Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования «Севастопольский государственный университет»

Институт информационных технологий и управления в технических системах
(полное название института)

Кафедра информационных технологий и компьютерных систем
(полное название кафедры)

Пояснительная записка

к выпускной работе бакалавра
(выпускной квалификационной работе, дипломному проекту (работе))

на тему: Разработка виртуальной инфраструктуры для реализации облачных услуг

Выполнил: студент 4 курса, группы ВТб-41д

направления подготовки (специальности) 09.03.01 – информатика и вычислительная техника
(шифр и название направления подготовки (специальности))

направленность/профиль/специализация 09.03.01.01 «ЭВМ, системы и сети»

Умеров Амет Ремзиевич
(фамилия, имя, отчество студента)

Руководитель Мащенко Е.Н., доцент
(фамилия, инициалы, ученая степень, звание, должность)

Дата допуска к защите «____»_____ 2015 г.

Зав. кафедрой _____ Брюховецкий А.А.
(подпись) (инициалы, фамилия)

2015 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Севастопольский государственный университет»

Институт информационных технологий и управления в технических системах
Кафедра информационных технологий и компьютерных систем
Направление подготовки (специальность) 09.03.01 «Информатика и вычислительная техника»
(код и название)
Направленность/профиль/специализация 09.03.01.01 «ЭВМ, системы и сети»

УТВЕРЖДАЮ

Заведующий кафедрой ИтиКС
А.А. Брюховецкий

“26” марта 2015 года

З А Д А Н И Е

на выпускную квалификационную работу бакалавра
(указать форму в соответствии с ФГОС, при наличии)
студенту Умерову Амету Ремзиевичу

1. Тема работы (проекта) Разработка виртуальной инфраструктуры для реализации облачных услуг
руководитель работы (проекта) Мащенко Елена Николаевна, канд.техн.наук, доцент
(фамилия, имя, отчество, степень, звание, должность)
Утверждены приказом высшего учебного заведения от “01” апреля 2015 года № 129-П

2. Срок подачи студентом работы (проекта) 08.06.2015 г.

3. Входные данные к работе (проекту) параметры клиентских контейнеров, параметры администрирования виртуальной инфраструктуры, параметры системы мониторинга и резервного копирования, перечень оказываемых облачных услуг; критерии качества обслуживания клиентов. Технологии виртуализации: OpenVZ, KVM. Критерии эффективности виртуальной инфраструктуры по производительности и надежности определены в соглашении об уровне обслуживания облачного провайдера.

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые нужно разработать) Введение. 1. Постановка задачи. 2. Обзор современных методов и технологий серверной виртуализации. 3. Системный анализ виртуальной инфраструктуры 4. Описание виртуальной инфраструктуры. 5. Руководство администратора. 6. Руководство пользователя 7. Результаты тестирования. 8. Безопасность жизнедеятельности. Заключение. Библиографический список. Приложения.

5. Перечень графического материала (с точным указанием обязательных чертежей)
1.Постановка задачи (1 плакат)
2. Структурная схема виртуальной инфраструктуры (1 лист)
3. Схемы алгоритмов функционирования виртуальной инфраструктуры (2 листа)
4. Результаты тестирования виртуальной инфраструктуры (1 плакат)

6. Консультанты разделов работы (проекта)

Раздел	Фамилия, инициалы и должность консультанта	Подпись, дата	
		задание выдал	задание принял
Безопасность жизнедеятельности	Доц. Азаренко Е.И.		
Нормоконтроль	Доц. Корепанова Н.Л.		

7. Дата выдачи задания 26.03.2015 г.

КАЛЕНДАРНЫЙ ПЛАН

№ п/п	Название этапов работы (проекта)	Срок выполнения этапов работы (проекта)	Примечание
1	Анализ постановки задачи, системный анализ	30.03.2015 – 26.04.2015	
2	Разработка алгоритмов функционирования системы и основных подсистем	27.04.2015 – 10.05.2015	
3	Разработка виртуальной инфраструктуры	11.05.2015 – 21.05.2015	
4	Испытание виртуальной инфраструктуры	22.05.2015 – 31.05.2015	
5	Оформление пояснительной записки и чертежей	01.06.2015 – 07.06.2015	
6	Представление работы на кафедру	08.06.2015	
7	Защита работы в ГЭК	17.06.2015	

Студент _____ Умеров А.Р.
(подпись) (фамилия и инициалы)

Руководитель работы (проекта) _____ Мащенко Е.Н.
(подпись) (фамилия и инициалы)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Облачные услуги — это способ предоставления, потребления и управления технологией. Данный тип услуг выводит гибкость и эффективность на новый уровень, путем эволюции способов управления, таких как непрерывность, безопасность, резервирование и самообслуживание, которые соединяют физическую и виртуальную среду. В данной сфере возрастает потребность в качественно продуманной архитектуре, позволяющей надежно и правильно организовать облачную инфраструктуру.

Для эффективной работы облачной инфраструктуры требуется эффективная структура и организация. Небольшая команда из специалистов и бизнес-пользователей может создать обоснованный план и организовать свою работу в инфраструктуре. Данная выделенная группа может намного эффективнее построить и управлять нестандартной облачной инфраструктурой, чем если компании будут просто продолжать добавлять дополнительные сервера и сервисы для поддержки центра обработки данных (ЦОД).

IaaS (Infrastructure as a Service) — это предоставление пользователю компьютерной и сетевой инфраструктуры и их обслуживание как услуги в форме виртуализации, то есть виртуальной инфраструктуры. Другими словами, на базе физической инфраструктуры дата-центров (ДЦ) провайдер создает виртуальную инфраструктуру, которую предоставляет пользователям как сервис. Стоит отметить, что IaaS не предполагает передачи в аренду программного обеспечения, а всего лишь предоставляет доступ к вычислительным мощностям.

Технология виртуализации ресурсов позволяет физическое оборудование (сервера, хранилища данных, сети передачи данных) разделить между пользователями на несколько частей, которые используются ими для выполнения текущих задач. К примеру, на одном физическом сервере можно запустить сотни виртуальных серверов, а пользователю для решения задач выделить время доступа к ним.

1 ПОСТАНОВКА ЗАДАЧИ

Конечная цель проектирования — разработка виртуальной инфраструктуры для реализации облачных услуг.

Виртуальная инфраструктура должна обладать следующими характеристиками:

- использование, по возможности, продуктов, распространяющихся под свободной лицензией (GNU GPL) для организации инфраструктуры;
- устранение единой точки отказа при проектировании инфраструктуры;
- защита от распределенных атак на отказ (DDoS);
- использование инфраструктуры в бизнесе для предоставления облачных услуг клиентам.

Аппаратное обеспечение должно базироваться в дата-центре ориентированным на требования стандарта Tier III, состоящего из информационной, телекоммуникационной и инженерной инфраструктуры, с возможностью аренды выделенных серверов и аппаратной защиты от DDoS-атак, а также поддержкой аппаратного RAID. Рекомендуемые характеристики выделенных серверов виртуализации:

- 2xIntel® Xeon E5430 @ 2.66GHz;
- минимальный объем ОЗУ 32 Гб;
- минимум 960 Гб места на жестком диске (SSD);
- поддержка аппаратного RAID;
- операционная система не ниже Debian 7 GNU/Linux или CentOS 6.5;
- интернет-канал с пропускной способностью не менее 100 Мб/с;
- возможность добавления дополнительных IP-адресов к серверам;
- возможность удаленного доступа к серверам посредством IPMI (Intelligent Platform Management Interface).

Следует предусмотреть возможность развертывания дополнительных виртуальных серверов, для организации DNS-серверов, системы мониторинга и платежной системы. Также необходимо предусмотреть наличие

системы хранения данных (СХД) для резервных копий, а также расширение дискового пространства на выделенных серверах.

Для разработки виртуальной инфраструктуры необходимо реализовать следующие этапы:

- выбор технологий виртуализации;
- выбор физических серверов на основе услуги IaaS в ДЦ;
- приобретение лицензий на программное обеспечение;
- приобретение подсетей IP-адресов;
- создание и подключение инфраструктуры мониторинга, платежной системы (биллинга), резервного копирования;
- реализация аппаратной защиты от DDoS-атак;
- выбор перечня PaaS-услуг;
- внедрение перечня предоставляемых PaaS-услуг;
- создание руководства администратора по виртуализации;
- создание руководства для клиентов по часто задаваемым вопросам;
- внедрение инфраструктуры на рынке PaaS-услуг.

2 ОБЗОР СОВРЕМЕННЫХ МЕТОДОВ И ТЕХНОЛОГИЙ СЕРВЕРНОЙ ВИРТУАЛИЗАЦИИ

Термин «облачные вычисления» сегодня уже достаточно хорошо известен и в информационных технологиях (ИТ), и в бизнес-кругах. Почти каждую неделю появляются новые статьи, книги, презентации об облачных вычислениях — новой сервисной модели предоставления вычислительных услуг.

За время существования информационных технологий сменилось несколько моделей построения информационных систем. Все начиналось с монолитной архитектуры (mainframe), когда и база данных, и приложения работали на одном большом компьютере, а пользователи сидели у «тонких» терминалов, которые только отображали информацию. У такой архитектуры было много недостатков, и ее сменила более перспективная архитектура «клиент-сервер». В ней был свой выделенный сервер баз данных (БД) и пользователи на «толстых» клиентах, которые разгружали сервер БД. Затем появилась еще более современная архитектура — многоуровневая (или трехуровневая), где логика приложений была вынесена на отдельный компьютер, называемый сервером приложений, а пользователи работали на «тонких» клиентах через веб-браузеры. Большинство приложений сегодня выполнено именно в этой архитектуре. Она подразумевает развертывание всей ИТ-инфраструктуры на территории заказчика [1].

Облачные вычисления — это следующий шаг в эволюции архитектуры построения информационных систем. Благодаря огромным преимуществам этого подхода очевидно, что многие информационные системы в ближайшее время будут перенесены в облако. Этот процесс уже начался и его игнорирование или недооценка может привести к поражению в конкурентной борьбе на рынке. Имеется ввиду не только отставание ИТ, или неоправданные затраты на него, но и отставание в развитии основного бизнеса

компании, зависящего от гибкости ИТ-инфраструктуры и скорости вывода новых сервисов и продуктов на рынок.

ИТ-директор американского правительства Вивек Кундра, в феврале 2011 года опубликовал стратегию стратегию переноса части информационных систем в облако. Документ под названием «Federal Cloud Computing Strategy» четко описывает порядок и сроки переноса. Цель работ — уменьшение сложности и повышение управляемости ИТ, увеличение нагрузки оборудования до 70-80%, уменьшение количества центров обработки данных.

Основным требованием, предъявляемым к центрам обработки данных является отказоустойчивость. При этом подразумевается отключение ЦОД как на время планово-предупредительных работ и профилактики оборудования, так и внеплановых аварийных ситуаций.

Классификация Tier описывает надежность функционирования ЦОД и является необходимой для компаний, как желающих построить свой ЦОД, так и для арендующих чужие вычислительные мощности. В зависимости от критичности бизнеса, в зависимости от потерь, которые понесет компания в случае остановки бизнес-процессов выбирается тот или иной уровень надежности. В свою очередь, высокий уровень надежности требует высоких материальных и эксплуатационных затрат, поэтому и стоимость вычислительных мощностей зависит от уровня надежности ЦОД [2].

На сегодняшний день существует четыре уровня надежности ЦОД названные Tier I, Tier II, Tier III и Tier IV, которые были введены организацией Uptime Institute (Институт бесперебойных процессов, США):

- Tier I: время простоя 28.8 часов в год, коэффициент отказоустойчивости 99.671%;
- Tier II: 22.0 часа в год, 99.749%;
- Tier III: 1.6 часа в год, 99.982%;
- Tier IV: 0.4 часа в год, 99.995%.

ЦОД уровня Tier I (базовый уровень) подвержен нарушениями работы как от плановых, так и от внеплановых действий. Применение фальшпола, источников бесперебойного питания (ИБП), дизель-генераторных установок

(ДГУ) не обязательно. Если ИБП и ДГУ используются, то выбираются более простые модели, без резерва, с множеством точек отказа. Возможны самопроизвольные отказы оборудования. К простою ЦОД также приведут ошибки в действиях обслуживающего персонала. В таких ЦОД отсутствует защита от случайных и намеренных событий, обусловленных действиями человека.

В ЦОД уровня Tier II (с резервированными компонентами) время простоя возможно в связи с плановыми и внеплановыми работами, а также аварийными ситуациями, однако оно сокращено благодаря внедрению одной резервной единицы оборудования в каждой системе. Таким образом, системы кондиционирования, ИБП и ДГУ имеют одну резервную единицу, тем не менее, профилактические работы требуют отключения ЦОД. В центрах обработки данных с резервированными компонентами требуется наличие минимальных защитных мер от влияния человека.

Третий уровень надежности (уровень с возможность параллельного проведения ремонтных работ) требует осуществления любой плановой деятельности без остановки ЦОД. Под плановыми работами подразумевается профилактическое и программируемое техническое обслуживание, ремонт и замена компонентов, добавления или удаление компонентов, а также их тестирование. В таком случае необходимо иметь резерв, благодаря которому можно пустить всю нагрузку по другому пути, во время работ на первом. Для реализации Tier III необходима схема резервирования блоков схем кондиционирования, ИБП, ДГУ N+1, также требуется наличие двух комплектов трубопроводов для системы кондиционирования, построенной на основе чиллера (холодильной машины). Строительные требования обязывают сохранять работоспособность ЦОД при большинстве случаев намеренных и случайных вмешательств человека. Также следует предусмотреть резервные входы, дублирующие подъездные пути, контроль доступа, отсутствие окон, защиту от электромагнитного изучения.

Четвертый уровень надежности ЦОД (отказоустойчивый) характеризуется безостановочной работой при проведении плановых мероприятий и

способен выдержать один серьезный отказ без последствий для критически важной нагрузки. Необходим дублированный подвод питания, резервирование системы кондиционирования и ИБП по схеме 2(N+1). Для дизель-генераторных установок необходима отдельная площадка с зоной хранения топлива. Tier IV требует защиту от всех потенциальных проблем в связи с человеческим фактором. Регламентированы избыточные средства защиты от намеренных или случайных действий человека. Учтено влияние сейсмоявлений, потопов, пожаров, ураганов, штормов, терроризма.

Дата-центры по виду использования подразделяют на корпоративные и коммерческие (аутсорсинговые). Корпоративные ДЦ предназначены для обслуживания конкретной компании, коммерческие, в свою очередь, предоставляют услуги всем желающим.

Некоторые ДЦ предлагают клиентам дополнительные услуги по использованию оборудования, по автоматическому уходу от различных видов атак. Команда специалистов круглосуточно производит мониторинг серверов. Для обеспечения сохранности данных используются системы резервного копирования. Для предотвращения кражи данных, в дата-центрах используются различные системы ограничения физического доступа, системы видеонаблюдения.

Дата-центры предоставляют несколько основных типов услуг, среди которых:

- виртуальный хостинг (shared hosting);
- виртуальный сервер (virtual private/dedicated server);
- выделенный сервер (dedicated server);
- размещение сервера (colocation);
- выделенная зона (dedicated area).

Виртуальный хостинг используется для размещения большого количества сайтов на одном веб-сервере. В основном для построения веб-сервра используется типичный стек технологий LAMP, где в качестве операционной системы выступает GNU/Linux, http-сервер Apache (зачастую в связке с nginx), сервер баз данных MySQL, интерпретируемые скриптовые языки

PHP, Perl, Python. Существует решение на базе ОС Windows Server, где в качестве http-сервера используется IIS, в качестве СУБД выступает MS SQL, а также существует поддержка платформы ASP.NET. Разделение ресурсов на виртуальном хостинге основывается на ограничении дискового пространства, сетевого трафика, количества используемых доменов, почтовых ящиков, баз данных, FTP-аккаунтов, ограничение на использование процессорного времени, памяти для PHP-скриптов и так далее.

Виртуальный выделенный сервер эмулирует работу отдельного физического сервера. На одной физической машине может быть запущено несколько виртуальных серверов, при этом каждый виртуальный сервер имеет свои процессы, ресурсы и отдельное администрирование. Для реализации виртуальных машин используются технологии виртуализации, как системы с открытым исходным кодом, так и коммерческие.

В случае выделенного сервера, клиенту целиком предоставляется отдельная физическая машина. Владелец сервера имеет возможность смены конфигурации оборудования, установки любой операционной системы. Такой тип хостинга подходит для высоконагруженных проектов.

Размещение сервера отличается от услуги предоставления выделенного сервера тем, что ДЦ размещает у себя сервер, который заранее подготовил клиент. Дата-центр подключает его в общую инфраструктуру ЦОДа, обеспечивает бесперебойное электропитание, охлаждение, доступ к сетевому каналу, удаленный доступ к серверу, охрану, мониторинг и другие услуги.

Выделенная зона предоставляется в основном для специальных клиентов, имеющих строгие нормы безопасности. В этом случае дата-центр предоставляет выделенную зону, обеспеченную электроснабжением, холодоснабжением и системами безопасности, а клиент сам создает свой дата-центр внутри этого пространства.

Также можно выделить такую услугу, как аренда телекоммуникационных стоек, которая является частным случаем размещения сервера, с отличием в том, что арендаторами в основном являются юридические лица.

При построении облачной инфраструктуры важную роль играет виртуализация.

Виртуализация — абстракция вычислительных ресурсов и предоставление пользователю системы, которая инкапсулирует (скрывает в себе) собственную реализацию. Проще говоря, пользователь работает с удобным для себя представлением объекта, и для него не имеет значения, как объект устроен в действительности. Термин «виртуализация» появился в шестидесятых годах XX века, а в девяностых — стали очевидны перспективы подхода: с ростом аппаратных мощностей, как персональных компьютеров, так и серверных решений, вскоре представится возможность использовать несколько виртуальных машин на одной физической платформе.

Понятие виртуализации можно условно разделить на две категории [3]:

- виртуализация платформ, продуктом этого вида виртуализации являются виртуальные машины — некие программные абстракции, запускаемые на платформе реально аппаратно-программных систем;
- виртуализация ресурсов преследует целью комбинирование или упрощение представления аппаратных ресурсов для пользователя и получение неких пользовательских абстракций оборудования, пространств имен, сетей.

Когда производится виртуализация, существует множество способов ее осуществления. Фактически есть несколько путей, с помощью которых достигаются одинаковые результаты через разные уровни абстракции [4]:

- эмуляция аппаратных средств;
- полная виртуализация;
- паравиртуализация;
- виртуализация уровня ОС.

Эмуляция аппаратных средств является одним из самых сложных методов виртуализации. В то же время, главной проблемой при эмуляции аппаратных средств является низкая скорость работы, в связи с тем, что каждая команда моделируется на основных аппаратных средствах. В эмуляции оборудования используется механизм динамической трансляции, то

есть каждая из инструкций эмулируемой платформы заменяется на заранее подготовленный фрагмент инструкций физического процессора [5]. Архитектура процесса эмуляции представлена на рисунке 2.1.

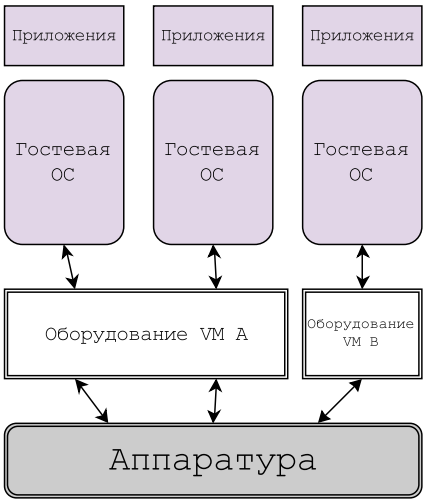

Рисунок 2.1 – Эмуляция аппаратных средств

Примерами виртуализации посредством эмуляции являются программные платформы QEMU и Bochs.

Система QEMU поддерживает два режима эмуляции: пользовательский и системный. Пользовательский режим эмуляции позволяет процессу, созданному на одном процессоре, работать на другом (выполняется динамический перевод инструкций для принимающего процессора и конвертация системных вызовов Linux). Системный режим позволяет эмулировать систему целиком, включая процессор и разнообразную периферию. Достоинством QEMU является его быстрый и компактный динамический транслятор. Динамический транслятор позволяет во время исполнения переводить инструкции целевого (гостевого) процессора в инструкции центрального процессора хоста для обеспечения эмуляции. QEMU обеспечивает динамическую трансляцию преобразованием целевой инструкции в микрооперации. Эти микрооперации представляют собой элементы С-кода, которые компилируются в объекты. Затем запускается основной транслятор, который отображает целевые инструкции на микрооперации для динамической трансляции. Такой подход не только эффективен, но и обеспечивает переносимость.

Использование QEMU в качестве эмулятора персонального компьютера обеспечивает поддержку разнообразных периферийных устройств. Сюда входят стандартные периферийные устройства — эмулятор аппаратного видеоадаптера (VGA), мыши и клавиатуры PS/2, интерфейс IDE для жестких дисков, интерфейс CD-ROM и эмуляция дисковода. Кроме того, QEMU имеет возможность эмуляции сетевых адаптеров NE2000 (PCI), последовательных портов, многочисленных звуковых плат и контроллера PCI Universal Host Controller Interface (UHCI), Universal Serial Bus (USB) (с виртуальным USB концентратором). Также поддерживается до 255 процессоров с поддержкой симметричной многопроцессорности (SMP).

Полная виртуализация использует гипервизор, который осуществляет связь между гостевой ОС и аппаратными средствами физического сервера. В связи с тем, что вся работа с гостевой операционной системой проходит через гипервизор, скорость работы данного типа виртуализации ниже чем в случае прямого взаимодействия с аппаратурой. Основным преимуществом является то, что в ОС не вносятся никакие изменения, единственное ограничение — операционная система должна поддерживать основные аппаратные средства. Архитектура полной виртуализации представлена на рисунке 2.2.

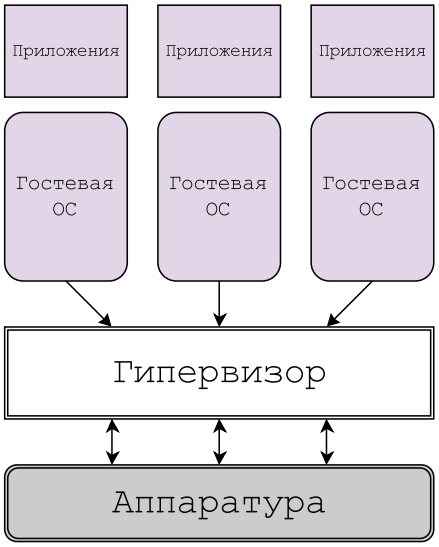

Рисунок 2.2 – Архитектура полной виртуализации

Полная виртуализация возможна исключительно при условии правильной комбинации оборудования и программного обеспечения. Например,

она была невозможной ни в серии IBM System/360, за исключением IBM System/360-67, ни в ранних IBM System/370, пока IBM не добавила оборудование виртуальной памяти в своих System/370 в 1972 г. Аналогичная ситуация и с платформой x86: полная виртуализация была возможна не в полной мере, до добавления технологий AMD-V и Intel VT.

KVM (Kernel-based Virtual Machine) — программное решение, обеспечивающее виртуализацию в среде Linux, которая поддерживает аппаратную виртуализацию на базе Intel VT (Virtualization Technology) либо AMD SVM (Secure Virtual Machine) [6]. KVM не выполняет никакой самоэмуляции, вместо этого, программа, работающая в пользовательском пространстве, применяет интерфейс /dev/kvm для настройки адресного пространства гостевого виртуального сервера, берет его смоделированные ресурсы ввода/вывода и отображает его образ на образ хоста.

В архитектуре KVM, виртуальная машина выполняется как обычный Linux-процесс, запланированный стандартным планировщиком Linux. На самом деле виртуальный процессор представляется как обычный Linux-процесс, это позволяет KVM пользоваться всеми возможностями ядра Linux. Эмуляцией устройств управляет модифицированная версия QEMU, которая обеспечивает эмуляцию BIOS, шины PCI, шины USB, а также стандартный набор устройств, таких как дисковые контроллеры IDE и SCSI, сетевые карты и другие.

Hyper-V — система аппаратной виртуализации разработанная компанией Microsoft. Hyper-V может использоваться в двух вариантах, как отдельный продукт Microsoft Hyper-V Server или как роль в Windows Server 2008 и выше. Отдельная версия Hyper-V Server является бесплатной.

Система Hyper-V поддерживает разграничение согласно понятию раздел. Раздел — логическая единица разграничения, поддерживаемая гипервизором, в котором работают операционные системы. Каждый экземпляр гипервизора должен иметь один родительский раздел, с запущенной Windows Server. Стек виртуализации запускается на родительском разделе и обладает прямым доступом к аппаратным устройствам. Затем родительский раздел

порождает дочерние разделы. Родительский раздел создает дочерние при помощи API-гипервизора, представленного в Hyper-V.

Виртуализированные разделы не имеют ни доступа к физическому процессору, ни возможности управлять его реальными прерываниями. Вместо этого у них есть виртуальное представление процессора и гостевой виртуальный адрес, зависящий от конфигурации гипервизора, вовсе необязательно при этом занимающий все виртуальное адресное пространство. Гипервизор может определять подмножество процессоров для каждого раздела. Гипервизор управляет прерываниями процессора и перенаправляет их в соответствующий раздел, используя логический контроллер искусственных прерываний. Hyper-V может аппаратно ускорять трансляцию адресов между различными гостевыми виртуальными адресными пространствами при помощи IOMMU (I/O Memory Management Unit), которое работает независимо от аппаратного управления памятью, используемого процессором.

Дочерние разделы не имеют непосредственного доступа к аппаратным ресурсам, но зато получают виртуальное представление ресурсов, называемое виртуальными устройствами. Любая попытка обращения к виртуальным устройствам перенаправляется через VMBus к устройствам родительского раздела, которые и обработают данный запрос. VMBus — это логический канал, осуществляющий взаимодействие между разделами. Ответ возвращается также через VMBus. Если устройства родительского раздела также являются виртуальными устройствами, то запрос будет передаваться дальше, пока не достигнет такого родительского раздела, где он получит доступ к физическим устройствам. Родительские разделы запускают провайдер сервиса виртуализации, который соединяется с VMBus и обрабатывает запросы доступа к устройствам от дочерних разделов. Виртуальные устройства дочернего раздела работают с клиентом сервиса виртуализации, который перенаправляет запрос через VMBus к VSP родительского раздела. Этот процесс прозрачен для гостевой ОС.

Hyper-V обеспечивает базовую поддержку виртуализации гостевых Linux-систем в режиме эмуляции устройств, не требуя никаких изменений.

Эмулируются контроллеры дисков IDE PIIX4 и PCI Ethernet адаптер DEC 21140 Tulip, однако скорость работы может быть невысокой и существует ограничение 128Гб на диск.

Паравиртуализация достижима при включении модулей ядра Linux или при установке дополнительных компонентов интеграции. Ранние версии компонентов интеграции функционировали как прослойка между интерфейсом гостевого ядра Xen и Hyper-V (Hypercall Translator). Позднее была реализована прямая поддержка шины VMBbus без Xen. 20 июля 2009 Microsoft опубликовала эти драйверы под лицензией GPL, и они были официально включены в ядро Linux (опция STAGING/HYPERV). В процессе работы над драйверами различные компоненты постепенно покидали ветку STAGING и начиная с версии ядра Linux 3.4 были перенесены в основное дерево [7]. Таким образом, дистрибутивы с ядрами новее, чем 2.6.32 могут включать встроенную поддержку паравиртуализации Hyper-V. Данные драйвера содержат поддержку шины VMbus и позволяют гостевой операционной системе Linux работать с устройствами в режиме Enlightened I/O. Поддерживаются устройства Synthetic IDE, Synthetic SCSI и Synthetic Ethernet. Поддерживаются SMP до 4 ядер и такие функции, как синхронизация времени (в RHEL5 только для 32-битных систем), остановка системы (shutdown) и проверка активности (heartbeat).

Паравиртуализация имеет некоторые сходства с полной виртуализацией. Этот метод использует гипервизор для разделения доступа к основным аппаратным средствам, но объединяет код, касающийся виртуализации, в непосредственно операционную систему, поэтому недостатком метода является то, что гостевая ОС должна быть изменена для гипервизора. Но паравиртуализация существенно быстрее полной виртуализации, скорость работы виртуальной машины приближена к скорости реальной, это осуществляется за счет отсутствия эмуляции аппаратуры и учета существования гипервизора при выполнении системных вызовов в коде ядра. Вместо привилегированных операций совершаются гипервызовы обращения ядра

гостевой ОС к гипервизору с просьбой о выполнении операции. Архитектура паравиртуализации представлена на рисунке 2.3.

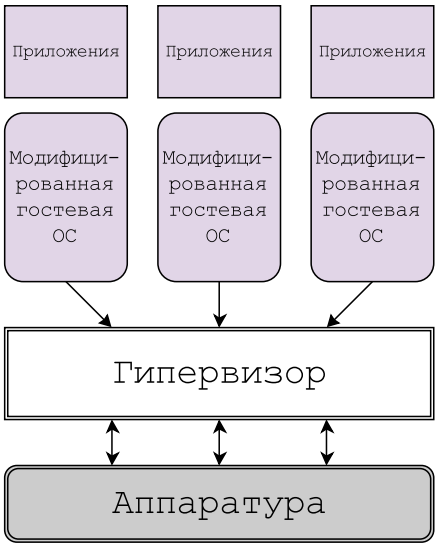

Рисунок 2.3 – Архитектура паравиртуализации

Для организации паравиртуализации используется программный продукт Xen.

Xen — это монитор виртуальных машин (VMM, Virtual Machine Monitor) или гипервизор с поддержкой паравиртуализации для процессоров x86 архитектуры, распространяющийся с открытым исходным кодом. Xen может организовать совместное безопасное исполнение нескольких виртуальных машин на одной физической системе, с производительностью близкой к непосредственной. Он перекладывает большинство задач по поддержке аппаратуры на гостевую операционную систему, работающую в управляющей виртуальной машине, также известной как домен 0 (dom0) [8]. Сам Xen содержит только код, необходимый для обнаружения и запуска остальных процессоров системы, настройки обработки прерываний и нумерации PCI шины. Драйверы устройств работают внутри привилегированной гостевой операционной системы, а не в самом Xen. Такой подход обеспечивает совместимость с большинством устройств, поддерживаемых Linux. Сборка XenLinux по умолчанию содержит поддержку большинства серверного сетевого и дискового оборудования, но при необходимости можно добавить

поддержку других устройств, переконфигурировав Linux-ядро стандартным способом.

В паравиртуальном режиме (PV) оборудование не эмулируется, и гостевая ОС должна быть специальным образом модифицирована, чтобы работать в таком окружении. Начиная с версии 3.0, ядро Linux поддерживает запуск в паравиртуальном режиме без перекомпиляции со сторонними патчами. Преимущество режима паравиртуализации состоит в том, что он не требует поддержки аппаратной виртуализации со стороны процессора, а также не тратит вычислительные ресурсы для эмуляции оборудования на шине PCI. Режим аппаратной виртуализации (HVM) появился в Xen, начиная с версии 3.0 гипервизора, и требует поддержки со стороны оборудования. В этом режиме для эмуляции виртуальных устройств используется QEMU, который «неповоротлив» даже с паравиртуальными драйверами. Однако со временем поддержка аппаратной виртуализации в оборудовании получила настолько широкое распространение, что используется даже в процессорах лэптопов. Поэтому у разработчиков возникло желание использовать быстрое переключение контекста исполнения между гипервизором и гостевой ОС и в паравиртуальном режиме, используя возможности оборудования. Так появился новый режим — аппаратная паравиртуализация (PVH), который доступен в Xen с версии 4.4.

Виртуализация уровня операционной системы отличается от других. Она использует технику, при которой сервера виртуализируются непосредственно над ОС. Недостатком метода является то, что поддерживается одна единственная операционная система на физическом сервере, которая изолирует контейнеры друг от друга. Преимуществом виртуализации уровня ОС является «родная» производительность. Виртуализация уровня ОС — метод виртуализации, при котором ядро операционной системы поддерживает несколько изолированных экземпляров пространства пользователя, вместо одного. Эти экземпляры с точки зрения пользователя полностью идентичны реальному серверу. Для систем на базе UNIX, эта технология может рассматриваться как улучшенная реализация механизма chroot. Ядро

обеспечивает полную изолированность контейнеров, поэтому программы из разных контейнеров не могут воздействовать друг на друга. Архитектура виртуализации уровня ОС представлена на рисунке 2.4.

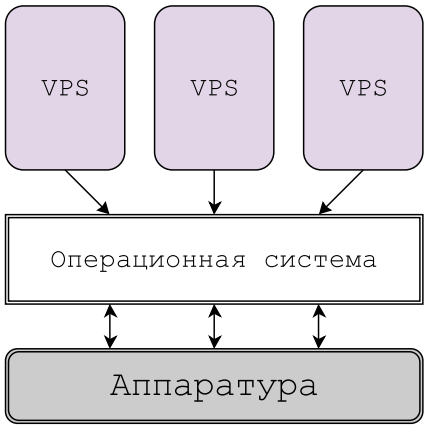

Рисунок 2.4 – Архитектура виртуализации уровня ОС

Для реализации виртуализации уровня операционной системы часто используется продукт OpenVZ.

OpenVZ разрабатывается как патч к исходным текстам ядра Linux. В модифицированном ядре добавлен массив дополнительных сущностей — виртуальных окружений (VE) или контейнеров (CT), а для всех имеющихся объектов (процессы, сокеты и прочие) введены дополнительные поля — номер контейнера, к которому этот объект относится, и номер объекта внутри контейнера. Каждое виртуальное окружение имеет собственный набор квот на потребление системных ресурсов и отдельный каталог для использования в качестве корневой файловой системы. Дополнительные модули ядра — vzdev, vzmon и прочие, отвечают за работу ограничений, мониторинг, эмуляцию сети в контейнере, сохранение и восстановление текущего состояния запущенных контейнеров. К преимуществам OpenVZ, по сравнению с более универсальными инструментами виртуализации, такими как Xen и KVM, относят прозрачный доступ из внешней системы к процессам, файлам и прочим ресурсам в контейнере.

OpenVZ разрабатывается фирмой Parallels как часть более крупного коммерческого продукта под названием Parallels Virtuozzo Containers (PVC) [9]. В число преимуществ Virtuozzo, по сравнению с OpenVZ, входят:

- файловая система VZFS;
- управление через графическую консоль и веб-интерфейс;
- программный интерфейс на базе XML для создания собственных инструментов управления и контроля;
- средства миграции с физическом системы в контейнер и обратно;
- средства контроля за полосой и суммарным потреблением трафика;
- интеграция с Plesk, коммерческой панелью управления хостингом;
- круглосуточная техническая поддержка.

VZFS позволяет совмещать файловые системы контейнеров, при этом базовый образ используется всеми контейнерами, а изменения в нем для каждого контейнера сохраняются раздельно. Преимущества такого подхода:

- место, занимаемое программами на диске, становится фиксированным и не зависит от количества контейнеров, в которые эти программы установлены;
- уменьшается расход оперативной памяти, так как код нескольких экземпляров программы или библиотеки, запущенной из одного и того же исполняемого файла, размещается в памяти в единственном экземпляре;
- обновление программного обеспечения в группе контейнеров выполняется одной командой.

LXC (Linux Containers) — система виртуализации на уровне операционной системы. Данная система сходна с OpenVZ и Linux-VServer для Linux, а также FreeBSD jail и Solaris Containers. LXC основана на технологии cgroups, входящей в ядро Linux, начиная с версии 2.6.29. Ее нельзя рассматривать как законченный продукт, фактически это набор из нескольких совершенно самостоятельных функций ядра Linux и пользовательских утилит, которые позволяют удобно создавать и управлять изолированными

контейнерами [10]. Практически вся функциональность LXC представления известными механизмами ядра cgroups и namespaces:

cgroups (Control Groups) — позволяет ограничить аппаратные ресурсы некоторого набора процессов. Под аппаратными ресурсами подразумеваются: процессорное время, память, дисковая и сетевая подсистемы. Набор или группа процессов могут быть определены различными критериями. Например, это может быть целая иерархия процессов, получающая все лимиты родительского процесса. Кроме этого, возможен подсчет расходуемых группой ресурсов, заморозка (freezing) групп, создание контрольных точке (checkpointing) и их перезагрузка. Для управления этим полезным механизмом существует специальная библиотека libcgroups, в состав которой входят такие утилиты, как cgcreate, cgexec и некоторые другие.

namespaces — пространства имен. Это механизм ядра, который позволяет изолировать процессы друг от друга. Изоляция может быть выполнена в шести контекстах (пространствах имен):

- mount — предоставляет процессам собственную иерархию файловой системы и изолирует ее от других таких же иерархий, по аналогии с chroot;
- PID — изолирует идентификаторы процессов (PID) одного пространства имен от процессов с такими же идентификаторами другого пространства;
- network — предоставляет отдельным процессам логически изолированный от других стек сетевых протоколов, сетевой интерфейс, IP-адрес, таблицу маршрутизации, ARP и прочие реквизиты;
- IPC — обеспечивает разделяемую память и взаимодействие между процессами;
- UTS — изоляция идентификаторов узла, такого как имя хоста (hostname) и домена (domainname);
- user — позволяет иметь один и тот же набор пользователей и групп в рамках разных пространств имен, в каждом контейнере могут быть свой root и любые другие пользователи и группы.

Одно из главных преимуществ LXC — это присутствие его базовых блоков (cgroups и namespaces) во всех современных ядрах Linux. Это означает, что нет необходимости что-то компилировать или использовать стороннее ядро, как в случае с OpenVZ. Единственное, что необходимо установить, это пакет утилит управления (vzctl, Docker, libvirt, systemd).

К системам управления можно отнести Docker.

Docker — программное обеспечение для автоматизации развертывания и управления приложениями в среде виртуализации на уровне операционной системы, например LXC. Docker позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть перенесен на любой Linux-системе с поддержкой cgroups в ядре, а также предоставляет среду по управлению контейнерами.

Для экономии дискового пространства проект использует файловую систему Aufs с поддержкой каскадно-объединенного монтирования: контейнеры используют образ базовой операционной системы, а изменения записываются в отдельную область. Также поддерживается размещение контейнеров в файловой системе Btrfs с включенным режимом копирования при записи. В состав программных средств входит демон — сервер контейнеров, клиентские средства, позволяющие из интерфейса командной строки управлять образами и контейнерами, а также API, позволяющий в стиле REST управлять контейнерами программно. Демон обеспечивает полную изоляцию запускаемых на узле контейнеров на уровне файловой системы (у каждого контейнера собственная файловая система), на уровне процессов (процессы имеют доступ только к собственной файловой системе контейнера, а ресурсы разделены средствами LXC), на уровне сети (каждый контейнер имеет доступ только к привязанному к нему сетевому пространству имен и соответствующим виртуальным сетевым интерфейсам).

Набор клиентских средств позволяет запускать процессы в новых контейнерах, останавливать и запускать контейнеры, приостанавливать и возобновлять процессы в контейнерах. Серия команд позволяет осуществлять мониторинг запущенных процессов (по аналогии с ps, top). Новые

образы возможно создавать из специального сценарного файла (dockerfile), возможно записать все изменения, сделанные в контейнере в новый образ. Все команды могут работать как с docker-демоном локальной системы, так и с любым сервером docker, доступным по сети. Кроме того, в интерфейсе командной строки встроены возможности по взаимодействию с публичным репозиторием Docker Hub, в котором размещены предварительно собранные образы контейнеров, образы можно скачивать в локальную систему, возможно также отправить локально собранные образы в Docker Hub.

Виртуальная инфраструктура — это частное облако, размещаемое на оборудовании провайдера. По сути, виртуальная инфраструктура представляет собой динамическое распределение ресурсов в соответствии предприятия. Виртуальная машина использует материальные ресурсы одного компьютера, а виртуальная инфраструктура — материальные ресурсы всей ИТ-среды, формируя из компьютеров, а также из подключенных к ним сетей и хранилищ единый пул ИТ-ресурсов.

Виртуальная инфраструктура включает в себя следующие компоненты [11]:

- гипервизоры для одного узла для полной виртуализации каждого компьютера;
- пакет услуг инфраструктуры распределенных систем на основе виртуализации (например, управление ресурсами) для оптимального распределения доступных ресурсов между виртуальными машинами;
- решения для автоматизации, обеспечивающие особые возможности оптимизации того или иного ИТ-процесса (например инициализации или восстановления в критических ситуациях).

Благодаря отделению всей программной среды от исходной аппаратной инфраструктуры виртуализация позволяет объединить ряд серверов, инфраструктур хранения и сетей в единый пул ресурсов, динамически, безопасно и надежно распределяемый между приложениями по мере необходимости. С помощью этого инновационного решения организации могут создать вычислительную инфраструктуру с максимальной эффективностью,

доступностью, автоматизацией и гибкостью, состоящую из недорогих серверов, соответствующих отраслевому стандарту.

Предпочтение виртуальной инфраструктуре отдают по причинам:

- экономии на обслуживающем персонале при условии сохранения 100% отказоустойчивости системы;
- отсутствии необходимости выделять бюджет на модернизацию оборудования;
- возможности объединения в общую виртуальную среду офисы, географически находящиеся в разных местах;
- возможности быстрого масштабирования проекта в соответствии с текущими решаемыми задачами;
- быстрого доступа к данным.

Под каждый тарифный план создается изолированное частное облако с фиксированным количеством выделенных ресурсов. Выделенное в рамках тарифного плана частное облако становится гибкой виртуальной оболочкой. Функционирующие внутри нее виртуальные машины могут объединяться в виртуальные сети. Изменение их вычислительной мощности происходит в зависимости от решаемых задач. Смена или трансформация тарифных планов производится в режиме реального времени без перебоев в работе.

В случае необходимости, пользователь имеет возможность создавать нужное количество сегментов сети, добиваясь удобной конфигурации виртуальной рабочей среды.

Управление ресурсами осуществляется через удобный для пользователя веб-интерфейс, что позволяет делать все необходимые операции, такие как:

- создание арендуемых виртуальных серверов самостоятельно;
- изменение конфигурации за несколько минут, часть операций даже без остановки и перезагрузки сервера (для некоторых операционных систем);
- включение, выключение, установка, переустановка ОС и приложений самостоятельно и удаленно;

– осуществление резервного копирование и сохранение состояний работающих виртуальных машин.

3 СИСТЕМНЫЙ АНАЛИЗ ВИРТУАЛЬНОЙ ИНФРАСТРУКТУРЫ

На данный момент при анализе и синтезе сложных программных и аппаратных систем все чаще используется системный подход. Важным моментом для системного подхода является определение структуры системы — совокупности связей между элементами системы, отражающих их взаимодействие. Совокупность элементов и связей между ними позволяет судить о структуре системы.

Принципы системного анализа — это некоторые положения общего характера, являющиеся обобщением опыта работы человека со сложными системами. Общепринятых формулировок в настоящее время нет, но все формулировки так или иначе описывают одни и те же понятия. Пренебрежение принципами при проектировании любой нетривиальной технической системы, непременно приводит к потерям того или иного характера, от увеличения затрат в процессе проектирования до снижения качества и эффективности конечного продукта.

Системный анализ выполнен в соответствии с [12].

3.1 Принцип конечной цели

Принцип конечной цели — это абсолютный приоритет конечной цели, он имеет несколько правил:

- для проведения системного анализа необходимо, в первую очередь, сформулировать цель функционирования системы, так как не полностью определенные цели влекут за собой неверные выводы;
- анализ системы следует вести на базе уяснения основной цели, что позволит определить существенные свойства показателей качества и критериев оценки;
- при синтезе систем любая попытка изменения или совершенствования должна оцениваться относительно конечной цели;
- цель функционирования искусственной системы задается, как правило, системой, в которой исследуемая система является составной частью.

При использовании данного принципа, разрабатываемая виртуальная инфраструктура будет рассматриваться в виде «черного ящика», функционирование которого описывается формулой (3.1):

$$Y=F(X, Z) \quad (3.1)$$

где Y — выходной вектор системы, который функционально зависит от входного вектора X и вектора внутреннего состояния системы Z (рисунке 3.1). Входными данными (вектор X) будут являться запросы пользователей на обработку и обслуживание. Внутреннее состояние системы (вектор Z) представляет собой создание услуг по требованию пользователя. Выходными данными (вектор Y) будут являться предоставление доступа к услугам пользователя.

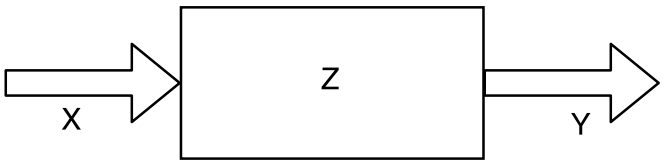

Рисунок 3.1 – Проектируемая система в виде черного ящика

В соответствии с данным принципом должна быть четко сформулирована конечная цель — назначение проектируемой системы и сформирован список функций, которые должна выполнять система. Кроме того, необходимо определить перечень входных воздействий, на которые реагирует система.

Цель проектирования — разработка виртуальной инфраструктуры для реализации облачных услуг. Список функций проектируемой системы:

- Ф1 — прием и регистрация обращений пользователей;
- Ф2 — идентификация и обработка инцидентов и запросов на обслуживание;
- Ф3 — создание, смена, обновление и удаление услуг по требованию;
- Ф4 — предоставление доступа к услугам;
- Ф5 — мониторинг состояния инфраструктуры.

Перечень входных воздействий на систему:

– внесение изменений данных о заявке;

– мониторинг хода решения заявки.

3.2 Принцип единства

Принцип единства — это совместное рассмотрение системы как целого и как совокупности частей. Принцип ориентирован на декомпозицию с сохранением целостных представлений о системе.

На основании функций проектируемой системы, представленных выше, в ней можно выделить следующие подсистемы:

а) подсистема взаимодействия с пользователем;

б) подсистема управления услугами;

в) подсистема управления инфраструктурой;

г) подсистема защиты и обеспечения целостности данных.

На рисунке 3.2 представлена схема взаимодействия между подсистемами.

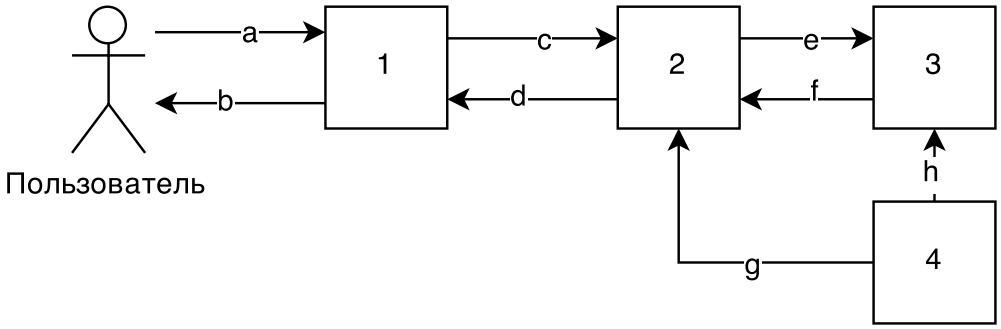

Рисунок 3.2 – Взаимодействие между подсистемами и их связь с окружающей средой

Обозначения, приведенные на рисунке 3.2 требуют пояснения:

a — информация, предоставляемая пользователем, передается на сервер;

b — выходная информация (результат выполнения);

c — проверка корректности переданных данных;

d — в случае неправильно введенных данных, возвращается управление к подсистеме взаимодействия с пользователем;

e — создание услуги;

f — возврат информации о созданной услуге;

g — обеспечение целостности информации об услуге;

h — обеспечение целостности данных пользователя.

3.3 Определение взаимосвязей между подсистемами на основе принципа связности

Рассмотрение любой части совместно с ее окружением подразумевает проведение процедуры выявления связей между элементами системы и выявление связей со средой. В соответствии с этим принципом систему в первую очередь следует рассматривать как часть другой системы, называемой суперсистемой или старшей системой.

Подсистема взаимодействия с пользователем представлена на рисунке 3.3.

Рисунок 3.3 – Внутренние и внешние связи подсистемы взаимодействия с пользователем

Подсистема управления услугами представлена на рисунке 3.4.

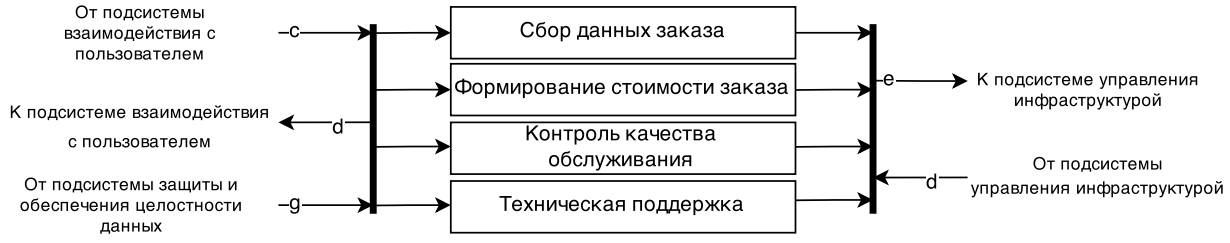

Рисунок 3.4 – Внутренние и внешние связи подсистемы управления услугами

Подсистема управления инфраструктурой представлена на рисунке 3.5.

Рисунок 3.5 – Внутренние и внешние связи подсистемы управления инфраструктурой

Подсистема защиты и обеспечения целостности данных представлена на рисунке 3.6.

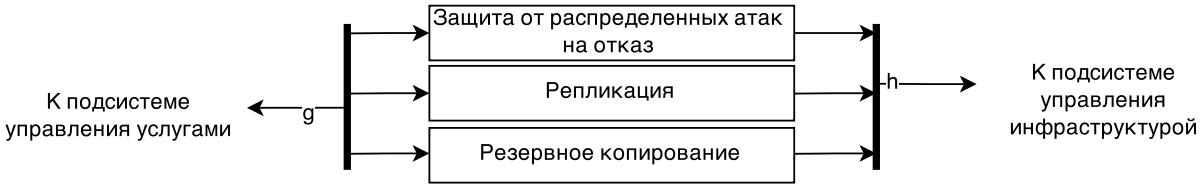

Рисунок 3.6 – Внутренние и внешние связи подсистемы защиты и обеспечения целостности данных

3.4 Принцип модульности

Полезно выделение модулей в системе и рассмотрение ее как совокупности модулей. Принцип указывает на возможность вместо части системы исследовать совокупность ее входных и выходных воздействий (абстрагирование от излишней детализации).

Принцип модульности для разрабатываемой системы поясняется с помощью рисунке 3.7, описывающего разбиение на модули системы взаимодействия с пользователем.

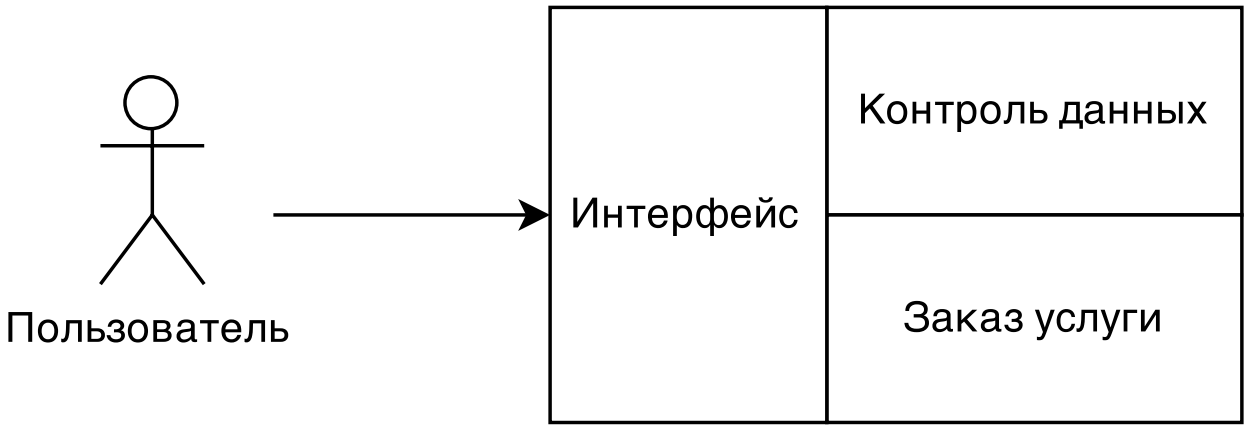

Рисунок 3.7 – Принцип модульности на примере подсистемы взаимодействия с пользователем

Излишняя детализация не требуется, поэтому остальные системы на модули принято решение не разбивать.

3.5 Принцип функциональности

Принцип утверждает, что любая структура тесно связана с функцией системы и ее частей. В случае придания системе новых функций полезно пересматривать ее структуру, а не пытаться втиснуть новую функцию в старую схему. Поскольку выполняемые функции составляют процессы, то целесообразно рассматривать отдельно процессы, функции, структуры. В свою очередь, процессы сводятся к анализу потоков различных видов:

- материальный поток;
- поток энергии;
- поток информации;
- смена состояний.

С этой точки зрения структура есть множество ограничений на потоки в пространстве и времени.

Функции подсистем приведены в пункте 3.1.

Матрица инциденций функций системы и функций назначения подсистем приведена в таблице 3.1.

Таблица 3.1 – Матрица инциденций

Функции	Подсистемы				Виртуальная инфраструктура
	1	2	3	4	
Ф1	+				+
Ф2	+				+
Ф3		+			+
Ф4		+			+
Ф5			+	+	+

В матрице инциденций знаком «+» обозначены функции, которые реализуются для каждой из подсистем.

Детализация функций подсистемы на примере подсистемы взаимодействия с пользователем:

а) регистрация и авторизация пользователя в платежной системе;

б) обеспечение перечня услуг;

в) обеспечение возможных способов оплаты услуг;

г) возможность уточнения или изменения услуги.

Входными данными для подсистемы является информация о пользователе, а выходными — подтвержденный заказ услуг.

3.6 Принцип иерархии

Полезно введение иерархии частей и их ранжирование, что упрощает разработку системы и устанавливает порядок рассмотрения частей. Разработка иерархий классов является нетривиальной задачей. Грамотно спроектированные иерархии классов позволяют создавать высокоэффективные системы. Плохо спроектированная иерархия приводит к созданию сложных и запутанных систем.

Выполнение принципа иерархичности для разрабатываемой системы на примере подсистемы защиты и обеспечения целостности данных проиллюстрировано на рисунке 3.8.

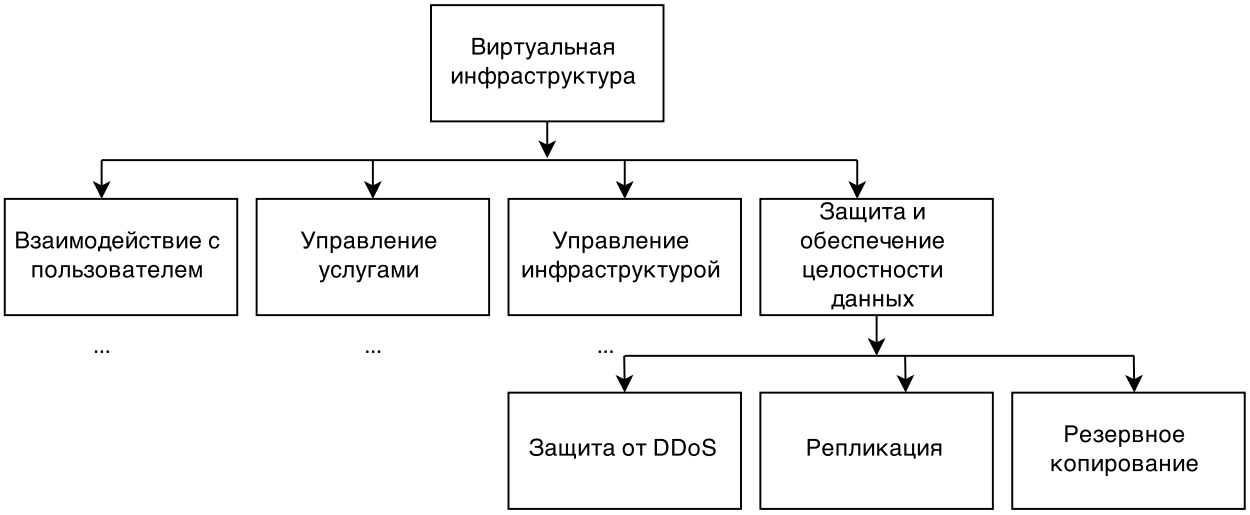

Рисунок 3.8 – Принцип иерархии на примере подсистемы защиты и обеспечения целостности данных

3.7 Принцип сочетания централизации и децентрализации

Степень централизации должна быть минимальной, обеспечивающей выполнение поставленной цели. Соотношение централизации и децентрализации определяется уровнями, на которых вырабатываются и принимаются управленческие решения.

Недостаток децентрализованного управления — увеличение времени адаптации системы. Он существенно влияет на функционирование системы в быстро меняющихся средах. То, что в централизованных системах можно сделать за короткое время, в децентрализованной системе будет осуществляться весьма медленно. Данный недостаток нивелируется налаживанием горизонтальных связей.

Недостатком централизованного управления является сложность управления из-за огромного потока информации, подлежащей переработке в старшей системе управления. Поэтому в сложной системе обычно присутствуют два уровня управления. В медленно меняющейся обстановке децентрализованная часть системы успешно справляется с адаптацией поведения системы к среде и с достижением глобальной цели системы за счет оперативного управления, а при резких изменениях среды осуществляется централизованное управление по переводу системы в новое состояние.

Например, можно выполнить декомпозицию подсистемы взаимодействия с пользователем таким образом:

а) подсистема работы с пользователями;

б) подсистема работы пользователей с услугами;

в) подсистема учета финансовых средств.

Такое разбиение позволит реализовать полученные подмножества в виде отдельных модулей.

3.8 Принцип развития

Учет изменяемости системы, ее способности к развитию, адаптации, расширению, замене частей, накапливанию информации. В основу синтезируемой системы требуется закладывать возможность развития, наращивания усовершенствования. Обычно расширение функций предусматривается за счет обеспечения возможности включения новых модулей, совместимых с уже имеющимися. С другой стороны, при анализе принцип развития ориентирует на необходимость учета предыстории развития системы и тенденций, имеющихся в настоящее время, для вскрытия закономерностей ее функционирования.

Одним из способов учета этого принципа разработчиками является рассмотрение системы относительно ее жизненного цикла. Условными фазами жизненного цикла являются: проектирование, изготовление, ввод в эксплуатацию, эксплуатация, наращивание возможностей (модернизация), вывод из эксплуатации (замена), уничтожение.

Отдельные авторы этот принцип называют принципом изменения (историчности) или открытости. Для того чтобы система функционировала, она должна изменяться, взаимодействовать со средой.

Проектируемая система может быть развита следующим образом:

- добавлением модуля конвертации валюты для оплаты услуги;
- расширением перечня услуг;
- внедрением дополнительных технологий виртуализации;
- увеличением штата технической поддержки;

– введением расширенного мониторинга;
– многоуровневой защитой от атак на отказ;
– расширением памяти на системах хранения данных;
– многоуровневой репликацией и резервным копированием.

3.9 Принцип учета случайностей

Принцип утверждает, что можно иметь дело с системой, в которой структура, функционирование или внешние воздействия не полностью определены.

Сложные системы не всегда подчиняются вероятностным законам. В таких системах можно оценивать «наихудшие ситуации» и рассмотрение проводить для них. Этот способ обычно называют методом гарантируемого результата. Он применим, когда неопределенность не описывается аппаратом теории вероятностей.

При наличии информации о вероятностных характеристиках случайностей можно определять вероятностные характеристики выходов в системе.

События и действия, некорректные с точки зрения правил функционирования системы:

– попытка заказа услуги без наличия финансовых средств на счету;
– неподтвержденные заказы услуг;
– сбой в работе аппаратуры, в частности устройств хранения данных;
– сбой сети в в пределах дата-центра или на магистрали;
– отсутствие своевременной работы технической поддержки;
– ошибки и уязвимости в используемых программных платформах.

Кроме того, необходимо вести контроль успешности и целостности проведения операций с компонентами системы, данными пользователей и корректно обрабатывать исключения, возникновение которых возможно в процессе работы системы.

Перечисленные принципы обладают высокой степенью общности. Такая интерпретация может привести к обоснованному выводу о незначимости какого-либо принципа. Однако знание и учет принципов позволяют лучше

увидеть существенные стороны решаемой проблемы, учесть весь комплекс взаимосвязей, обеспечить системную интеграцию.

4 ОПИСАНИЕ ВИРТУАЛЬНОЙ ИНФРАСТРУКТУРЫ

4.1 Назначение виртуальной инфраструктуры

5 РУКОВОДСТВО АДМИНИСТРАТОРА

6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

7 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

8 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

8.1 Краткая характеристика помещения

Согласно санитарным нормам, ширина стола должна быть не менее 60 см, глубина не менее 80 см, также необходимо обеспечить расстояние между боковыми поверхностями мониторов не менее 1.2 м.

На рис. 8.1 изображена планировка и размещение оборудования на рабочем месте.

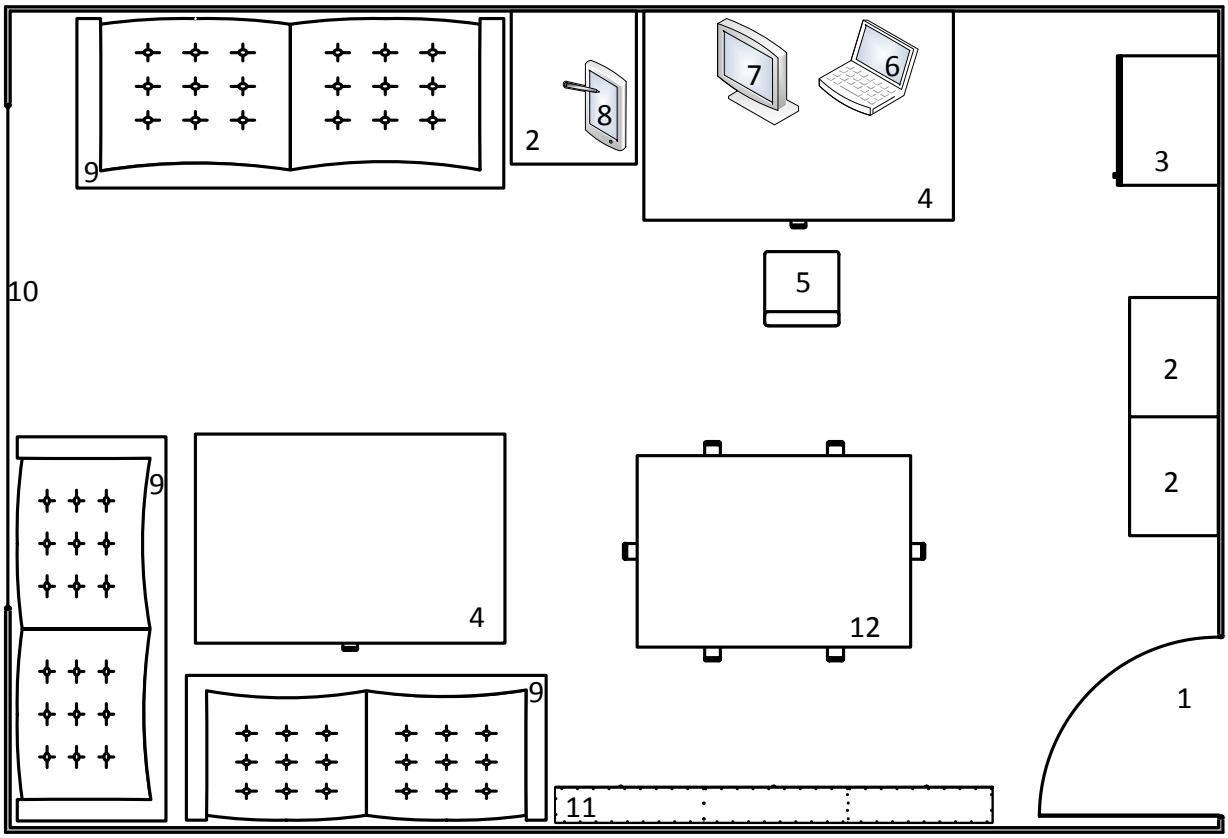

Рисунок 8.1 – Планировка и размещение оборудования на рабочем месте

1 — дверь; 2 — тумбочка; 3 — холодильник; 4 — стол; 5 — кресло; 6 — ноутбук; 7 — монитор; 8 — планшет; 9 — кровать; 10 — окно; 11 — шкаф; 12 — обеденный стол

8.2 Микроклимат

Метеорологические условия (микроклимат) характеризуются следующими параметрами:

- температурой воздуха;
- относительной влажностью;
- скоростью движения воздуха на рабочем месте;
- барометрическим давлением.

В комнате, где находится персональный компьютер, должен поддерживаться определенный температурный режим для нормальной эксплуатации ЭВМ и условий труда администратора.

Параметры воздушной среды в кабинете должны соответствовать требованиям ГОСТ 12.1.005-88 «Воздух рабочей зоны». При этом необходимо учитывать, что работа администратора относится к разряду легких работ (разряд 1а — затраты энергии до 150 ккал/час), а кабинет — к производственным помещениям.

В помещении имеются источники избыточного тепла:

- тепловыделение от ноутбука;
- тепло, выделяемое персоналом.

Тепловыделение от светильников отсутствует, так как используется люминисцентное освещение. Для поддержания оптимального уровня температуры используется как естественная вентиляция (через двери и окна путем проветривания), так и искусственная (при помощи вентилятора), так как в текущих условиях проживания невозможно установить кондиционер. Для обогрева помещения в зимнее время используется водяное отопление.

8.3 Шум и вибрация

Источниками шума в помещении являются:

- непосредственно персональный компьютер (вентиляторы охлаждения процессора и видеокарты);
- разговорная речь;
- шум вне рабочей зоны.

Постоянный шум оказывает отрицательное воздействие на человека, как биологически, так и психологически, что отражается на качестве работы и общей производительности труда сотрудников. Снижается производительность труда и повышается количество допущенных ошибок, некоторые из которых могут быть критическими.

В помещениях для администраторов по ГОСТ 12.1.003-83 «Шум. Общие требования безопасности» допустимые уровни звукового давления приведены в табл. 8.2. Допустимый уровень звука — 50 дБА, при работающем оборудовании в кабинете ожидаемый уровень звука — 40-48 дБА.

Таблица 8.2 – Допустимые уровни звукового давления

Тип помещения	Допустимые уровни звукового давления
Служебные помещения	50 дБА
Помещения для отдыха	38 дБА

Сравнив допустимый и ожидаемый уровни звука в помещении, видно, что нет необходимости проводить мероприятия по борьбе с шумом.

8.4 Освещение

В зависимости от необходимости, производственное освещение в кабинете может быть как естественным, создаваемым непосредственно солнцем и диффузным (рассеянным) светом, так и искусственным, осуществляемым электрическими лампами. Естественное освещение характеризуется тем, что создаваемое освещение изменяется в очень широких пределах, в зависимости от времени года, дня и метеорологических факторов. При выборе норм естественного освещения учитывается разряд выполняемых работ, система освещения, коэффициент солнечности, коэффициент светового климата. По СНиП П-4-79 для 3 разряда зрительной работы (контраст большой, фон светлый) освещенность при боковом освещении от окон равна 2.0%.

Для искусственного освещения учитывается разряд выполняемых зрительных работ, подразряд работы, вид ламп, система освещения. В данном

случае подразряд работы будет «В» при котором нормируемое значение для комбинированного освещения 2500 лк и 750 лк для общего освещения.

В кабинете имеются светильники с люминесцентными лампами расположены параллельно стене. Коэффициент естественного освещения E_H должен быть не ниже 1.5%.

Мероприятия, за счет которых выполняются требования норм СНиП П-4-79:

- проверка, не реже одного раза в год, соответствия освещенности на рабочей поверхности нормам искусственного освещения;
- очистка светильников не реже одного раза в три месяца;
- протирка окон (стекол) не реже двух раз в год.

Освещение рабочих мест в помещении для работы администратора должно планироваться так, чтобы свет не падал прямо в глаза, отсутствовали мерцающие тени, блики, «мигание» люминесцентных ламп, яркость должна быть распределена равномерно.

8.5 Электробезопасность

В кабинет электроэнергия поступает для питания персональных компьютеров и электрического освещения. Питание осуществляется от трехфазной сети переменного тока напряжением 380/220 В (+10..-15%) частотой 50 Гц (+1 Гц).

Поскольку помещение сухое (относительная влажность не более 75%), температура не превышает 30 °С, то, согласно ПУЭ («Правилам устройства электроустановок»), оно не относится к категории помещений повышенной опасности. Однако возможна потенциальная опасность поражения людей электрическим током. Источниками и причинами опасности являются:

- открытые токопроводящие части оборудования, кабельной проводки;
- неисправность электрооборудования, электрических розеток;
- короткое замыкание в результате повреждения изоляции.

Для предотвращения поражения электрическим током потребителей электроэнергии в кабинете необходимо предусмотреть следующие технические мероприятия:

- все токопроводящие части оборудования и кабельной проводки должны быть защищены ограждающими кожухами;
- все металлические конструкции, которые могут оказаться под напряжением в результате короткого замыкания, должны быть заземлены, защищены и выполнено защитное отключение.

В качестве заземляющих проводников должны быть использованы элементы металлических конструкций, металлическое обрамление кабельных каналов. Здание должно быть оборудовано комплексом мер, предотвращающих попадание энергии молнии в электрическую сеть, а также поражение людей, для чего на здание устанавливаются громоотводы.

Кроме технических, необходимо проведение организационных мероприятий:

- к работе с электроустановками допускаются только лица, прошедшие инструктаж и проверку знаний правил техники безопасности в соответствии с ГОСТ 12.0.004-79, ПТЭ и ПТБ;
- периодически осуществляется контроль сопротивления электрической изоляции токоведущих частей (в соответствии с требованиями ПУЭ, оно не должно быть ниже 0.5 мм по отношению к корпусу ЭВМ).

8.6 Пожаробезопасность

Пожар в помещении может возникнуть при взаимодействии горючих веществ, окислителя (условия пожара) и источников воспламенения (причина пожара). Горючие вещества в кабинете: деревянные столы, двери, полы (паркет), покрытия стен, изоляция соединительных кабелей, жидкости для протирки узлов компьютера и другие.

Возможные источники и причины возникновения пожара:

- эксплуатация неиспользованного оборудования;
- неправильное применение электронагревательных приборов;

– короткое замыкание;
– неисправность проводки;
– нарушение правил пожарной безопасности.

Для отвода тепла от персонального компьютера необходимы работающие вентиляторы, помещение проветривается, поэтому кислород, как окислитель процессов горения, имеется в достаточном количестве. Исходя из этого, помещение кабинета, согласно нормам СНиП-П-90-81, по степени пожаробезопасности следует отнести к категории Д (помещения, в которых в обращении находятся негорючие вещества и материалы в холодном состоянии).

В качестве средств тушения пожара применяются углекислотные огнетушители, используемые для тушения электроустановок, находящихся под напряжением.

8.7 Эргономика и техническая эстетика

Эффективность работы администратора (программиста) во многом зависит от организации рабочих мест. Рабочее положение администратора — сидячее. Стул по возможности должен быть регулируемым по высоте, поскольку клавиатура и дисплей компьютеров должны находиться в зоне наилучшего обзора. Для сохранения работоспособности имеет большое значение выбор основной рабочей позы.

Техническая эстетика позволяет снижать нервное утомление и вредные воздействия на функции организма в процессе труда. Огромное значение в эстетическом оформлении производства имеет цвет. Окраска, форма, внешний вид производственного помещения и оборудования улучшают условия освещения, а также психологическое состояние человека. Стены имеют светло-зеленый цвет, не вызывающий раздражения, потолок — белый цвет, что обеспечивает максимальное отражение света.

Рассматриваемое помещение соответствует требованиям ГОСТ 12.2.032-78.

8.8 Режим труда и отдыха

Работа администратора относится к категории работ связанных с опасными и вредными условиями труда. В процессе труда на администратора оказывают действие следующие опасные и вредные производственные факторы. Физические:

- повышенный уровень статического электричества;
- повышенные уровни запыленности воздуха рабочей зоны;
- повышенный уровень шума;
- повышенный или пониженный уровень освещенности;
- неравномерность распределения яркости в поле зрения;
- повышенная яркость светового изображения;
- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.

Химические:

- напряжение зрения;
- напряжение внимания;
- интеллектуальные нагрузки;
- монотонность труда;
- большой объем информации, обрабатываемой в единицу времени;
- нерациональная организация рабочего места.

Биологическим фактором является повышенное содержание в воздухе рабочей зоны микроорганизмов.

Рациональный режим труда и отдыха — это правильное чередование работы и перерывов в ней в течение смены, суток, недели, года, устанавливаемое с целью обеспечения высокой производительности труда и сохранения здоровья работающих. Основным перерывом является перерыв на обед. Обеденный перерыв при 8-часовой рабочей смене устанавливается продолжительностью не менее 30 мин через 4 часа после начала работы.

Режим отдыха складывается из нескольких компонентов:

- времени на гигиенические процедуры и личные надобности (2-3) от сменного времени независимо от вида труда;
- времени регламентированных перерывов на отдых (входят в состав рабочего времени), определяемого по показателю условий труда или по интегральному показателю снижения работоспособности;
- времени микропауз, а также времени обеденного перерыва (нерабочего времени), остающегося от приема пищи.

Режим труда и отдыха должен быть построен в соответствии с особенностями трудовой деятельности пользователей персонального компьютера и характером функциональных изменений со стороны различных систем организма работников.

ЗАКЛЮЧЕНИЕ

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Каталог программных продуктов семейства Oracle [Электронный ресурс]. – Электрон. текстовые данные (11126315 bytes) – Режим доступа: http://oracle.ocs.ru/files/catalog_Oracle_Database_12C.pdf

2. Tier: уровни надежности ЦОД и что из этого следует [Электронный ресурс]. – Электрон. текстовые данные (38916 bytes) – Режим доступа: http://www.aboutdc.ru/page/390.php

3. Виртуальный Linux. Обзор методов виртуализации, архитектур и реализаций [Электронный ресурс]. – Электрон. текстовые данные (109051 bytes) – Режим доступа: http://www.ibm.com/developerworks/ru/library/l-linuxvirt/index.html

4. Руководство по созданию и управлению контейнерами на базе OpenVZ [Электронный ресурс]. – Электрон. текстовые данные (55237 bytes) – Режим доступа: https://github.com/Amet13/openvz-tutorial/blob/master/main.pdf

5. Эмуляция систем с помощью QEMU [Электронный ресурс]. – Электрон. текстовые данные (84008 bytes) – Режим доступа: http://www.ibm.com/developerworks/ru/library/l-qemu/

6. Гипервизоры, виртуализация и облако: Анализ гипервизора KVM [Электронный ресурс]. – Электрон. текстовые данные (79196 bytes) – Режим доступа: http://www.ibm.com/developerworks/ru/library/cl-hypervisorcompare-kvm/

7. Hyper-V drivers in mainline kernel - what’s next [Электронный ресурс]. – Электрон. текстовые данные (146405 bytes) – Режим доступа: https://social.technet.microsoft.com/Forums/en-US/7007beff-db11-4261-a9f2-0f4461a9cc92/hyperv-drivers-in-mainline-kernel-whats-next

8. Руководство пользователя Xen v3.0 [Электронный ресурс]. – Электрон. текстовые данные (272945 bytes) – Режим доступа: http://xgu.ru/xen/manual/

9. Контейнеризация на Linux в деталях – LXC и OpenVZ Часть 2 [Электронный ресурс]. – Электрон. текстовые данные (186899 bytes) – Режим доступа: http://habrahabr.ru/company/FastVPS/blog/209084/

10. Виртуализация на уровне ОС: теория и практика LXC [Электронный ресурс]. – Электрон. текстовые данные (118259 bytes) – Режим доступа: http://creativeyp.com/568-virtualizaciya-na-urovne-os-teoriya-i-praktika-lxc.html

11. Основы виртуализации. Инфраструктура [Электронный ресурс]. – Электрон. текстовые данные (32477 bytes) – Режим доступа: http://www.bw-it.ru/virtualization_virtual_infrastructure.php

12. Методические указания «Процедура системного анализа при проектировании программных систем» для студентов-дипломников дневной и заочной формы обучения специальности 7.091501 / Сост.: Сергеев Г.Г., Скатков А.В., Мащенко Е.Н. – Севастополь: Изд-во СевНТУ, 2005. – 32с.