Reboot
[root@loc
[root@loc

Приложение 7



Приложение 8

```
# xm info | grep caps
hw_caps                :
178bfbff:ebd3fbff:00000000:00000010:00002001:00000000:0000
001f
xen_caps               : xen-3.0-x86_32p hvm-3.0-x86_32
hvm-3.0-x86_32p
```

Приложение 9

```
kernel= "/usr/lib/xen/boot/hvmloader"
builder='hvm'
memory = 512
name = "vmname"
vcpus=1
pae=0
acpi=0
apic=0
vif = [ 'bridge=xenbr0' ]
disk               =               [
"file:/virt/windows.img,hda,w",
'file:/iso/windows.iso,hdc:cdrom,r' ]
on_poweroff = 'destroy'
on_reboot   = 'destroy'
on_crash    = 'destroy'
device_model = '/usr/lib/xen/bin/qemu-
dm'
boot='d'
sdl=0
vnc=1
vncconsole=1
stdvga=0
serial='pty'
```

Приложение 10

```
kernel= "/usr/lib/xen/boot/hvmloader"
builder='hvm'
memory = 512
name = "vmname"
vcpus=1
pae=0
acpi=0
apic=0
vif = [ 'bridge=xenbr0' ]
disk = [
"file:/virt/windows.img,hda,w",
'file:/iso/windows.iso,hdc:cdrom,r' ]
on_poweroff = 'destroy'
on_reboot   = 'destroy'
on_crash    = 'destroy'
device_model =
'/usr/lib/xen/bin/qemu-dm'
boot='c'
sdl=0
vnc=1
vncconsole=1
stdvga=0
serial='pty'
```

Приложение 11

```
[root@localhost ~]# virt-install --connect qemu:///system --arch=x86_64 \
-n VM_windows7 -r 2048 \
--vcpus=2 --disk pool=vm_images_dir,size=150,bus=virtio,cache=none \
-c /home/iso/7601.17514.101119-1850_x64fre_server_en-us VL-GRMSXVOL_EN_DVD.iso \
--graphics vnc,listen=0.0.0.0,keymap=ru,password=123456 \
--noautoconsole --os-type windows --os-variant win2k8 \
--network=bridge:br0,model=e1000 \
--disk path=/home/iso/virtio-win-0.1-81.iso,device=cdrom,perms=ro
```

Приложение 12

```
# virt-install -n CentOS --ram 2048 \
--arch=x86_64 \
--vcpus=1 --cpu host --check-cpu \
--extra-args="vnc sshd=1 sshpw=secret ip=static reboot=b selinux=0" \
--os-type linux --os-variant=rhel6 --boot cdrom,hd,menu=on \
--disk pool=vm_images_dir,size=50,bus=virtio \
--network=bridge:br0,model=virtio \
--graphics vnc,listen=0.0.0.0,keymap=ru,password=LinuxPassword \
--noautoconsole --watchdog default,action=reset --virt-type=kvm \
--autostart --location /media/CentOS-6.4-x86_64-bin-DVD1.iso
```

Приложение 13

```
# virt-install -n CentOS --ram 2048 \
--arch=x86_64 \
--vcpus=1 --cpu host --check-cpu \
--extra-args="vnc sshd=1 sshpw=secret ip=static reboot=b selinux=0" \
--os-type linux --os-variant=rhel6 --boot cdrom,hd,menu=on \
--disk pool=vm_images_dir,size=50,bus=virtio \
--network=bridge:br0,model=virtio \
--graphics vnc,listen=0.0.0.0,keymap=ru,password=LinuxPassword \
--noautoconsole --watchdog default,action=reset --virt-type=qemu \
--autostart --location /media/CentOS-6.4-x86_64-bin-DVD1.iso
```

Приложение 14

```
ddns-update-style interim;
ignore client-updates;
authoritative;
# Описание настроек подсети
subnet 192.168.0.0 netmask 255.255.255.0 {
# Диапазон адресов, который будет выдавать DHCP
сервер
range 192.168.0.50 192.168.0.254;
# шлюз по умолчанию
option routers 192.168.0.1;
# маска подсети
option subnet-mask 255.255.255.0;
# DNS
option domain-name-servers 192.168.0.1;
#Время на которое выдаться IP  адрес
default-lease-time 14400;
max-lease-time 86400;
# Путь до образа Thinstation Linux
filename "/visteh-0.1/pxelinux.0";
# TFTP-сервер
next-server 192.168.0.1;
                                }
```

Приложение 16

```
udp        0      0 0.0.0.0:69              0.0.0.0:*
3105/xinetd
```

Приложение 17

```
# IP адрес терминального сервера
 SESSION_0_FREERDP_SERVER=192.168.0.2
 # Протокол доступа к терминальному серверу
SESSION_0_TYPE=freerdp
```

Приложение 18

```
#!/bin/sh
# Add the following line to ensure you always have an
xterm available.
( while true ; do xterm ; done ) &
# Uncomment the following two lines for normal desktop:
unset SESSION_MANAGER
exec /etc/X11/xinit/xinitrc
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid grey
vncconfig -iconic &
xterm  -geometry  80x24+10+10  -ls  -title  "$VNCDESKTOP
Desktop" &
twm & SESSION_0_TYPE=freerdp
```

Приложение 20

```
#!/bin/sh
if [ $? -eq 0 ]; then
  gnome-session
  exit 0
fi
```

Приложение 21

```
#!/bin/sh
# xrdp control script
# Written : 1-13-2006 - Mark Balliet - posicat@pobox.com
# maintaned by Jay Sorg
# chkconfig: 2345 11 89
# description: starts xrdp

XRDP=xrdp
SESMAN=sesman
STARTWM=startwm.sh
```

```
XRDP_DIR=/usr/lib/xrdp/
LOG=/dev/null

cd $XRDP_DIR

if ! test -x $XRDP
then
  echo "$XRDP is not executable"
  exit 0
fi
if ! test -x $SESMAN
then
  echo "$SESMAN is not executable"
  exit 0
fi
if ! test -x $STARTWM
then
  echo "$STARTWM is not executable"
  exit 0
fi

xrdp_start()
{
  echo -n "Starting: xrdp and sesman . . "
  ./$XRDP >> $LOG
  ./$SESMAN >> $LOG
  echo "."
  sleep 1
  return 0;
}

xrdp_stop()
{
  echo -n "Stopping: xrdp and sesman . . "
  ./$SESMAN --kill >> $LOG
  ./$XRDP --kill >> $LOG
  echo "."
  return 0;
}

is_xrdp_running()
{
  ps u --noheading -C $XRDP | grep -q -i $XRDP
  if test $? -eq 0
  then
    return 1;
  else
    return 0;
```

```
  fi
}

is_sesman_running()
{
  ps u --noheading -C $SESMAN | grep -q -i $SESMAN
  if test $? -eq 0
  then
    return 1;
  else
    return 0;
  fi
}

check_up()
{
  # Cleanup : If sesman isn't running, but the pid exists, erase
it.
  is_sesman_running
  if test $? -eq 0
  then
    if test -e /var/run/sesman.pid
    then
      rm /var/run/sesman.pid
    fi
  fi
  # Cleanup : If xrdp isn't running, but the pid exists, erase
it.
  is_xrdp_running
  if test $? -eq 0
  then
    if test -e /var/run/xrdp.pid
    then
      rm /var/run/xrdp.pid
    fi
  fi
  return 0;
}

case "$1" in
  start)
    check_up
    is_xrdp_running
    if ! test $? -eq 0
    then
      echo "xrdp is already loaded"
      exit 1
    fi
```

```
    is_sesman_running
    if ! test $? -eq 0
    then
      echo "sesman is already loaded"
      exit 1
    fi
    xrdp_start
    ;;
  stop)
    check_up
    is_xrdp_running
    if test $? -eq 0
    then
      echo "xrdp is not loaded."
    fi
    is_sesman_running
    if test $? -eq 0
    then
      echo "sesman is not loaded."
    fi
    xrdp_stop
    ;;
  force-reload|restart)
    check_up
    echo "Restarting xrdp ..."
    xrdp_stop
    is_xrdp_running
    while ! test $? -eq 0
    do
      check_up
      sleep 1
      is_xrdp_running
    done
    xrdp_start
    ;;
  *)
    echo  "Usage:  xrdp_control.sh  {start|stop|restart|force-
reload}"
    exit 1
esac

exit 0
```

Приложение 22

Терминология видов виртуализации			**Иностранное обозначение**	
Аппаратная виртуализация	(Пара)виртуализация с поддержкой аппаратного обеспечения			
Программная виртуализация	Полная эмуляция	Эмуляция аппаратуры	Full	Native Virtualization
Паравиртуа-лизация	Частичная эмуляция		paravirtua-lization	
Виртуализация на уровне ядра ОС	Виртуализация уровня операционной системы	контейнеры	operating system-level virtualization	

Приложение 23

-- **nographic**	Так отмечается в каком режиме нужно провести процесс инсталляции. Установка будет производиться в консоли без графики.
–prompt	Создание Виртуальной машины произойдёт в интерактивном режиме. Необходимо будет ответить на следующие вопросы.
"What is the name of your virtual machine?"	Указывается имя виртуальной машины
"How much RAM should be allocated (in megabytes)?"	Указывается объём выделяемой виртуальной машине оперативной памяти
"What would you like to use as the disk (file path)?"	Указывается путь к файлу-образу будущей виртуальной машины
"How large would you like the disk (…) to be (in gigabytes	Указывается объём памяти для виртуальной машины в гигабайтах
"What is the install URL?"	Указывается откуда будет устанавливаться ОС.

kernel	VMX firmware loader, для HVM-домена обычно hvmloader
builder	Тип домена. Для HVM-домена обязательно hvm
acpi	Поддержка ACPI внутри HVM-домена, по умолчанию равно "0" (отключено)
apic	Поддержка APIC внутри HVM-домена, по умолчанию равно "0" (отключено)
pae	Поддержка PAE внутри HVM-домена, по умолчанию равно "0" (отключено)
vif	Описание сетевых интерфейсов. Представляет собой список строк, каждая из которых описывает один интерфейс. В каждой строке, соответствующей интерфейсу, обязательно должен присутствовать компонент bridge, указывающий к какому мосту подключён интерфейс. Опционально может указываться MAC-адрес интерфейса (по умолчанию генерируется случайным образом внутри диапазона, выделенного Xen Source). Здесь же может задаваться модель эмулируемой сетевой карты (параметр model).
disk	Определяет дисковые устройства, к которым гостевой домен должен иметь доступ. Если для домена используется физический носитель в качестве диска, то он должен быть описан строкой типа: phy:UNAME,DEV,MODE, где **UNAME** - имя устройства, **DEV** - имя диска, как его видит домен и **MODE** принимает значения **r** для read-only и **w** для read-write. Если используется образ диска, находящийся в файле, то строка принимает вид: file:FILEPATH,DEV,MODE Если используется больше одного диска, то они разделяются запятой. Например: disk = ['file:/var/images/image1.img,hda,w', 'file:/var/images/image2.img,hdb,w'] Если какой-то из образов является образом компакт-диска, и предполагается, что в виртуальной машине должен эмулироваться привод CD-ROM, нужно использовать суффикс :cdrom в описании диска: disk = ['file:/var/images/image1.img,hda,w', 'file:/var/images/image2.iso,hdc:cdrom,w'] CD-ROM'ов может быть несколько.
Boot	Загрузка с floppy (a), hard disk (c) или CD-ROM (d).

device_model	Инструмент эмуляции устройств для HVM-домена. Обычно qemu-dm
sdl	Задействует библиотеку SDL для отображения графики, по умолчанию равно "0" (отключено)
vnc	Задействует библиотеку VNC для отображения графики, по умолчанию равно "0" (отключено) Пользователь может использовать vncviewer для подключения к домену. Например: $ vncviewer domain0_IP_address:0
vncconsole	Нужно ли автоматически запускать **vncviewer** при старте домена. Имеет смысла только если vnc=1. По умолчанию равно 0.
Serial	Перенаправление последовательных портов гостевого домена на файл устройства в домене 0.
usb	Включение поддержки USB без указания специфического устройства. По умолчанию эта функция отключена, в случае же определения параметра usbdevice, ее необходимо задействовать.
usbdevice	Включение поддержки конкретных устройств. Например, поддержка мыши PS/2 через USB: usbdevice='mouse'
Localtime	Установка локального времени. По умолчанию равно "0", т.е UTC
Soundhw	Тип звукового устройства для эмуляции. Обычно sb16
full-screen	Поддержка полноэкранного режима.
nographic	Не использовать графический интерфейс, работать только через консоль. В этом случае опции 'sdl' или 'vnc' не работают.

Приложение 25

--arch	Сообщается архитектура процессора и битность ОС. Здесь в основном ставят x86_64 - для 64 разрядных ОС или i386 - для 32 разрядных ОС. А так же можно указать arm, то есть risc процессор для Android операционной системы.
-n	Задаётся имя создаваемой машины
-r	Задаёт объём оперативной памяти для гостевой машины.
--vcpus	Задаёт количество ядер ЦП.

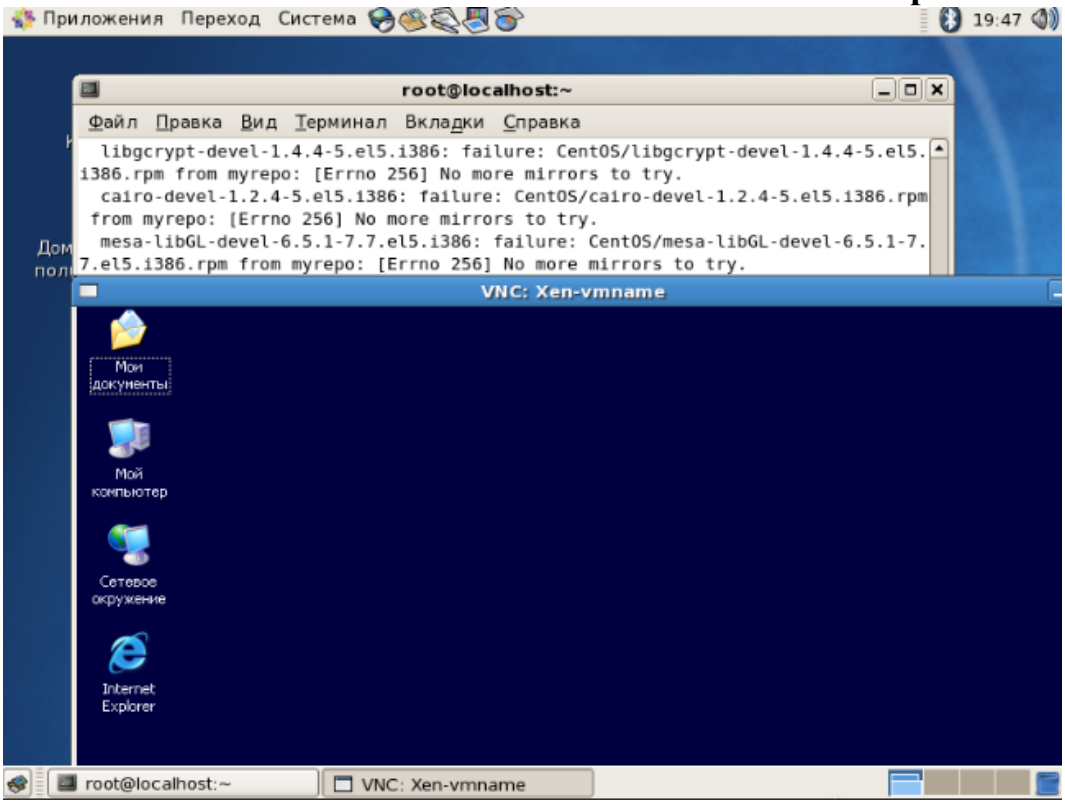
--disk pool	Указывает какое хранилище использовать для хранения машины.
size	Указывает объем дикого пространства (образа) виртуальной машины.
-c	Задаёт инсталляционный образ ОС, который будет загружен при первом старте виртуальной машины.
--graphics	Указывает на то, что необходимо использовать графический доступ к виртуальной машине через VNC Viewer. listen= 0.0.0.0 свидетельствует о том, что соединение к машине может быть инициировано с любого хоста, а password=WinPassword – задаёт пароль доступа к VNC.
--os-type	Задаёт тип устанавливаемой машины.
--os-variant	Указывает точное наименование ОС, которая будет работать и устанавливаться.
--network=bridge:br0	Задаёт какое сетевое соединение будет использовано.
--disk path=/iso/virtio-win-0.1-81.iso	Загружает виртуальный CD ROM с драйверами жёсткого диска.

Приложение 26

Таблица . Сравнительный анализ производительности KVM и XEN по отношению к производительности реального сервера

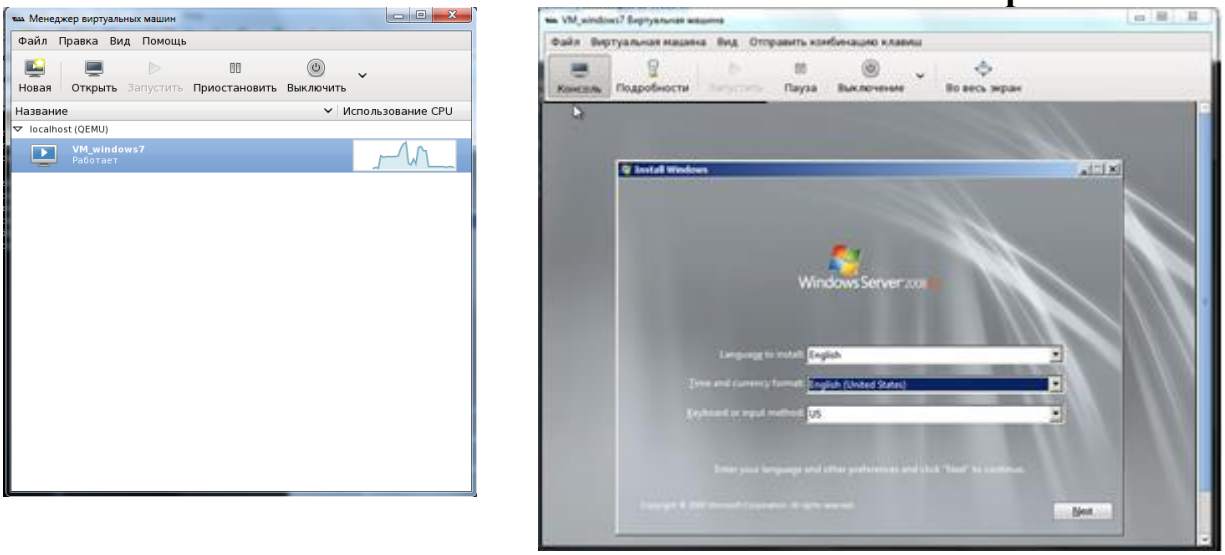
Название теста	Реальный сервер	KVM	XEN	Сравнение в пунктах			Сравнение в процентном отношении				
				KVM - реальная машина	XEN - реальная машина	Отклонение KVM от XEN	KVM / реальная машина	XEN / реальная машина	Отклонение KVM от реальной машины	Отклонение XEN от реальной машины	Отклонение KVM-XEN
C-Ray	35,35	35,66	36,13	0,31	0,78	-0,47	100,88	97,84	0,88	-2,16	3,04
POV-Ray	230,02	232,44	235,89	2,42	5,87	-3,45	101,05	97,51	1,05	-2,49	3,54
Smallpt	160	162	167,5	2	7,5	-5,5	101,25	95,52	1,25	-4,48	5,73
Timed MAFFT Alignment	7,78	7,795	8,42	0,015	0,64	-0,625	100,19	92,40	0,19	-7,60	7,79
John the Ripper (Blowfish)	3026	2991,5	2856	-34,5	-170	135,5	98,86	105,95	-1,14	5,95	-7,09
John the Ripper (DES)	7374833,5	7271833,5	6911167	-103000	-463666,5	360666,5	98,60	106,71	-1,40	6,71	-8,11
John the Ripper (MD5)	49548	48899,5	46653,5	-648,5	-2894,5	2246	98,69	106,20	-1,31	6,20	-7,51
OpenSSL	397,68	393,95	388,25	-3,73	-9,43	5,7	99,06	102,43	-0,94	2,43	-3,37
7-Zip	12467	12129,5	11879	-337,5	-588	250,5	97,29	104,95	-2,71	4,95	-7,66
CLOMP	3,3	3,285	3,125	-0,015	-0,175	0,16	99,55	105,60	-0,45	5,60	-6,05
PostMark	3667	3824	3205	157	-462	619	104,28	114,41	4,28	14,41	-10,13

Приложение 27



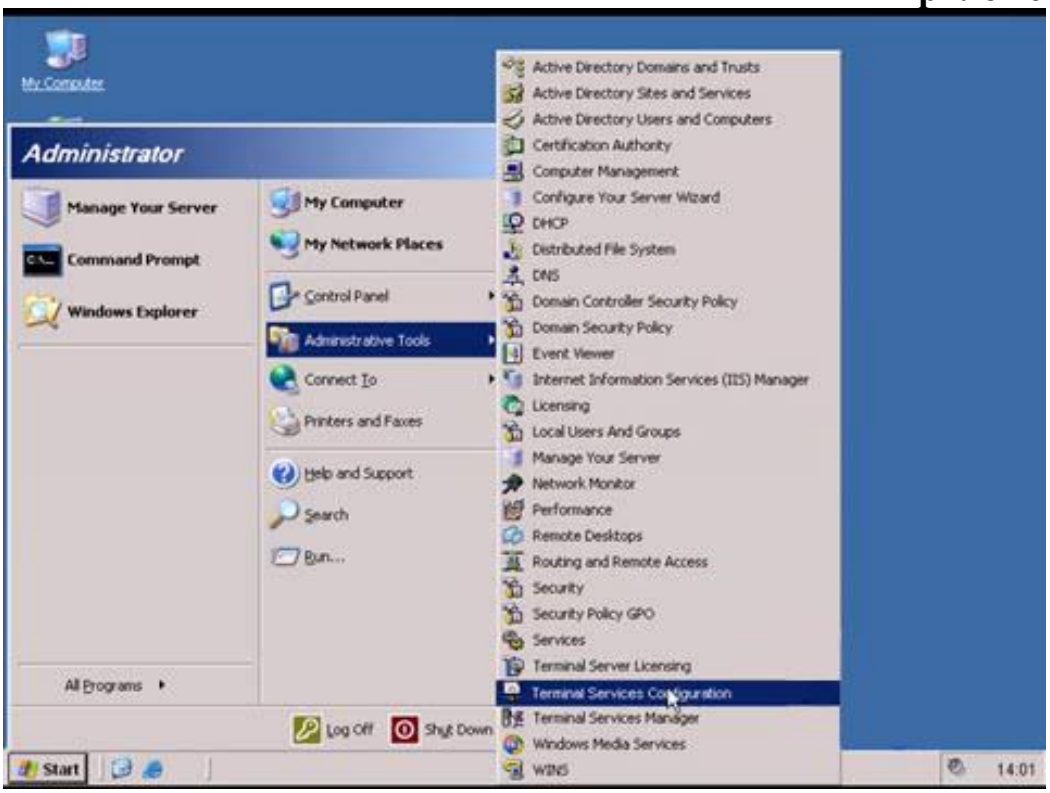
Запущенная виртуальная машина с ОС WINDOWS

Приложение 28

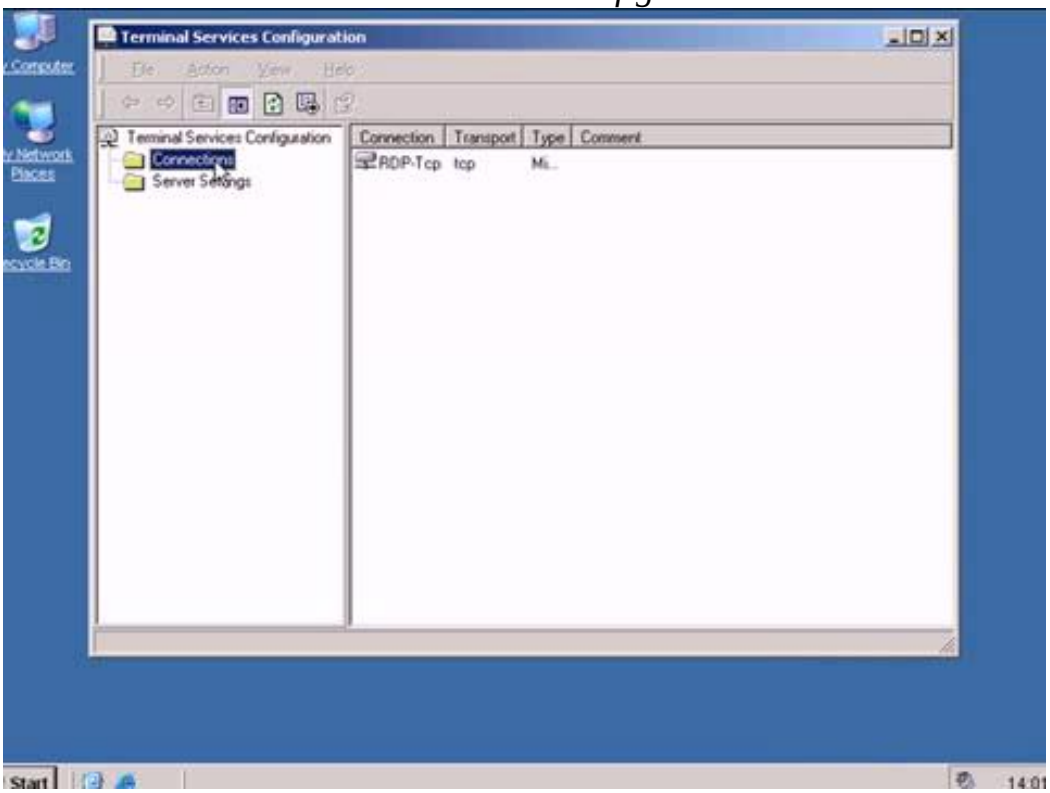


Управление гостевой машиной

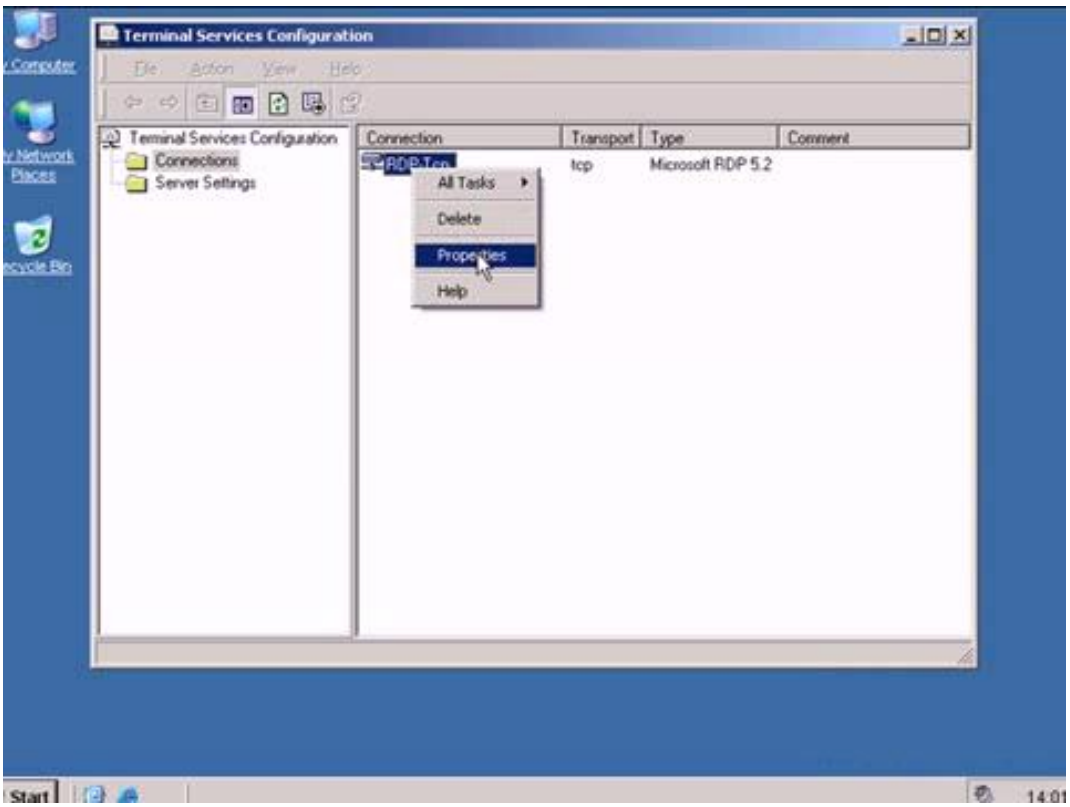
Приложение 29



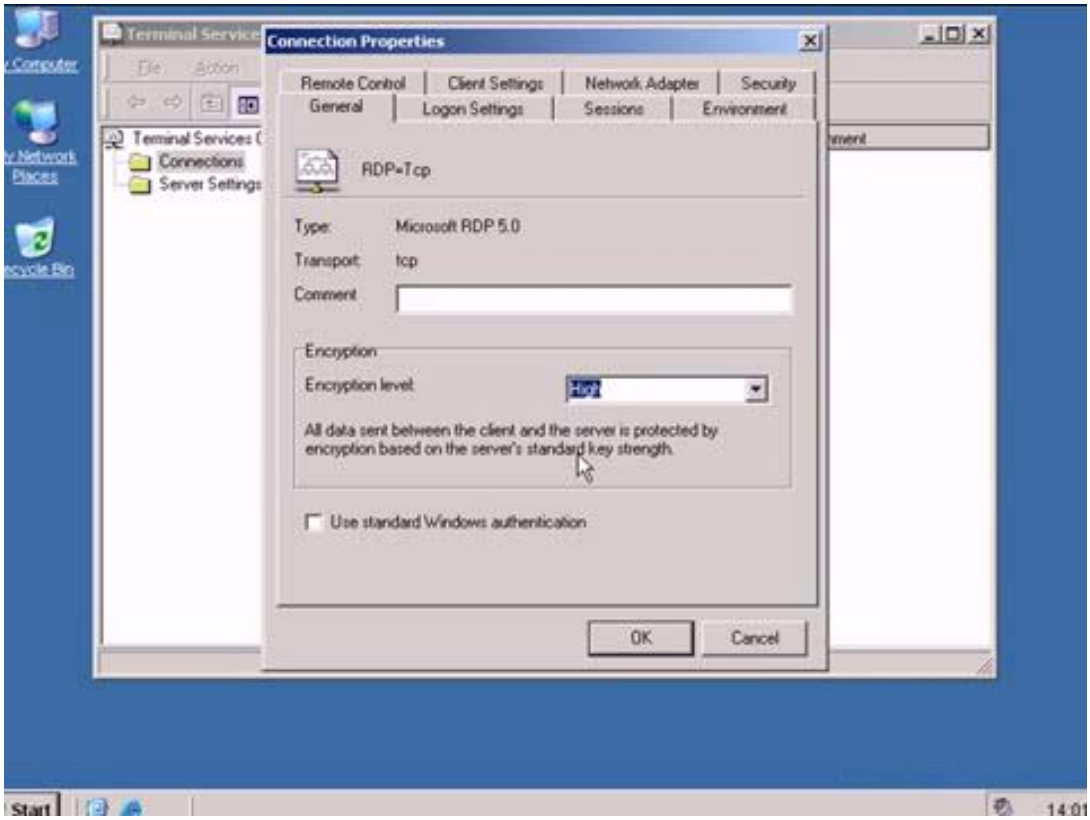
Terminal Service Configuration



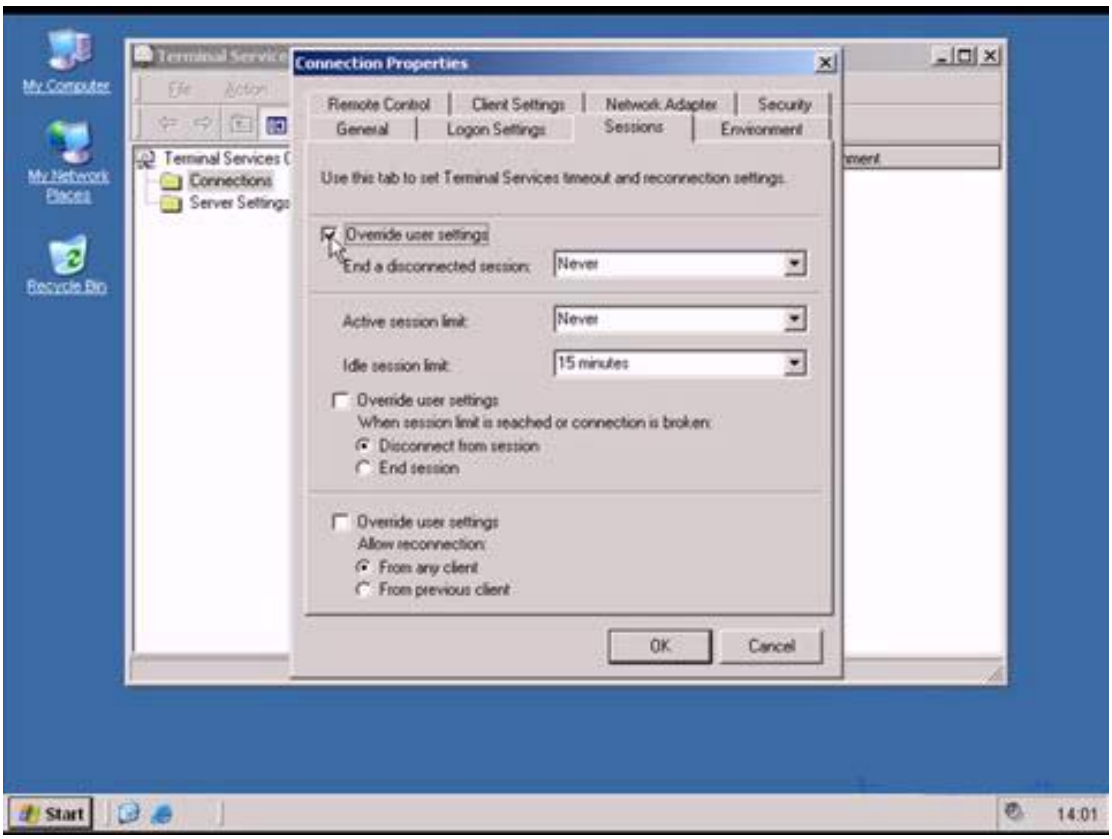
Папка «Corrections»



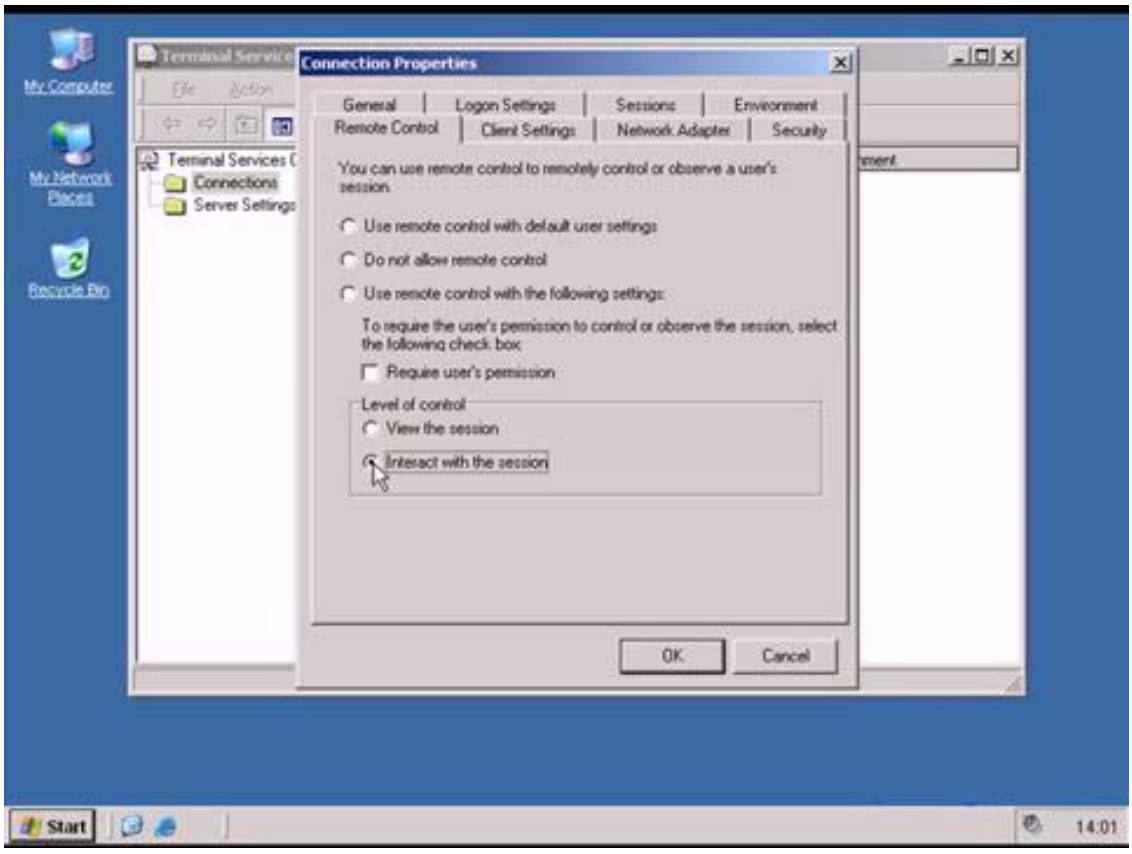
Свойства RDP-TCP



. Вкладка «General»



Вкладка «Sessions»



Вкладка «Remote Control»