

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

# ИСПОЛЬЗОВАНИЕ ОБЛАЧНЫХ ТЕХНОЛОГИЙ ПРИ СОЗДАНИИ РЕГИОНАЛЬНОГО ЦЕНТРА КОЛЛЕКТИВНОГО ДОСТУПА К ОБРАЗОВАТЕЛЬНЫМ ПРОДУКТАМ

Монография

Рекомендовано к изданию ученым советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет»

Оренбург  
2018

УДК 004.42  
ББК 32.973.26-018.2  
И88

Рецензент – профессор, доктор технических наук Н.Ф. Бахарева  
Авторы: И.П. Болодурина, А.Л. Коннов, П.Н. Полежаев, Ю.А. Ушаков, А.Е. Шухман, Л.В. Легашев, Д.И. Парфенов

Издание осуществлено при финансовой поддержке Министерства образования Оренбургской области (грант №37 от 30 июня 2016 г.), РФФИ (проекты № 16-07-01004, 16-47-560335), Президента Российской Федерации, стипендии для молодых ученых и аспирантов (СП-2179.2015.5).

И88      Использование облачных технологий при создании регионального центра коллективного доступа к образовательным продуктам : монография / И.П. Болодурина, А.Л. Коннов, П.Н. Полежаев, Ю.А. Ушаков, А.Е. Шухман, Л.В. Легашев, Д.И. Парфенов; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2018.  
ISBN 978-5-7410-1904-7

В монографии представлены результаты научных исследований и обоснованные разработки, обеспечивающие решение важной задачи развития региональной информационной образовательной среды. Выводы и рекомендации основываются на реальных результатах и могут быть внедрены в процессе реализации образовательной программы высшего образования по направлениям подготовки 02.04.02 Фундаментальная информатика и информационные технологии, 01.04.02 Прикладная математика и информатика, 27.03.03 Системный анализ и управление, 27.04.03 Системный анализ и управление, 27.03.04 Управление в технических системах, 27.04.04 Управление в технических системах.

УДК 004.42  
ББК 32.973.26-018.2

ISBN 978-5-7410-1904-7

© Болодурина И.П.,  
Коннов А.Л.,  
Полежаев П.Н.,  
Ушаков Ю.А.,  
Шухман А.Е.,  
Легашев Л.В.,  
Парфенов Д.И.,  
© ОГУ, 2018

## Содержание

Введение .....	5
1 Мониторинг потребностей образовательных учреждений в образовательных программных продуктах и возможностей использования сервисов отечественных провайдеров облачных услуг для их удовлетворения .....	10
2 Разработка критериев оценки и проведение сравнительного анализа различных технологий реализации виртуальных классов на базе виртуальных машин, серверов терминалов и контейнеров .....	23
3 Создание алгоритмических решений планирования уроков виртуальных классов с учетом пожеланий образовательных учреждений и лицензионных ограничений на используемое программное обеспечение, настройки виртуальной сети виртуальных классов .....	32
3.1 Постановка задачи планирования .....	32
3.2 Имитационная модель регионального центра коллективного доступа .....	32
3.3 Генерация заявок пользователей .....	35
3.4 Составление расписания .....	37
3.5 Эволюционные алгоритмы составления расписания регионального центра коллективного доступа .....	38
3.6 Сравнение эффективности алгоритмов по результатам работы симулятора .....	43
3.7 Алгоритм отслеживания использования лицензий РЦКД .....	46
4 Разработка методики оптимизации ресурсного обеспечения РЦКД .....	58
5 Разработка проекта системы управления регионального центра коллективного доступа к образовательным программным продуктам на базе облачных услуг отечественных провайдеров .....	63
5.1 Описание проекта ресурсного центра коллективного доступа .....	63
5.2 Анализ существующих решений для создания Web-приложений в рамках ресурсного центра коллективного доступа .....	68
6 Разработка технологии защиты информации в РЦКД .....	71
7 Реализация модулей системы управления РЦКД .....	79

7.1 Модуль динамической настройки nginx .....	79
7.2 Модуль методического электронного образовательного центра .....	866
7.3 Модуль подсистемы доступа к виртуальным серверам .....	866
7.4 Модуль запуска и отслеживания виртуальных серверов .....	87
7.5 Модуль обертки над API публичного облачного провайдера .....	90
8 Создание подсистемы совместного доступа к виртуальным урокам, включающим учебные, текстовые и видеоматериалы .....	93
9 Разработка типовых конфигураций виртуальных классов для предмета «Информатика» на основе используемых УМК с учетом необходимого ПО .....	97
10 Развертывание и опытная эксплуатация базовой конфигурации РЦКД .....	101
11 Экспериментальные исследования эффективности, сбалансированности и производительности сервисов РЦКД .....	104
12 Рекомендации по использованию ресурсов РЦКД для органов управления образованием и образовательных организаций .....	124
12.1 Общая информация .....	124
12.2 Методический электронный образовательный центр .....	125
12.3 Ресурсный центр коллективного доступа .....	132
12.3.2 Учитель ресурсного центра коллективного доступа .....	144
12.4 Ученик ресурсного центра коллективного доступа .....	150
Заключение .....	152
Список использованных источников .....	158

## Введение

В настоящее время Оренбургская область характеризуется высоким уровнем информатизации в системе образования. По состоянию на 1 января 2013 года в общеобразовательных учреждениях области насчитывалось 22768 персональных компьютеров, из них 18224 ПК (80%) использовалось в образовательной деятельности. За 2012 год компьютерный парк школ области обновлен на 1,3 тыс. единиц. Отметим, что это составляет 5,7% от общего количества компьютеров. При сохранении порядка финансирования для полного обновления компьютеров потребуется 17,5 лет. При этом в современных условиях срок службы персонального компьютера до морального устаревания составляет 5 лет. Таким образом, при сохранении традиционной ориентации школ на использование собственных вычислительных ресурсов необходимо увеличение финансирования как минимум в 3,5 раза. В современных тяжелых экономических условиях бюджеты образовательных организаций сокращаются, в то время как цены на вычислительную технику значительно выросли в связи с практически 100% импортом комплектующих. Это означает, что период полного обновления компьютеров в ближайшее время будет еще выше.

Другой проблемой информатизации общего образования является недостаточный уровень оснащения школ платными программными продуктами в связи с высокой стоимостью лицензий. Покупка платных лицензий экономически не обоснована, поскольку специализированные программные продукты (например, графические редакторы Adobe) используются лишь небольшую долю от общего времени эксплуатации компьютеров в учебном процессе.

Чаще всего, в системе общего образования предполагается использование открытых бесплатных программных продуктов для операционной системы Linux. Однако администрирование Linux довольно трудоемко, требует высокой квалификации технического персонала. Кроме того, открытые продукты обычно обладают урезанным функционалом или не удовлетворяют приемлемому уровню качества.

Таким образом, традиционная схема использования программных продуктов приводит к риску снижения уровня информатизации общего образования, ухудшению качества подготовки школьников в области современных информационных технологий.

Решением проблемы может стать использование облачных технологий, предполагающих виртуализацию и перенос вычислительных ресурсов в центры обработки данных поставщиков облачных услуг. Облачная система может работать на основе механизма DaaS (Desktop as a Service, рабочий стол в качестве сервиса), направленного на предоставление каждому учащемуся виртуальной машины и виртуального окружения (виртуального рабочего стола) со всем необходимым для обучения установленным программным обеспечением. Доступ к виртуальному окружению осуществляется учащимися удаленно с использованием компьютеров образовательных организаций (которые могут иметь устаревшую конфигурацию) или мобильных устройств (ноутбуков, планшетных компьютеров, смартфонов). Однако в этом случае организация также должна закупить или арендовать необходимое программное обеспечение.

Для сокращения издержек наиболее выгодно организовать совместное использование всеми образовательными организациями региона типовых программных конфигураций, включающих лицензии на платные продукты. При этом виртуальные ресурсы для запуска конфигураций также проще арендовать у отечественных облачных провайдеров. Как показывает обзор публикаций, в настоящее время отсутствуют разработки централизованных систем управления облачными ресурсами, которые учитывали бы специфику работы образовательных организаций – периодичность расписаний, наличие времени начала и окончания использования виртуальных машин во время занятий (с учетом поурочных планов), ограничения плавающих лицензий. Поэтому разрабатываемые решения будут иметь принципиальную новизну в сравнении с существующими разработками.

Для доступа к программному обеспечению планируется использовать удаленный доступ к виртуальным машинам с необходимыми установленными программами. Данные виртуальные машины запускаются в одном или нескольких

публичных облаках российских поставщиков облачных услуг, таких как «OneCloud», «Selectel».

Реализация набора виртуальных машин или виртуальных окружений на базе Windows систем связана с необходимостью предоставлять доступ к удаленному столу нестандартными средствами. Существуют несколько подходов как к организации виртуального класса (выделенные виртуальные машины, терминальный и псевдо-терминальный доступ на общем сервере, контейнеры), так и по доступу к ним (технологии VNC, RDP, SPICE, SDL, NX, TsPlis и прочие). Создание любого количества клонированных машин на базе Linux не вызывает проблем в облачных средах, что нельзя сказать о Windows машинах, которые не устанавливают сетевые параметры в авторежиме,

С целью экономии финансовых ресурсов необходима разработка критериев, по которым будут выбираться определенные сочетания технологий, которые позволят наиболее оптимальным образом использовать облачные ресурсы и снижать стоимость владения. Для различных ситуаций можно использовать различный набор критериев оптимизации и набор возможных состояний системы и тарифных планов с оплатой по нагрузке, по часам, по месяцам, за процессор и память, за операции ввода-вывода и прочее.

Разрабатываемый РЦКД поддерживает два способа проведения занятий:

а) по требованию – когда учитель отправляет заявку на проведения занятия (за 10-15 минут до его начала), указывая желаемый дисковый образ с необходимым ПО, количество пользователей (включая учителя) и продолжительность занятия;

б) по расписанию – координатор образовательной организации составляет шаблон расписания, в котором описываются занятия. Каждое занятие характеризуется: номером урока, днем недели, предметом, классом школьников, физическим классом, в котором оно проводится, дисковым образом и количеством пользователей. В начале каждой недели в 00:00 шаблон расписания переводится в расписание текущей недели, которое при необходимости может быть изменено или уточнено координатором. При заполнении расписания производится контроль лицензий на ПО.

Для проведения каждого типа занятия в РЦКД заранее до начала занятия создается и запускается виртуальная машина. Для ее создания используется указанный дисковый образ с необходимым ПО, а тип виртуальной машины (количество ядер, размер оперативной и дисковой памяти) выбирается исходя из требований к ПО. После запуска виртуальной машины ученики через учителя получают код доступа и могут через браузер, пройдя авторизацию по коду, работать со своими отдельными виртуальными окружениями.

В рамках данного исследования важной научной проблемой является разработка алгоритмов планирования уроков виртуальных классов. Данные алгоритмы должны:

а) составлять расписание работы виртуальных классов с учетом времени начала и окончания отдельных уроков в образовательных организациях и пожеланий преподавателей, ведущих предметы с использованием данных классов;

б) учитывать лицензионные ограничения на ПО, одновременно используемое при проведении уроков в нескольких виртуальных классах различных образовательных организаций (максимальное количество одновременно работающих экземпляров ПО при плавающих лицензиях, а также обычные лицензии);

в) для каждого урока оптимально подбирать конфигурации виртуальных машин под рекомендуемые требования ПО, используемого на данном уроке согласно плану (при этом минимизируются расходы на оплату услуг облачных провайдеров);

г) для реализации составленного расписания резервировать использование вычислительных ресурсов с указанием ПО, необходимого для урока.

Объект исследования – использование программных и методических информационных ресурсов в системе общего образования. Предмет исследования – методы проектирования и реализации центра коллективного доступа к образовательным программным продуктам на базе облачных технологий.

Цель научного исследования – проектирование, разработка и апробация информационного портала для доступа образовательных организаций Оренбургской области к методическим информационным ресурсам и программным продуктам.

Основные задачи:

а) мониторинг потребностей образовательных учреждений в образовательных программных продуктах и возможностей использования сервисов отечественных провайдеров облачных услуг для их удовлетворения;

б) разработка критериев оценки и проведение сравнительного анализа различных технологий реализации виртуальных классов на базе виртуальных машин, серверов терминалов и контейнеров;

в) создание алгоритмических решений планирования уроков виртуальных классов с учетом пожеланий образовательных учреждений и лицензионных ограничений на используемое программное обеспечение, настройки виртуальной сети виртуальных классов;

г) разработка методики оптимизации ресурсного обеспечения РЦКД;

д) разработка проекта системы управления регионального центра коллективного доступа к образовательным программным продуктам на базе облачных услуг отечественных провайдеров;

е) разработка технологии защиты информации в РЦКД;

ж) реализация модулей системы управления РЦКД;

и) создание подсистемы совместного доступа к виртуальным урокам, включающим учебные текстовые и видеоматериалы;

к) разработка типовых конфигураций виртуальных классов для конкретных школьных предметов на основе используемых УМК с учетом необходимого ПО и поурочных планов;

л) развертывание и опытная эксплуатация базовой конфигурации РЦКД;

м) экспериментальные исследования эффективности, сбалансированности и производительности сервисов РЦКД;

н) разработка рекомендаций по использованию ресурсов РЦКД для органов управления образованием и образовательных организаций.

Результаты соответствуют мировому уровню, разработанные информационные системы находятся в стадии опытной эксплуатации и подготовлены к внедрению в региональную информационно-образовательную среду Оренбургской области.

## **1 Мониторинг потребностей образовательных учреждений в образовательных программных продуктах и возможностей использования сервисов отечественных провайдеров облачных услуг для их удовлетворения**

На первом этапе выполнения исследования необходимо было провести мониторинг потребностей образовательных учреждений в образовательных программных продуктах.

Задачи мониторинга: исследовать реальный уровень использования программных продуктов в общеобразовательных организациях, определить используемые в настоящее время аппаратные и программные конфигурации компьютеров учителей и школьников, обобщить характеристики доступа организаций к сети Интернет, уточнить особенности расписания занятий.

Мониторинг проводился в общеобразовательных организациях Оренбургской области в августе-сентябре 2016 г.

В образовательные организации через Министерство образования Оренбургской области была разослана анкета. Ответы вводились в специально созданную форму на основе форм Google Docs, размещенную по короткому адресу <https://goo.gl/bzbE8j>. Внешний вид разработанной формы представлен на рисунке 1.1

Анкета включала в себя семь основных разделов.

Первый раздел включал общую информацию об организации, контактные данные лица, ответственного за информатизацию, и системного администратора, характеристики подключения организации к сети Интернет, общее количество компьютеров и компьютерных классов.

Следующие четыре раздела позволяли собрать информацию о самых худших и самых лучших конфигурациях компьютеров учеников и учителей, в том числе используемых операционных системах, браузерах, установленных процессорах и объеме оперативной памяти.

**Анкета образовательной организации**

Здравствуйте!

Данная анкета создана для сбора информации об использовании средств ИКТ в образовательных организациях общего образования с целью получения статистики, которая поможет при создании регионального центра коллективного доступа к образовательным программам.

При возникновении вопросов по заполнению анкеты или пожеланий, пожалуйста, обратитесь к контактному лицу:  
Шухман Александр Евгеньевич  
телефон: [REDACTED]  
e-mail: [REDACTED]

**\* Обязательно**

**1.1 Название образовательной организации \***

Мой ответ \_\_\_\_\_

**1.2 Ф.И.О. директора \***

Мой ответ \_\_\_\_\_

Рисунок 1.1 – Анкета образовательной организации

Собранные сведения необходимы для того, чтобы определить потребности в использовании виртуальных машин, а также минимальные технические требования для использования РЦКД.

Шестой раздел разработан для сбора информации об используемом в учебном процессе образовательном программном обеспечении. Предлагалось указать реально применяемые офисные приложения, графические редакторы, средства трехмерной графики, видео редакторы, среды программирования, развивающие программы, а также прочие программные продукты.

Последний седьмой раздел давал возможность уточнить особенности информатизации учебного процесса, определить для каких дисциплин используются школьные компьютеры, выявить особенности составления школьных расписаний.

Всего за два месяца получено 730 анкет, в том числе 8 – повторных. В тестировании приняло участие 722 общеобразовательных организации из всех районов и городов Оренбургской области, в том числе 457 средних общеобразовательных школ, 19 гимназий, 20 лицеев, 201 основная общеобразовательная школа, 13 начальных общеобразовательных школ, 7 детских садов, 5 организаций дополнительного образования.

Суммарное количество компьютеров, реально используемых учениками, по собранным данным составило 8820 шт. В некоторых школах указывалось количество ученических компьютеров от 60 до 105, при этом лишь отмечалось наличие только двух компьютерных классов, поэтому эта оценка несколько завышена.

Количество компьютерных классов составило 826. В большинстве школ имеется один компьютерный класс, в 99 – два класса, в 13 школах – три класса и в одной школе (ЗАО Комаровский) – 4 компьютерных класса. Среднее количество компьютеров в компьютерном классе – 9.

Диаграмма, представляющая среднее количество компьютеров в образовательной организации, представлена на рисунке 1.2.

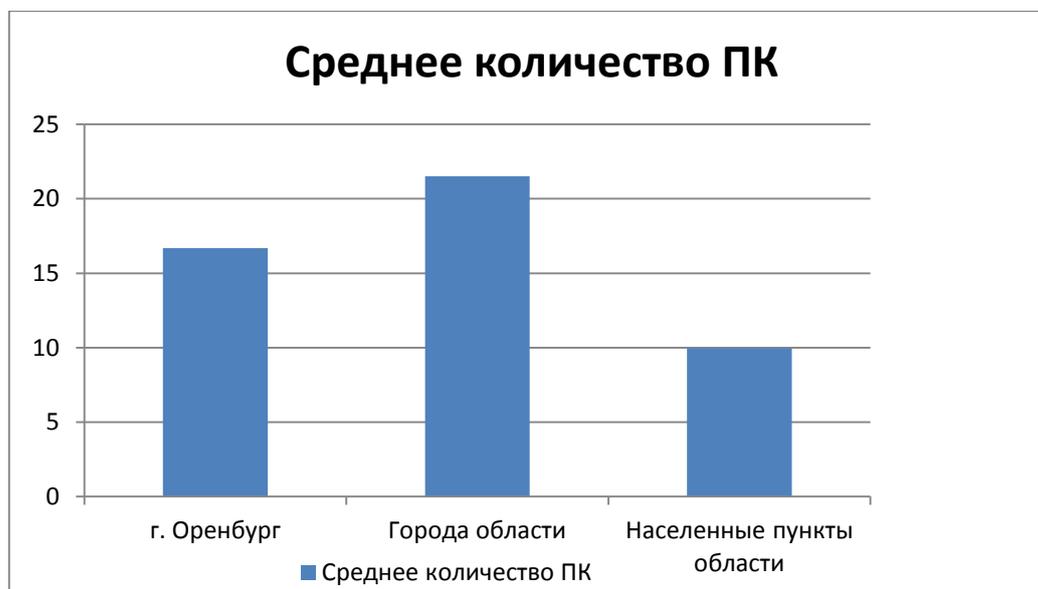


Рисунок 1.2 – Среднее количество компьютеров в одной организации

Из диаграммы видно, что, в среднем, в городских школах количество компьютеров выше, чем в сельских. В типовой городской школе два компьютерных класса, в сельской – только один. В школах г. Оренбурга в среднем используется 16 компьютеров, что близко к среднему значению по области. Более информативно было бы исследование количества компьютеров на одного ученика, но анкета не содержала вопросов о численности школьников.

Наиболее значимо разница между сельскими и городскими школами проявляется при исследовании скорости доступа в сеть Интернет. В анкете предлагалось указать реальную скорость чтения данных, полученную на основе тестирования с помощью сайта [www.speedtest.net](http://www.speedtest.net). В 243 школах (47%) скорость менее 1 Мбит/с, что не дает возможность нормально использовать современные информационно-образовательные облачные ресурсы. Выше 10 Мбит/с доступ только в 15% школ.

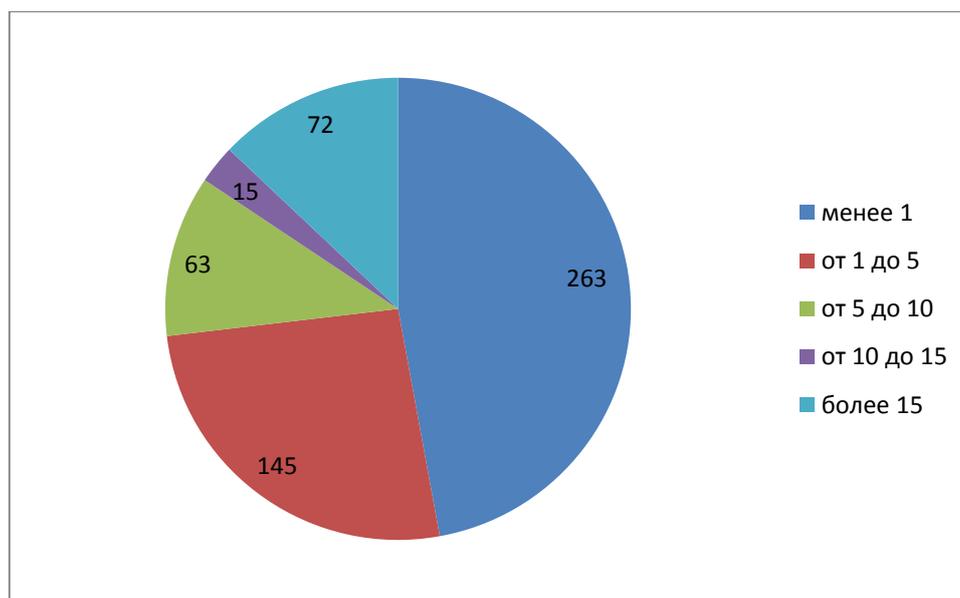


Рисунок 1.3 – Скорость доступа в сеть Интернет (сельские районы)

Диаграмма на рисунке 1.4 отражает скорость доступа в городах области кроме г. Оренбурга. Видно, что ситуация лучше, чем в области, ниже 1 Мбит/с скорость только в 13% школ, однако 54% организаций имеют подключение со скоростью менее

5 Мбит/с (в основном ADSL). 22% школ используют высокоскоростное соединение не менее 10 Мбит/с.

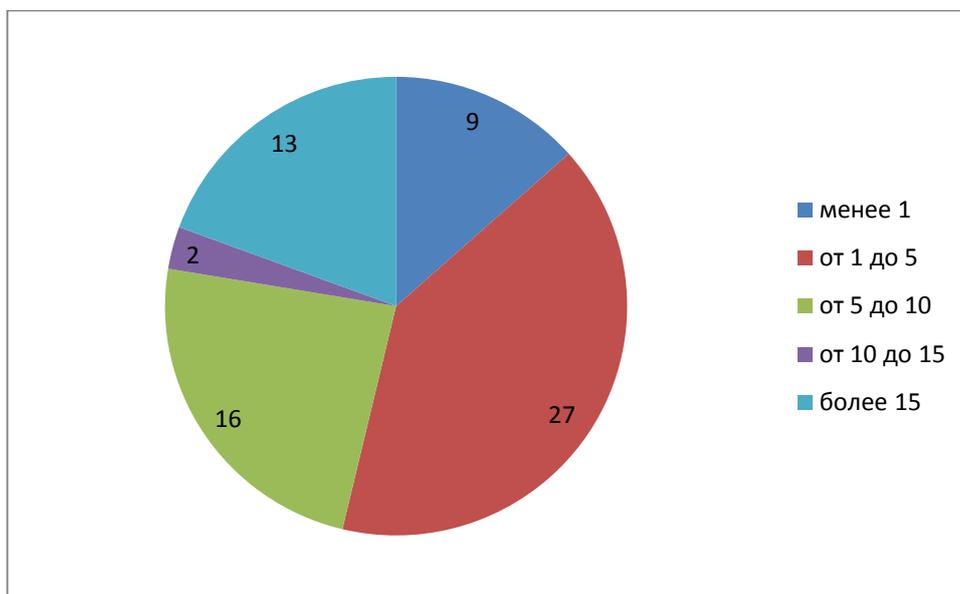


Рисунок 1.4 – Скорость доступа в сеть Интернет (города кроме Оренбурга)

Диаграмма на рисунке 1.5 иллюстрирует скорость доступа в Оренбурге, где есть возможность широкого выбора провайдеров Интернет услуг. 52% школ г. Оренбурга имеют возможность высокоскоростного доступа (максимум до 100 Мбит/с). Ниже 1 Мбит/с скорость только в 9% школ, 37% организаций имеют подключение со скоростью менее 5 Мбит/с.

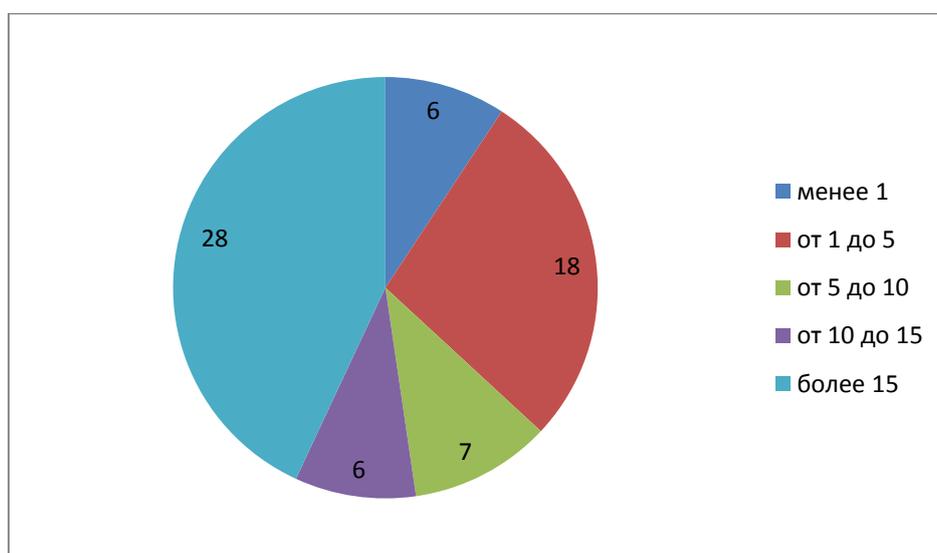


Рисунок 1.5 – Скорость доступа в сеть Интернет (город Оренбург)

Диаграмма на рисунке 1.6 показывает распределение образовательных организаций по провайдерам Интернет. К сожалению, возможность выбора провайдера имеется только в крупных населенных пунктах, поэтому безоговорочным лидером по количеству подключенных школ является компания «Ростелеком» (594 организации), на втором месте «Уфанет» (43 организации), остальные провайдеры имеют единичные подключения – не более 20 организаций.



Рисунок 1.6 – Скорость доступа в сеть Интернет (город Оренбург)

Диаграммы на рисунке 1.7 показывают процент использования различных операционных систем на компьютерах в худшей и лучшей конфигурациях.



Рисунок 1.7 – Использование различных операционных систем

На учебных компьютерах ситуация абсолютно аналогична. Видно, что ОС Linux используется редко, причем, чем лучше конфигурация компьютера, тем реже применяется ОС Linux. Опыт использования программного обеспечения под Linux в обучении показывает, что основные проблемы связаны не с функционалом программ, а с недостаточно высокой квалификацией учителей и, особенно, системных администраторов. Широкое использование Linux возможно только при централизованном администрировании компьютерной техники. Использование РЦКД дает хорошую возможность начать использовать программные продукты для Linux в виртуальных окружениях, которые уже заранее грамотно настроены. Без этого ни о каком импортозамещении говорить невозможно.

Диаграммы на рисунке 1.8 показывают процент использования различных версий Windows на компьютерах в худшей и лучшей конфигурациях.

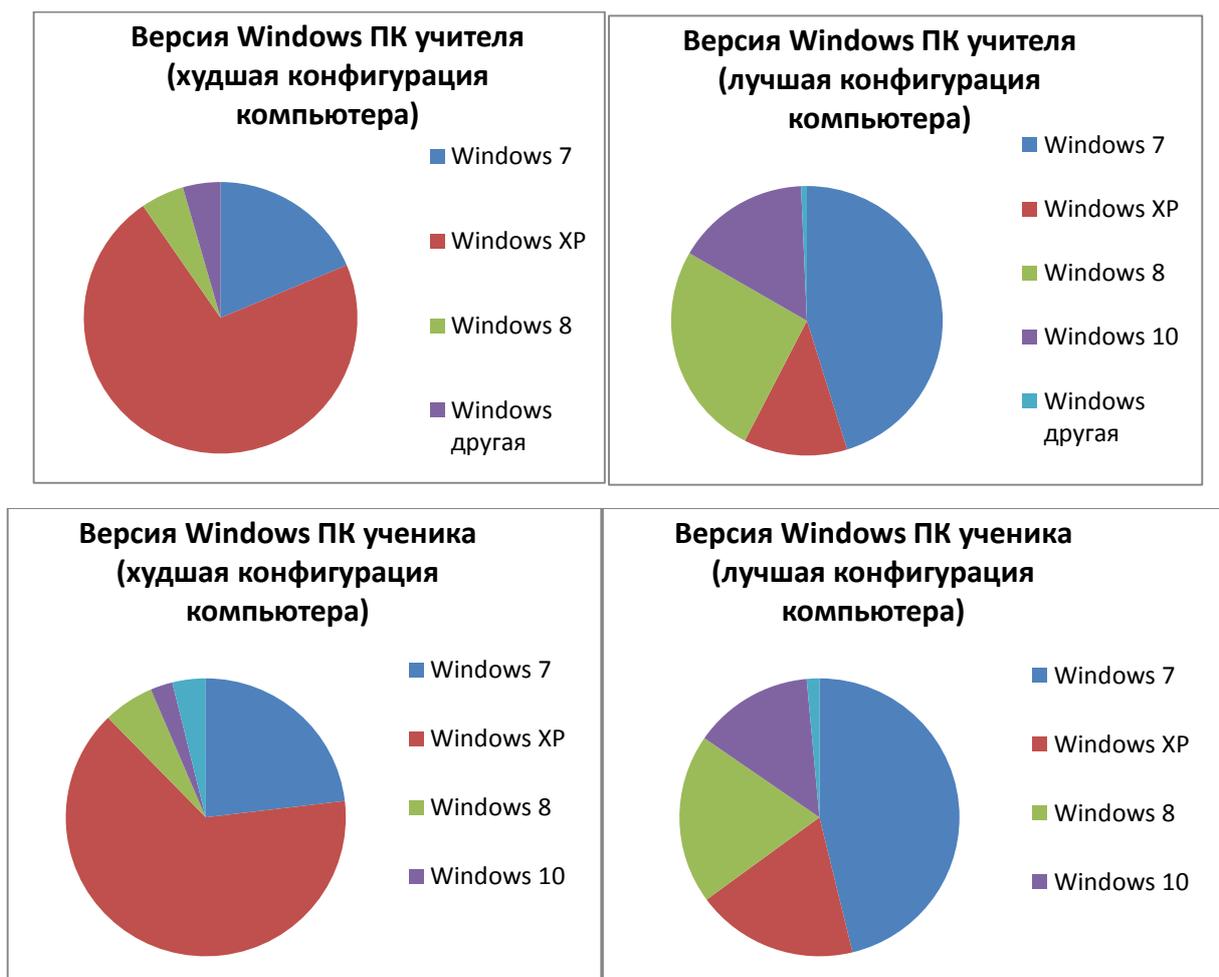


Рисунок 1.8 – Использование различных версий ОС Windows

Анализ результатов показывает, что давно неподдерживаемые официально операционные системы Windows 2000 и Windows XP в настоящее время используются в 353 школах на компьютерах учеников и 426 школах на компьютерах учителей. Если учесть, что наиболее популярной системой в школах является Windows 7, для которой не выпускаются критические обновления с 2015 года, то можно сделать вывод, что устаревшие операционные системы в том или ином виде применяются в 89% школ, а новейшая Windows 10 – только в 15% образовательных организаций и то на новых компьютерах.

Использование устаревших версий Windows опасно тем, что на них не выпускаются критические обновления, что может привести к заражению вредоносными программами. Кроме того, многое современное ПО не работает на этих версиях, например, Microsoft Office 2016, который предоставляется школам со значительной скидкой.

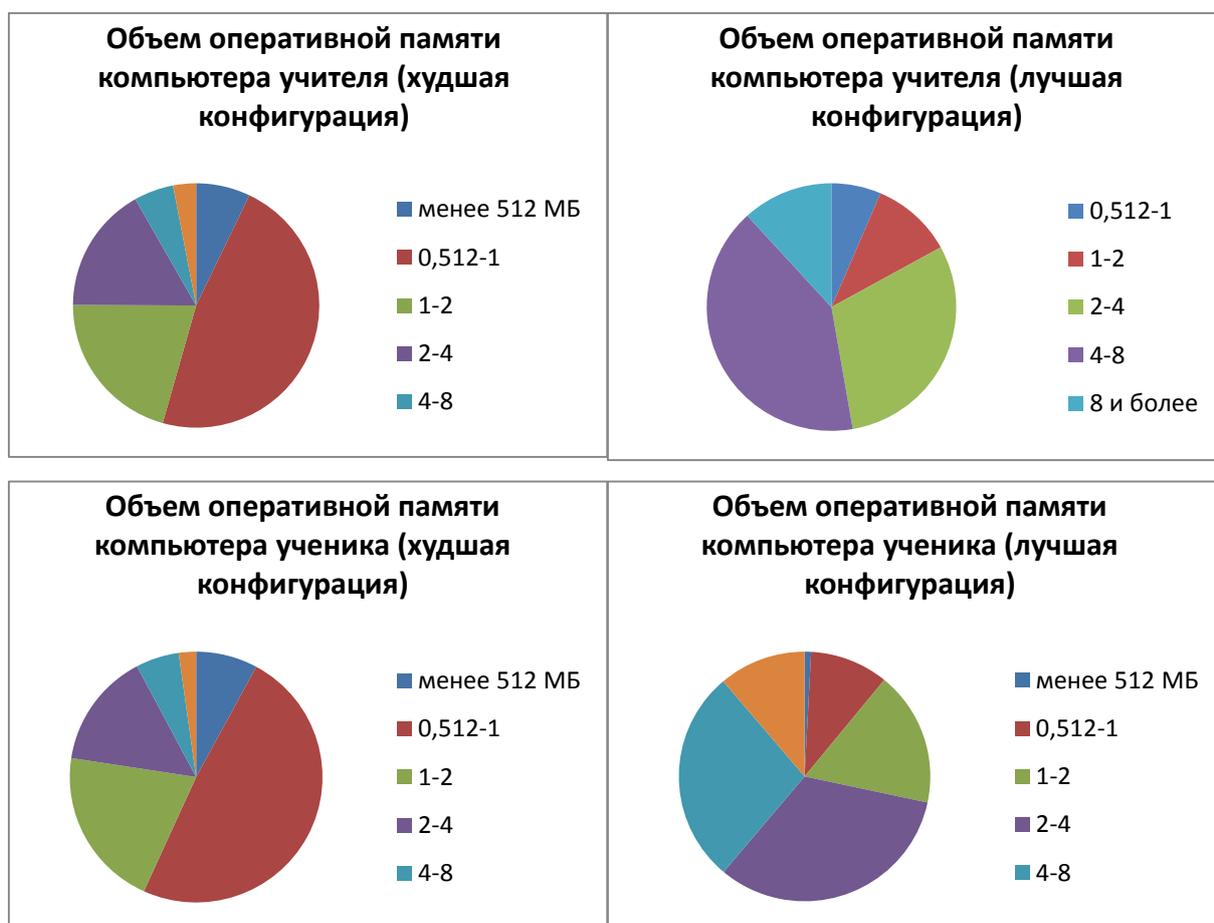


Рисунок 1.9 – Объем оперативной памяти

Основная причина использования устаревших версий Windows – устаревшие конфигурации компьютеров, поскольку достаточно долгое время переход на Windows 10 для владельцев предыдущих версий был бесплатным.

Диаграммы на рисунке 1.9 показывают объем оперативной памяти на компьютерах в худшей и лучшей конфигурациях. К сожалению, в 6 школах еще используются компьютеры с 128 Мб ОЗУ, 26 школах – с 256 Мб ОЗУ, 351 школе – с 512 Мб ОЗУ. Объем памяти менее 1 Гб ОЗУ абсолютно неприемлем в настоящее время, потому что не позволяет использовать современные программы и информационные ресурсы. Только в 41 школе минимальная конфигурация компьютеров имеет 3 Гб ОЗУ и более, что приемлемо для нормальной работы.

Диаграмма на рисунке 1.10 показывает распределение используемых офисных пакетов.

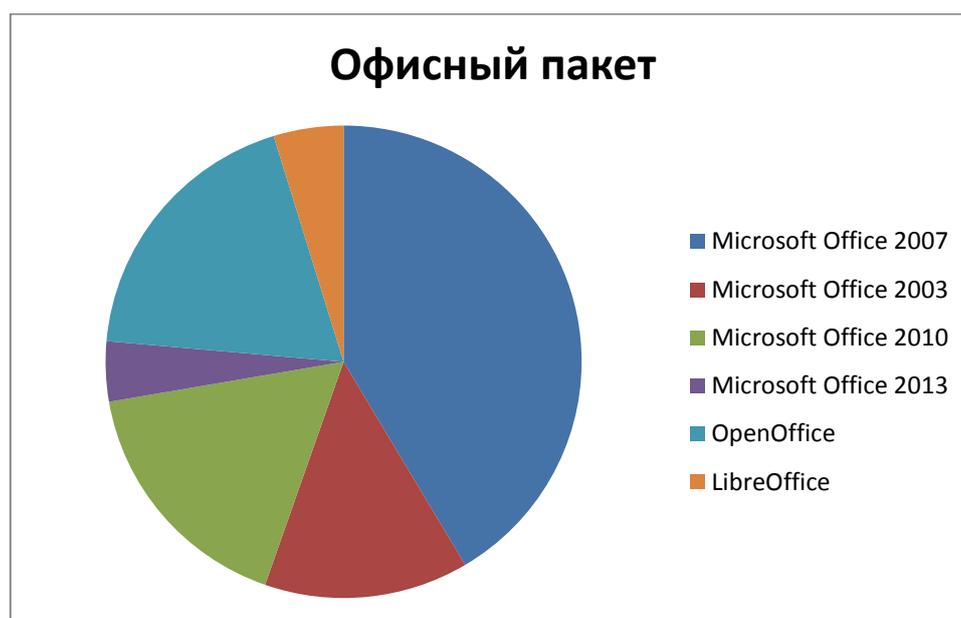


Рисунок 1.10 – Распределение офисных пакетов

В большинстве школ используются устаревшие версии Microsoft Office, что связано с использованием устаревших версий Windows. Однако также параллельно в 32% школ применяются бесплатные офисные пакеты OpenOffice и LibreOffice.

Диаграмма на рисунке 1.11 показывает распределение используемых графических пакетов.

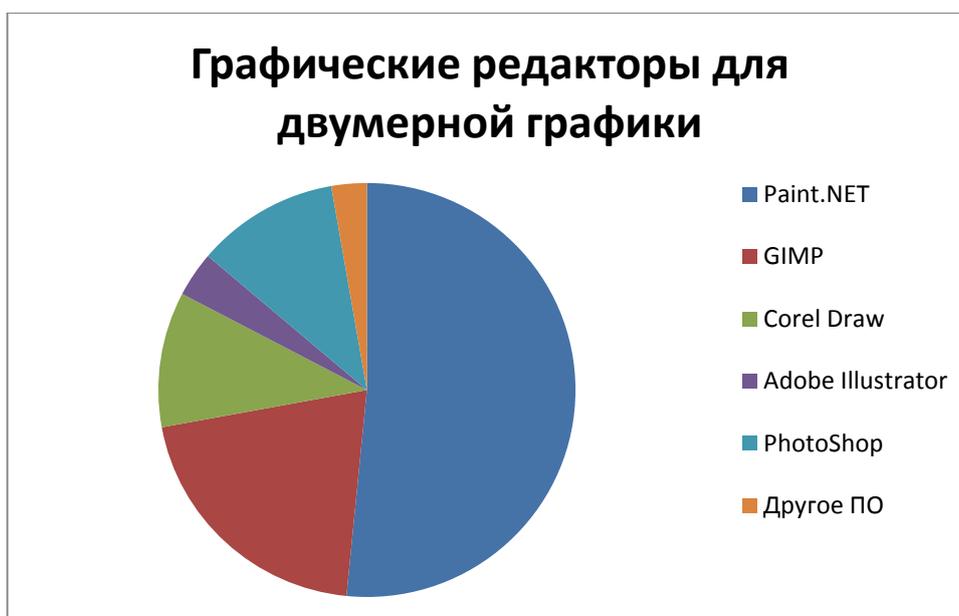


Рисунок 1.11 – Распределение графических пакетов

В основном в школах используются бесплатные пакеты GIMP и Paint.NET. Это связано с высокой стоимостью платных продуктов от Corel и Adobe. В то же время 108 школ используют Corel Draw и 143 школы – редакторы Adobe Photoshop или Illustrator. Отметим, что для образовательных учреждений одна бессрочная лицензия CorelDRAW Graphics Suite X8 Classroom License (16 компьютеров) стоит около 30 тыс. рублей, одна годовая лицензия Adobe Photoshop на 1 компьютер стоит около 10 тыс. рублей. Таким образом расходы на обновление используемых графических пакетов в Оренбургской области в год могут составлять до 15 млн. рублей, при том, что большинство школ не имеют возможности их использовать.

Пакеты трехмерной графики используются только в 16% школ, несмотря на то, что их изучение входит в программы по информатике. 32 школы используют 3ds Max, 63 школы – Blender, 6 школ – пакет Компас 3D. Все эти пакеты могут использоваться в школах бесплатно, поэтому основной причиной столь малого распространения, следует считать проблемы с устаревшими конфигурациями, а также недостаточный уровень учителей и системных администраторов.

В качестве редакторов видео в большинстве школ применяются встроенный в Windows Movie Maker (Киностудия в последних версиях). В качестве языков программирования, в основном, используются Pascal (бесплатные среды Free

Pascal/Lazarus или Pascal ABC), а также Visual Studio. Остальные программные продукты в анкетах встречаются только эпизодически.

Таким образом, большинство школ до сих пор использует устаревшие компьютеры, версии операционных систем и офисных пакетов. Это сильно ограничивает возможности применения современных программных продуктов. Скорость подключения к сети Интернет в городах составила в среднем 8 Мбит/с, однако в сельских районах много образовательных организаций подключены на скорости до 1Мбит/с.

Для исследования возможностей облачных провайдеров рассмотрим опыт применения облачных технологий для предоставления удаленного доступа к виртуальным окружениям и ПО образовательным организациям.

Национальный исследовательский университет ЮУрГУ разработал облачную образовательную платформу «Персональный виртуальный компьютер» [1]. Данный проект реализует механизм EaaS (Education as a Service) и представляет собой электронный образовательный ресурс, доступ к которому осуществляется удаленно с устройства пользователя. Доступ к виртуальным рабочим столам происходит на базе готового решения посредством платного импортного программного обеспечения Citrix Xen Desktop, имеющего значительную стоимость и не позволяющего контролировать ограничения плавающих лицензий.

В статье [2] Bo Wang и HongYu Xing описаны приложения облачных вычислений в системе образования Китая. Реализация механизма SaaS (Software as a Service), при использовании облачных приложений Google Apps и Zoho Office, позволяет учащимся получать доступ к офисным пакетам, библиотекам и облачным ресурсам с персонального компьютера, на котором установлена операционная система Linux и Web-браузер Firefox. Указанное программное обеспечение относится к открытому ПО, следовательно, образовательной организации не надо беспокоиться об установке платного программного обеспечения, обеспечивать поддержку и обновление ПО, а также своевременно закупать дорогостоящее оборудование. Предложенная система не позволяет получать удаленный доступ к обычным

настольным приложениям, не адаптированным под облачные модели, и их лицензионные ограничения.

Статья [3] Shinichiro Kibe, Teruaki Koyama и Minoru Uehara предлагает два вида реализации схемы DaaS (Desktop as a Service) – удалённый рабочий стол для одного или группы пользователей. Разработанные авторами статьи продукты поддерживают различные операционные системы (CentOS, Ubuntu, Windows и MacOSX). При увеличении количества пользователей возрастает нагрузка на облачную систему. Для решения этой проблемы авторы предложили ограничения по ролям для удаленных рабочих столов и сессионные ограничения на использование таких рабочих столов. В дальнейшем авторы планируют описать подходы, которые позволят снизить стоимость использования образовательного облака при увеличении числа активных пользователей. В отличие от решений авторов статьи, в рамках данной монографии предлагаются разработки по эффективному распределению нагрузки пользователей на виртуальные машины, а также поддержка плавающих лицензий.

Совершенно другой подход предложен Justin Cappos, Ivan Beschastnikh, Arvind Krishnamurthy и Tom Anderson в статье [4]. Здесь описывается разработанная облачная вычислительная образовательная платформа Seattle. Данная платформа зависит от ресурсов, которыми поделился пользователь программного обеспечения. Он имеет возможность установить Seattle на свой персональный компьютер и использовать программное обеспечение этой платформы, задействуя часть вычислительных ресурсов компьютера с возможностью выполнения программ других пользователей в песочницах.

Идея использования обычных компьютеров, вместо высокопроизводительных серверов, для построения облачных образовательных систем представлена в работах [5-6]. Основной недостаток [5] – низкая производительность, в том числе подсистемы ввода/вывода, связанная с доступом к дисковым образам виртуальных машин.

В работе [7] предлагается использовать ресурсы провайдеров облачных услуг для создания образовательных облаков нескольких высших учебных заведений. Авторы рассматривают только услуги IaaS, PaaS и SaaS, оставив за рамками

предоставление услуги рабочего стола в качестве сервиса – DaaS-услуг (Desktop as a Service), кроме того, они не описывают никаких деталей возможной реализации.

Таким образом, в настоящее время отсутствуют разработки централизованных систем управления облачными ресурсами, которые учитывали бы специфику работы образовательных организаций – периодичность расписаний, наличие времени начала и окончания использования виртуальных машин во время занятий (с учетом поурочных планов), ограничения плавающих лицензий. Поэтому разрабатываемые решения будут иметь принципиальную новизну в сравнении с существующими разработками.

## **2 Разработка критериев оценки и проведение сравнительного анализа различных технологий реализации виртуальных классов на базе виртуальных машин, серверов терминалов и контейнеров**

Виртуальное рабочее место как явление возникло еще на заре многопользовательских операционных систем. Комплексы с удаленными терминалами давали возможность работать на едином сервере нескольким людям параллельно. Мейнфреймы использовали технологию эмуляции терминала (например vt100) и позволяли в текстовом диалоговом и интерактивном режиме работать с операционной системой. С появлением сетей передачи информации эмуляция терминала через сетевое соединение постепенно вытеснила физическое подключение к серверу. Протоколы telnet, rlogin, ssh, а затем и графические эмуляторы терминала заняли всю нишу виртуальных рабочих мест. Например бездисковые терминалы на Novel Netware сначала обеспечивали работу в DOS на удаленном сервере, затем начали поддерживать Windows начиная с версии 3.11 в режиме графического интерфейса.

Самым популярным на сегодняшний день решением удаленного доступа к графической среде являются в Windows протоколы RDP и VNC, в Linux – X, VNC, SPICE. Реализация же именно удаленного рабочего места в этих системах принципиально отличается. Примем за удаленное рабочее место систему, позволяющую в графическом режиме работать с удаленными вычислительными ресурсами напрямую в интерфейсе операционной системы с доступом к личной информации на выделенном месте накопителя. Существует довольно много реализаций рабочего места, как средствами операционной системы, так и сторонними программами.

Рассмотрим технологии реализации рабочих мест в системе Linux. Эта операционная система изначально создавалась как многопользовательская, поэтому все механизмы разграничения доступа, выделения ресурсов, выделение отдельного терминала, хранилища реализованы на уровне операционной системы. Поэтому вопрос именно реализации рабочего места не стоит, поскольку каждый пользователь

в Linux и так работает в эмуляторе терминала, независимо локально или удаленно. Различные протоколы доступа используются для оптимизации различных дополнительных возможностей при удаленном выполнении программ. Например, SDL позволяет напрямую работать с удаленным графическим адаптером и используется в системах CAD и для игр, а SPICE позволяет получить доступ с любых устройств просто через браузер. Даже встроенная оболочка X позволяет использовать сетевое соединение для удаленной работы через любые средства связи, в том числе через эмуляторы терминала на SSH. На рисунке 2.1 показан общий принцип работы удаленного рабочего места на Linux. Все обеспечение рабочего места производится средствами ОС, протоколы используются только для передачи изображения.

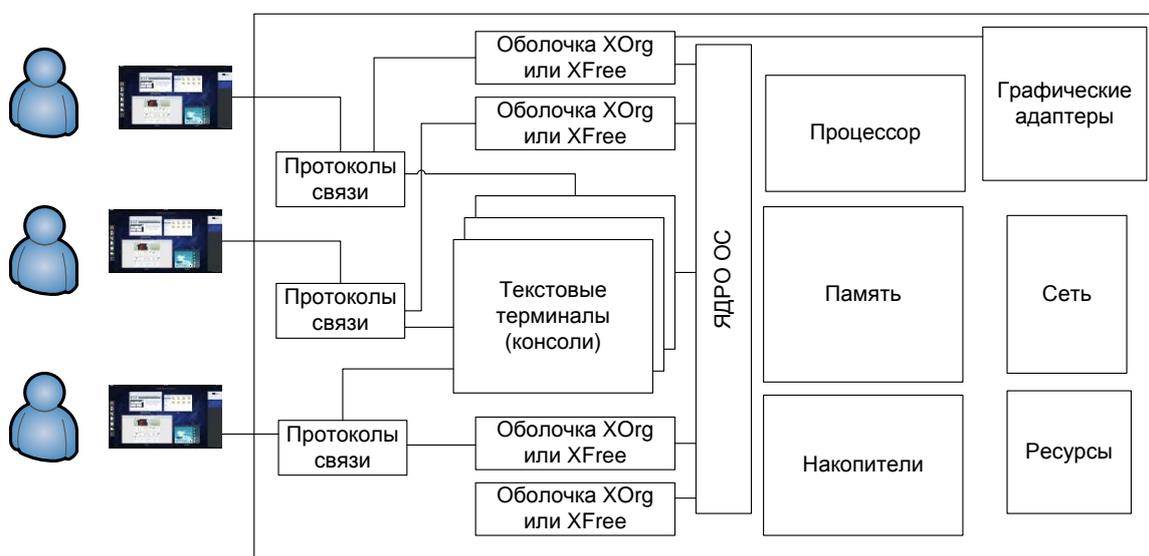


Рисунок 2.1 – Схема организации удаленного рабочего места на Linux.

Поскольку все взаимодействие с ОС происходит встроенными средствами и через ядро, появляется возможность виртуализации и накопителя, и сетевых ресурсов (например WebDAV). В таблице 1 показаны основные средства организации доступа в Linux систему с графическим интерфейсом

Таблица 2.1 – Способы удаленного графического доступа Linux

Название	Особенности	Недостатки
1	2	3
X	Можно установить на любой Linux и подключаться с любого Linux. Сразу работает на большинстве дистрибутивов. Предоставляет множество возможностей проброса устройств, звука, видео.	Для подключения с Windows систем нужна или прямая двусторонняя связь или проброс порта (VPN, SSH). Довольно сложно без дополнительных средств получить доступ через браузер.
VNC	Самый популярный способ получения удаленного доступа к Linux системам. Существует возможность доступа через браузер, с мобильных устройств.	В базовом протоколе нет возможности проброса на клиентскую сторону устройств, дополнительных потоков (есть в некоторых проприетарных реализациях).
SDL	Создавалась для отображения динамических графических потоков, поэтому очень быстро работает с графическим интерфейсом. Существует возможность доступа через браузер.	Слабая поддержка в Windows и в мобильных устройствах
SPICE	Создавалась, как альтернатива VNC, но более функциональная. Встроенная возможность работы через браузер, есть проброс устройств, дисков, интеграция с X и множеством сред виртуализации.	Нет поддержки в Windows
NX	Семейство протоколов для доступа к удаленному столу. Создавалась как альтернатива RDP, есть поддержка работы через браузер. Отличается хорошей работой с графикой, работает с многими средами виртуализации.	Большинство реализаций проприетарны, платны, требуют только совместимых клиентов
RDP	Один из самых распространенных и функциональных протоколов. Поддерживается всеми ОС, есть реализация для браузера, проброс портов, устройств, шифрование, буфер обмена.	Реализация в Window требует лицензирования служб терминалов, сторонние реализации также платные
XDMC	Протокол прямого доступа к удаленному столу Linux для тонких клиентов.	Нет шифрования, поддержки браузеров
TeamViewer AMMY admin ChromeRD AnyDesk и подобные	Множество протоколов доступа через сервер-посредник. Используются для временного удаленного доступа, могут пробрасывать порты и устройства. Есть поддержка работы через браузер	Требуют доступа к серверу посредника, наиболее функциональные версии платны для коммерческого использования

Продолжение таблицы 2.1

1	2	3
Parallels	Представляет собой сервер терминалов с поддержкой удаленного стола. Есть поддержка работы в браузере, проброс ресурсов. Ориентирован на виртуальные среды	Не под все ОС есть клиент

Для Windows решений возможность организации выделенного рабочего места упирается в ограничения ОС и лицензирования. Клиентские версии ОС не поддерживают реальное многопользовательское использование одновременно работающих терминалов. Существует возможность реализации только сервера приложений на базе Citrix, Parallels и подобных систем. Серверные версии Windows позволяют реализовывать полнофункциональную инфраструктуру средствами сервера терминалов или сторонними средствами. Поскольку в Windows Server отдельно реализуется система удаленного доступа и отображения, и отдельно система организации виртуального окружения, рассмотрим подробнее эти механизмы.

Реализация виртуального окружения может быть выполнена в Windows Server несколькими способами:

а) создание перемещаемых профилей. Самый распространенный способ, требует сервер, на котором будут храниться профили. Каждому пользователю задается путь до хранимого профиля, при входе профиль копируется на сервер, при выходе копируется обратно в хранилище. Требуется работающей инфраструктурой на Windows, как правило, еще требуется работающего каталога Active Directory. Не позволяет под одной учетной записью заходить на один сервер с отдельным окружением – окружение станет общим на несколько сессий;

б) создание виртуального диска – при каждом входе создается или подключается VHD диск, где хранится профиль пользователя. Диск может существовать локально или на сервере. Также не позволяет под одной учетной записью заходить на один сервер с отдельным окружением – окружение станет общим на несколько сессий;

в) перенаправление папок – самый гибкий способ, позволяет при входе делать ссылки на папки окружения – рабочий стол, документы, параметры. Папки могут быть как локальными, так и сетевыми, перенаправлять можно уже после входа; Можно перенаправлять централизованно через групповые политики или скриптами;

г) создание временного окружения из профиля по умолчанию. Способ, при котором окружение создается при входе копированием профиля Default. После выхода все изменения теряются, этот способ используется для гостевого входа.

Каждый описанный способ можно сочетать с другими, добиваясь гибкости и решения поставленных задач. Особенностью является довольно сложная настройка каждого из способов для работы в виртуальном облачном окружении.

Для реализации удаленного доступа используются множество как встроенных, так и сторонних средств. В таблице 2.2 показаны наиболее распространенные способы. Доступ через VNC/TeamViewer/Radmin и подобные не рассматривается, поскольку он несовместим с созданием виртуального окружения(виртуального входа в систему)

Таблица 2.2 – Удаленный доступ в Windows Server для виртуальных терминалов

Название	Особенности	Недостатки
1	2	3
Windows RDS	Встроенная служба сервера терминалов.	Негибкая настройка, слабая автоматизация, необходимость держать сервер лицензирования
Citrix VDI	Самая популярная альтернатива встроенным службам RDS, есть множество уникальных возможностей	Стоимость
TSplus Server	Альтернатива для Citrix XenDesktop со встроенной возможностью распределения нагрузки, работы через браузер, создание псевдонимов аккаунтов и управления сессиями.	Некоторые недоработки в области перевода на русский язык.
Parellels RAS / 2X Computing	Практически полный аналог Citrix, реализует как виртуальные рабочие столы, так и публикацию отдельных приложений.	Ограниченная возможность работы в браузере
Ulteo OVD	Система виртуализации рабочих столов. Есть версия с открытым кодом, работает как на Window, так и на Linux. Реализован как отдельный сервер плюс серверы терминалов	Ограниченный функционал на Windows, нет возможности управлять сессиями, требуется дополнительный сервер

Исходя из анализа продуктов, реализующих функционал удаленного рабочего места, были разработаны следующие критерии выбора:

- а) возможность организации удаленного рабочего места;
- б) встроенная возможность работы через браузер;
- в) возможность распределять нагрузку между несколькими серверами;
- г) стоимость подключения 1 пользователя/сервера к рабочему столу;
- д) возможность переноса лицензий между серверами (плавающая лицензия);
- е) автоматизация настройки новых серверов.

В соответствии с этими критериями были выбраны продукты для Linux и Windows. Условные обозначения: «+» - полное соответствие, «-» - несоответствие, «±<sup>1</sup>» - неполное соответствие, подробности в примечании после таблицы «n\а» - не применимо.

Таблица 2.3 – Критерии выбора продукта

Продукт	Критерий 1	Критерий 2	Критерий 3	Критерий 4	Критерий 5	Критерий 6
1	2	3	4	5	6	7
Linux						
X	+	-	-	0	n\а	± <sup>4</sup>
VNC	+	± <sup>1</sup>	-	0	n\а	± <sup>4</sup>
SDL	+	± <sup>2</sup>	-	0	n\а	± <sup>4</sup>
SPICE	+	+	-	0	n\а	± <sup>4</sup>
NX	+	+	+	350000р/серв./год	-	+
RDP	+	± <sup>3</sup>	± <sup>3</sup>	0	n\а	± <sup>4</sup>
XDMC	+	-	-	0	n\а	± <sup>4</sup>
TeamViewer / AMMY admin / ChromeRD / AnyDesk и подобные	+	+	-	0-60000р/польз.	+	-
Parallels RAS/2X	+	+	+	6990/польз./год	+	+

Продолжение таблицы 2.3

1	2	3	4	5	6	7
Windows Server						
Windows RDS	+	± <sup>5</sup>	+	RDP CALs От 5000р/пол ьз +нужна лицензия SPLA или CALs	+	± <sup>9</sup>
Citrix VDI	+	+	+	От 6000р/пол ьз.	+	+
TSplus Server	+	+	+	От 800руб/по льз.	+	+
Parallels RAS / 2X Computing	+	+	+	6990/поль з./год	+	± <sup>8</sup>
Ulteo OVD	± <sup>6</sup>	+	-	от 7800/серв ер	± <sup>7</sup>	+

Примечания:

- а) с использованием стороннего ПО поVNC;
- б) с помощью дополнительных компонентов;
- в) нужны компоненты HTML5RDP, через них же и распределяется нагрузка;
- г) стандартная автоматизация Linux скриптами;
- д) только через некоторые браузеры;
- е) только ограниченный функционал;
- ж) лицензии есть двух видов, и они не смешиваются в лицензировании;
- з) ограниченная возможность автоматизации;
- и) весьма сложная автоматизация на powershell.

В результате для Linux приоритетными продуктами стали Spice и Parallels, для Windows – TSplus и Citrix.

Следующая возможность реализации виртуальных классов – выделенные виртуальные машины на пользователя. В этом случае не требуется реализовывать виртуальное окружение, пользователю предоставляется полноценная виртуальная машина для работы. Этот подход популярен для руководителей, программистов,

администраторов поскольку там требуются альтернативные конфигурации и наборы ПО, а также ресурсы для работы. Этот же подход является более затратным, поскольку виртуализация всей ОС требует высоких накладных расходов. В случае с Windows Server это до 1Гб ОЗУ на виртуальную машину. Для массовых классов образовательных учреждений данный подход мало применим.

Для реализации виртуализации самих систем на данный момент существует несколько видов виртуализации. Возможны следующие решения:

а) полноценная виртуализация IaaS – виртуальные машины Hyper-V, VMware, KVM, XEN;

б) на основе паравиртуализации – OpenVZ, Jail, WindowsNanoServer;

в) на основе изоляции сервисов – Docker, WindowsServer Container, Linux Namespaces.

Существующие экспериментальные исследования по использованию контейнеров и виртуальных машин показывают, что издержки виртуализации являются существенными. Контейнеры имеют вычислительную производительность, скорость доступа к оперативной и дисковой памяти практически эквивалентные показателям физического сервера. Использование виртуальных машин снижает вычислительную производительность на 4 % - 5 %, приводит к замедлению на 30 % операций при работе с оперативной памятью (трансляция адресов), на 30 % - 50 % при доступе к файлам на жестких дисках, на 40 % - 60 % снижается пропускная способность сети.

С другой стороны, анализ степени изолированности контейнеров и виртуальных машин показывает, что виртуальные машины обеспечивают более честное распределение физических ресурсов, чем контейнеры, а также более высокий уровень безопасности и надежности (гипервизоры виртуальных машин имеют меньший размер кодовой базы и лучше протестированы).

Контейнеры в сравнении с виртуальными машинами запускаются значительно быстрее, т.к. нет необходимости загружать ОС, достаточно только запустить необходимые приложения.

Поскольку необходимо предоставлять пользователям виртуальный рабочий стол, необходима виртуализация с поддержкой графической среды. Для Windows систем это только полноценная виртуализация IaaS, такая, как Hyper-V, VMware, KVM. Для промежуточной инфраструктуры тип виртуализации можно выбрать как паравиртуализацию, так и IaaS виртуализацию, поскольку работа происходит на Linux с серверными приложениями. Контейнеры целесообразно использовать для запуска множества экземпляров служб на нескольких серверах, тогда как в монографии предполагается использование выделенных виртуальных серверов из-за большой нагрузки.

### **3 Создание алгоритмических решений планирования уроков виртуальных классов с учетом пожеланий образовательных учреждений и лицензионных ограничений на используемое программное обеспечение, настройки виртуальной сети виртуальных классов**

#### **3.1 Постановка задачи планирования**

Описанные во введении ограничения на работу регионального центра коллективного доступа ставят задачу оптимизации глобального расписания.

Расписание регионального центра составляется непрерывно во времени по мере поступления заявок пользователей. Если запрос очередного пользователя удовлетворить не удастся, то ему предлагаются другие варианты предоставления виртуальных машин. Если отсутствует возможность удовлетворить пожелания пользователя, его заявка отклоняется.

Составление расписания в условиях ограничений представляет собой нетривиальную задачу, для которой необходимо найти решение. Для решения задачи требуется разработать имитационную модель регионального центра коллективного доступа, описывающую структуру связей основных компонентов и механизмы их взаимодействия. На основе имитационной модели разрабатывается симулятор РЦКД, который генерирует запросы конечных пользователей и составляет оптимальное расписание в соответствии с ограничениями. Для оценки адекватности модели и эффективности оптимизационных алгоритмов разработана система критериев. После проверки работы системы на симуляторе РЦКД, она будет развернута на испытательном сегменте РЦКД в Оренбургском государственном университете.

#### **3.2 Имитационная модель регионального центра коллективного доступа**

Для создания симулятора регионального центра коллективного доступа необходимо разработать имитационную модель РЦКД. Цель моделирования состоит

в составлении оптимального расписания, которое удовлетворяет всем требованиям текущих пользователей РЦКД.

Объектом моделирования является расписание центра, которое реализуется в виде совокупности записей о назначении каждой виртуальной машины пользователя на определенный сервер РЦКД в указанный промежуток времени.

Для описания и воспроизведения основных этапов работы регионального центра коллективного доступа с целью проведения экспериментов по предоставлению услуги DaaS разработана имитационная модель, реализованная в виде симулятора РЦКД на языке программирования C++. Симулятор регионального центра позволяет сравнить алгоритмы составления расписания с помощью экспериментальных исследований.

Имитационная модель РЦКД является непрерывной во времени. Региональный центр коллективного доступа к образовательным программным продуктам  $CRD = (ScheduleTemplates, Software, Requests)$  представляет собой совокупность шаблонов расписаний  $ScheduleTemplates = \{ScheduleTemplate_i\}$ , доступного программного обеспечения  $Software = \{Software_j\}$  и заявок пользователей  $Requests = \{Request_k\}$ .

Шаблон  $ScheduleTemplate_i = (Title_i, Length_i, Intervals_i)$  расписания содержит наименование шаблона  $Title_i$ , продолжительность занятия  $Length_i$  в минутах и множество временных интервалов занятий  $Intervals_i = \{[x_i, x_i + Length_i] | x_i \in \mathbb{N}\}$ . Мощность множества  $Intervals_i$  показывает общее количество временных интервалов (слотов) в текущем расписании.

Каждое программное обеспечение характеризуется вектором  $Software_j = (OS_j, Title_j, LicenseCount_j)$ , для которого указывается поддерживаемая операционная система  $OS_j \in \{ "Windows", "Linux" \}$ , наименование программного обеспечения  $Title_j$  и имеющееся количество лицензий  $LicenseCount_j$ . Если количество лицензий неограниченно, то  $LicenseCount_j = \infty$ .

Пользователи, число которых выражается величиной  $UsersCount$ , оставляют по одной заявке на Web-портале РЦКД. Каждая заявка представляет собой кортеж  $Request_k = (ScheduleTemplate_k, VMCount_k, s_k, t_{arrive.k}, t_k, w_k, Status_k)$  и содержит: используемый шаблон расписания пользователя  $ScheduleTemplate_k \in ScheduleTemplates$ , количество экземпляров виртуальных машин  $VMCount_k$ , подмножество запрашиваемого ПО  $s_k \subseteq Software$  из списка программного обеспечения, доступного для РЦКД, время появления заявки  $t_{arrive.k}$ , набор индексов временных слотов  $t_k = \{I_{1k}, I_{2k}, \dots, I_{r_k k}\}$ , где  $I_{lk} \in Intervals_k$ , ( $l = \overline{1, r_k}$ , слоты принадлежат шаблону заявок), весовой коэффициент пользователя  $w_k$ , а также статус размещения заявки пользователя в расписании  $Status_k = \{-1, 0, 1\}$ .

Значение  $Status_k = 1$  показывает, что заявка размещена в расписании в указанное пользователем время. Иначе, если планировщику не удалось разместить заявку пользователя в указанное им время, и он подобрал другие временные слоты, то статус заявки принимает значение  $Status_k = 0$ . Значение  $Status_k = -1$  соответствует случаю, когда заявку пользователя вообще не удалось разместить в текущем расписании.

Расписание РЦКД представляет собой совокупность пар статусов заявок пользователей  $Status_k$  и назначенных временных слотов  $AssignedIntervals_k$ :  $S = \{(Status_k, AssignedIntervals_k)\}$ . Причем, если  $Status_k = -1$ , то  $AssignedIntervals_k = \emptyset$ .

Генерация заявок и составление расписания регионального центра коллективного доступа происходит на основе имитационной модели [6, 7], представленной на рисунке 3.1. Рассмотрим основные этапы функционирования данной модели.

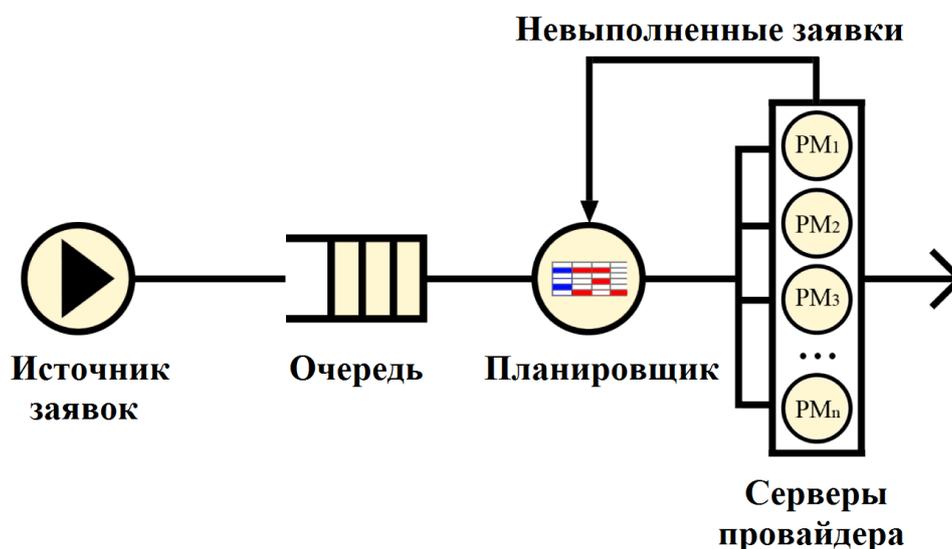


Рисунок 3.1 – Имитационная модель РЦКД

### 3.3 Генерация заявок пользователей

Запросы конечных пользователей на предоставление услуг DaaS генерируются случайным образом. Для формирования закона распределения при генерации заявок необходимо учитывать особенности использования программного обеспечения в учебном процессе.

Обозначим множество всех возможных дисковых образов с установленной операционной системой через  $Im = \{Im_1, Im_2, \dots, Im_k\}$ . Множество всех пакетов программного обеспечения обозначим через  $SP = \{SP_1, SP_2, \dots, SP_N\}$ . Элементы множества  $S$  представляют собой готовые пакеты программного обеспечения, которые заранее формируются в региональном центре: «Пакет программирования», «Пакет работы с 2D графикой», «Пакет работы с 3D графикой», «Пакет работы с офисными приложениями», и т.д. По желанию, пользователь может сформировать собственный пакет программного обеспечения, выбрав доступные позиции из списка. Для обработки такой ситуации последний элемент множества включает в себя весь список доступного облачному центру программного обеспечения. Данный выбор обусловлен тем, что и школьная, и университетская образовательные программы обязательно включают в себя изучение всех дисциплин в соответствии с учебным планом, для которых используется компьютерное оборудование. Множество шаблонов

конфигураций виртуальных машин обозначим через  $FL = \{FL_1, FL_2, \dots, FL_M\}$ . Отообразим все множества и связи между ними на рисунке 3.2.

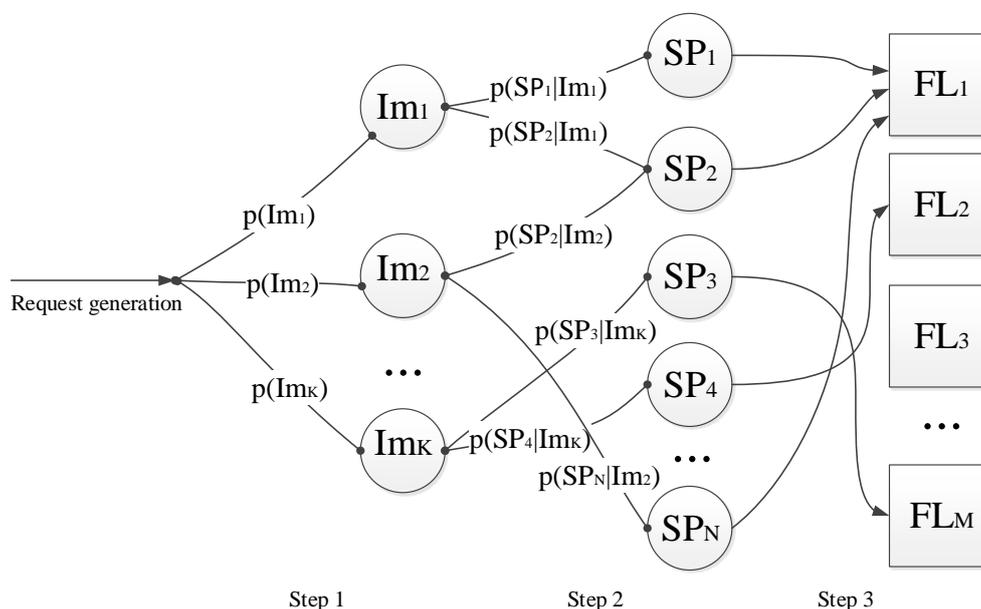


Рисунок 3.2 – Генерация заявок

Каждый пользователь формирует одну заявку, поэтому количество заявок `RequestsCount` совпадает с количеством пользователей `UsersCount`. Шаблон расписания  $ScheduleTemplate_k$ , набор индексов временных слотов  $t_k$  и количество требуемого программного обеспечения  $|s_k|$  представляют собой равномерно распределенные случайные величины. Само множество  $s_k$  формируется путем случайной перестановки элементов множества `Software` с последующим выбором первых  $|s_k|$  элементов. Время поступления заявок пользователей  $t_{arrive.k}$  подчиняется распределению Пуассона. Соответственно, интервал  $\Delta t_k = t_{arrive.k} - t_{arrive.k-1}$  между появлением двух заявок имеет экспоненциальное распределение и рассчитывается по формуле  $\Delta t_k = -\frac{1}{\lambda \ln(r)}$ , где  $\lambda$  представляет собой интенсивность потока, а  $r$  – равномерно распределенное число из  $[0, 1]$ . Границы временных интервалов  $Interval_k$  и время прибытия заявки  $t_{arrive.k}$  измеряются в минутах. В случае составления расписания РЦКД на сутки, отрезок  $[480, 1080]$  соответствует временному интервалу

8:00–18:00. Количество экземпляров виртуальной машины пользователя  $VMCount_k$  распределено равномерно на отрезке  $[5, 15]$ , который соответствует минимальному и максимальному количеству учеников/студентов в классе/группе.

Пример сгенерированной заявки:

$\{ScheduleTemplate_1\}$ ,  $\{VMCount: 10\}$ ,  $\{Software_5, Software_9\}$ ,  $\{t_{arrive}: 240\}$ ,  $\{t: 2, 3, 6\}$ ,  $\{w: 0.5\}$ ,  $\{Status: -1\}$ ).

Пример программного обеспечения, доступного для использования в региональном центре коллективного доступа:

$Software_5 = (\{OS: \text{“Linux”}\}, \{Title: \text{“LibreOffice”}\}, \{LicenseCount: \infty\})$ ,

$Software_9 = (\{OS: \text{“Windows”}\}, \{Title: \text{“Blender”}\}, \{LicenseCount: 24\})$ .

Пример шаблона расписания:

$ScheduleTemplate_1 = (\text{“Физико-математический лицей”}, 45 \text{ минут}, \{[8 \cdot 60 + 30, 9 \cdot 60 + 15], [9 \cdot 60 + 25, 10 \cdot 60 + 10], [10 \cdot 60 + 25, 11 \cdot 60 + 10], [11 \cdot 60 + 25, 12 \cdot 60 + 10], [12 \cdot 60 + 25, 13 \cdot 60 + 10], [13 \cdot 60 + 15, 14 \cdot 60], [14 \cdot 60 + 10, 14 \cdot 60 + 55], [15 \cdot 60 + 5, 15 \cdot 60 + 50]\})$ .

### 3.4 Составление расписания

Генерируемые заявки попадают в очередь, которая функционирует по принципу FCFS (First-Come, First-Served), т.е. заявки обрабатываются в порядке поступления их в очередь. Планировщик извлекает из очереди сгенерированную заявку и размещает её в расписании РЦКД, при этом соблюдаются следующие условия:

1. Максимизируется функция пожеланий пользователей:

$$F(S) \rightarrow \max,$$

$$F(S) = \sum_{k=1}^{UsersCount} w_k [\alpha \cdot PreferredTS_k + \beta \cdot notPreferredTS_k - \gamma \cdot unsTS_k], \quad (1)$$

где  $\alpha$  – коэффициент поощрения за размещение виртуальной машины в указанные пользователем временные промежутки;  $PreferredTS_k$  – количество таких машин в  $k$ -й заявке;  $\beta$  – коэффициент поощрения за размещение машины в промежуток времени, не указанный пользователем;  $notPreferredTS_k$  – количество таких машин в  $k$ -й заявке;  $\gamma$  – штрафной коэффициент за невозможность размещения виртуальной машины пользователя в указанное время;  $unsTS_k$  – количество таких машин в  $k$ -й заявке.

2. Для всех единиц программного обеспечения и всех временных интервалов пользователей выполняются ограничения на максимальное число лицензий программного обеспечения в каждую минуту времени:

$$\forall j = 1..|Software|, \forall l = [480, 1080]:$$

$$AssignedVMsC(S, j, l) \leq LicenseCount_j, \quad (2)$$

где  $AssignedVMsC(S, j, l)$  – общее число виртуальных машин, выполняющих  $j$ -е программное обеспечение в  $l$ -й интервал времени согласно расписанию  $S$ .

Задача составления расписания является NP-полной, для её решения в симуляторе РЦКД реализованы алгоритм имитации отжига и генетический алгоритм.

### **3.5 Эволюционные алгоритмы составления расписания регионального центра коллективного доступа**

Метод имитации отжига является стохастическим, он позволяет найти оптимальное решение поставленной задачи за разумное время, избегая при этом попадания целевой функции в локальные максимумы. Ниже приведен псевдокод для нахождения максимума целевой функции методом имитации отжига:

1.  $temp_{max} = 100$ ; // задание максимальной температуры.
2.  $S_{initial} = CreateInitialSolution(CRD)$ ; // построение начального расписания.
3.  $S_{best} = S_{initial}$ ; // исходное расписание принимается за оптимальное расписание.

4.  $\text{temp}_{\text{curr}} = \text{temp}_{\text{max}}$ ; // в качестве текущей температуры процесса выставляется максимальная температура.

5.  $S_{\text{current}} = S_{\text{initial}}$ ; // в качестве текущего расписания устанавливается начальное расписание.

6. while( $\text{temp}_{\text{curr}} > 0.05$ )

6.1 if( $\text{random}[0,1] < 0.01$ )

6.1.1  $k_{\text{selected}} = \text{RandomSelect}(\text{Requests}, S_{\text{current}})$ ; // случайно выбирается заявка, для которой  $\text{Status}_k = 1$ .

6.1.2  $S_{\text{current}} = \text{RemoveTimeSlots}(S_{\text{current}}, k_{\text{selected}})$ ; // снятие виртуальных машин пользователя  $k_{\text{selected}}$ .

6.1.3  $\text{Status}_{k_{\text{selected}}} = -1$ ; // смена статуса заявки пользователя.

6.2 for( $k = 1, k < \text{UsersCount}, k++$ )

6.2.1 if ( $\text{Status}_k = -1$ )  $S_{\text{current}} = \text{SelectTimeSlotsFor}(S_{\text{current}}, \text{Request}_k)$ ; // выбор новых временных слотов.

6.3 if ( $F(S_{\text{current}}) > F(S_{\text{best}})$ ) // выполнение проверки и обновление расписания.

6.3.1  $S_{\text{best}} = S_{\text{current}}$ ;

6.4 else if  $\text{Probability}(S_{\text{current}}, S_{\text{best}}, \text{temp}_{\text{curr}}) > \text{random}[0,1]$

6.4.1  $S_{\text{best}} = S_{\text{current}}$ ; // текущее состояние системы принимается в качестве оптимального состояния.

6.5  $\text{temp}_{\text{curr}} = \text{temp}_{\text{curr}} \cdot 0.9$ ; // уменьшение температуры на фиксированный шаг.

В алгоритме первоначальный вариант расписания РЦКД обозначен в качестве  $S_{\text{initial}}$ . Для составления этого расписания на шаге 2 используется метод Round-Robin, который случайным образом выбирает пользователя, и по возможности размещает в расписании все виртуальные машины из его заявки. Выбор пользователя и размещение его машин осуществляется циклично до тех пор, пока не будут выполнены все заявки (очередь заявок пуста), либо перестанут выполняться

ограничения на число лицензий платного программного обеспечения (будет превышено максимально допустимое значение).

После получения нового варианта расписания (шаги 6.1 – 6.2) подсчитывается значение целевой функции  $F(S)$ . Если оно больше лучшего, новое состояние системы принимается в качестве оптимального. В противном случае – новое состояние системы также принимается в качестве оптимального состояния, если вероятность:

$$\text{Probability}(S_{\text{current}}, S_{\text{best}}, \text{temp}_{\text{curr}}) = e^{-\frac{|F(S_{\text{current}}) - F(S_{\text{best}})|}{\text{temp}_{\text{curr}}}} \quad (3)$$

больше вероятности  $\text{random} [0, 1]$ , представляющей собой равномерно распределенное случайное число на отрезке  $[0, 1]$ .

Температура процесса  $\text{temp}_{\text{curr}}$  регулирует количество итераций алгоритма. При достижении температурой значения 0.05 алгоритм завершает свою работу, при этом состояние системы  $S_{\text{best}}$  соответствует оптимальному варианту расписания РЦКД.

Каждая особь в генетическом алгоритме представляет собой один из вариантов составления расписания РЦКД. Размер начальной популяции `InitialPopulation` соответствует количеству заявок регионального центра `RequestsCount`. Для обеспечения разнообразия исходных вариантов расписания, применим алгоритм Round-Robin к случайным перестановкам текущих пользователей РЦКД в очереди.

Ниже приведен псевдокод генетического алгоритма составления оптимального расписания:

1. `InitialPopulation = CreateInitialPopulation(CRD);` // создание начальной популяции.
2. `Population = InitialPopulation;` // в качестве текущей популяции принимается начальная популяция.
3. `for (e = 1, e < EpochsNumber, e ++)` // цикл по всем эпохам алгоритма.

3.1  $CrossoverPopulation = \emptyset$ ; // в начале результаты операции скрещивания пустые.

3.2 for ( $i = 1, i < UsersCount, i ++$ ) // цикл по всем вариантам расписаний.

3.2.1  $Parent_{i1}, Parent_{2i} = random(Population)$  // случайным образом выбирается пара родителей из текущей популяции.

3.2.2 for ( $k = 1, k < UsersCount, k ++$ ) // цикл по всем пользователям.

3.2.2.1 if ( $Status_k^{Parent_{i1} = 1} \ \&\& \ Status_k^{Parent_{2i} = 1}$ ) попарно сравниваются статусы заявок пользователей родительских вариантов расписаний.

3.2.2.1.1  $Status_k^{Child_i} = 1$ ; // дочернее расписание наследует заявки, размещенные в указанные пользователями временные сроки.

3.2.2.1.2  $AssignedIntervals_k^{Child_i} = AssignedIntervals_k^{Parent_{i1}}$ ; // в качестве назначенных интервалов выбираем интервалы одного из родителей, например,  $Parent_{i1}$ .

3.2.2.2 else  $Child_i = SelectTimeSlotsFor(Child_i, k)$ ; // выбор новых временных слотов для k-й заявки в дочернем расписании.

3.2.3  $CrossoverPopulation = CrossoverPopulation \cup \{Child_i\}$ ; // добавление нового дочернего расписания в  $CrossoverPopulation$ .

3.3  $NewPopulation = Population \cup CrossoverPopulation$ ; // полученная в результате операции скрещивания популяция присоединяется к текущей популяции и имеет размер  $2 \cdot RequestsCount$ .

3.4 for ( $l = 1, l < 2 \cdot RequestsCount, l ++$ )

3.4.1  $Mutated_l = random(NewPopulation)$ ; // равновероятно выбирается один из вариантов расписания.

3.4.2 for ( $k = 1, k < UsersCount, k ++$ )

3.4.2.1 if ( $Status_k^{Mutated_l} = -1$ )  $Mutated_l = SelectTimeSlotsFor(Mutated_l, k)$ ;

// выбор новых временных слотов.

3.4.3 if (random[0,1] < 0.01)

3.4.3.1  $k_{\text{selected}} = \text{RandomSelect}(\text{Requests});$  // случайный выбор заявки, для которой  $\text{Status}_k = 1.$

3.4.3.2  $\text{Mutated}_1 = \text{RemoveTimeSlots}(\text{Mutated}_1, k_{\text{selected}});$  // снятие виртуальных машин пользователя с расписания.

3.4.3.3  $\text{Status}_{k_{\text{selected}}} = -1;$  // смена статуса заявки пользователя.

3.5 foreach(S in NewPopulation)

3.5.1  $C[i] = F(S_i);$  // для каждого варианта расписания подсчитывается значение целевой функции.

3.5.2  $\text{Num}[i] = i;$  // запоминается индекс особи.

3.6  $\text{Sort}(C, \text{Num});$  // сортируется значение целевой функции по невозрастанию, и одновременно сортируются индексы в Num.

3.7  $\text{Population} = \emptyset;$

3.8 for (k = 1, k < UsersCount, k ++)

3.8.1  $\text{Population} = \text{Population} \cup \{\text{NewPopulation}_{\text{Num}[k]}\}$  в следующее поколение записывается выборка, состоящая из особей с максимальным значением целевой функции  $F(S).$

Работа алгоритма завершается, если было достигнуто максимальное число эпох функционирования генетического алгоритма EpochsNumber .

### 3.6 Сравнение эффективности алгоритмов по результатам работы симулятора

Для оценки эффективности работы алгоритма имитации отжига и генетического алгоритма в симуляторе регионального центра коллективного доступа был реализован алгоритм, строящий оптимальное расписание путем полного перебора. Комбинация размещения всех заявок пользователей в расписании представляет собой декартово произведение сочетаний без повторений из всех временных слотов шаблона расписания по количеству временных слотов пользователя. Время работы переборного алгоритма быстро возрастает с увеличением значения UsersCount. Уже для значения UsersCount = 5 время в среднем составляет около 20 минут. Коэффициенты поощрения и штрафа подбирались эмпирически в ходе тестирования работы симулятора, были выбраны следующие значения:  $\alpha = 0.7$ ,  $\beta = 0.4$ ,  $\gamma = 0.9$ .

Для количества заявок из интервала [2,5] и количества виртуальных машин каждого пользователя из интервала [5,15] существует возможность реализации алгоритма перебора, не требующей особых вычислительных мощностей. Проведено 50 экспериментов по генерации и размещению заявок (рисунок 3.3).

Результаты размещения заявок алгоритмом имитации отжига и алгоритмом перебора совпадают. Генетический алгоритм в 48 случаях из 50 разместил виртуальные машины так же, как и переборный алгоритм.

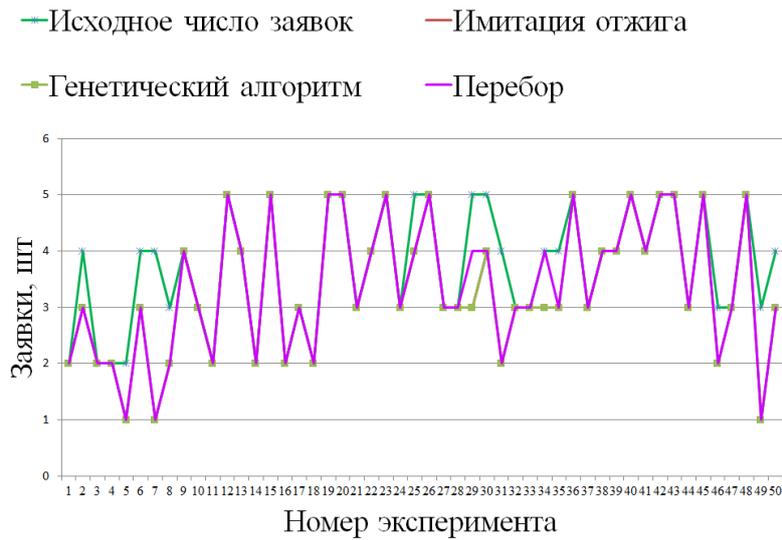


Рисунок 3.3 – Количество исходных и размещенных заявок

Следующие 150 экспериментов проведены для большего числа пользователей, что соответствует реальной нагрузке РЦКД. Результаты представлены на рисунке 3.4. Число пользователей варьировалось в промежутке [10,40]. Каждая точка на оси X представляет собой отрезок усреднения значения целевой функции в результате 10 экспериментов.

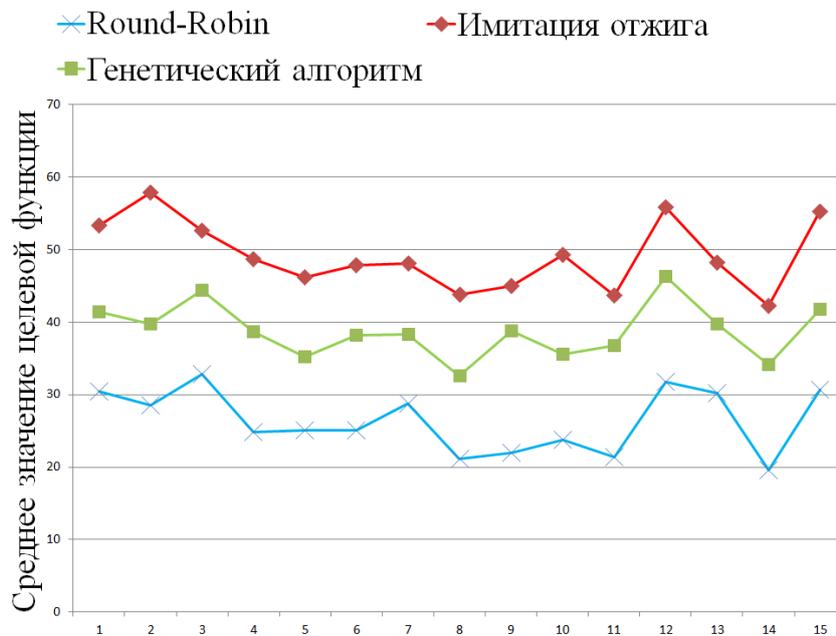


Рисунок 3.4 – Среднее значение целевой функции для каждого алгоритма составления расписания

Алгоритм имитации отжига разместил 89.7 % от общего числа сгенерированных заявок, генетический алгоритм – 86.5 %.

Для описанных ранее 150 экспериментов также было измерено время выполнения алгоритмов, результаты представлены на рисунке 3.5.

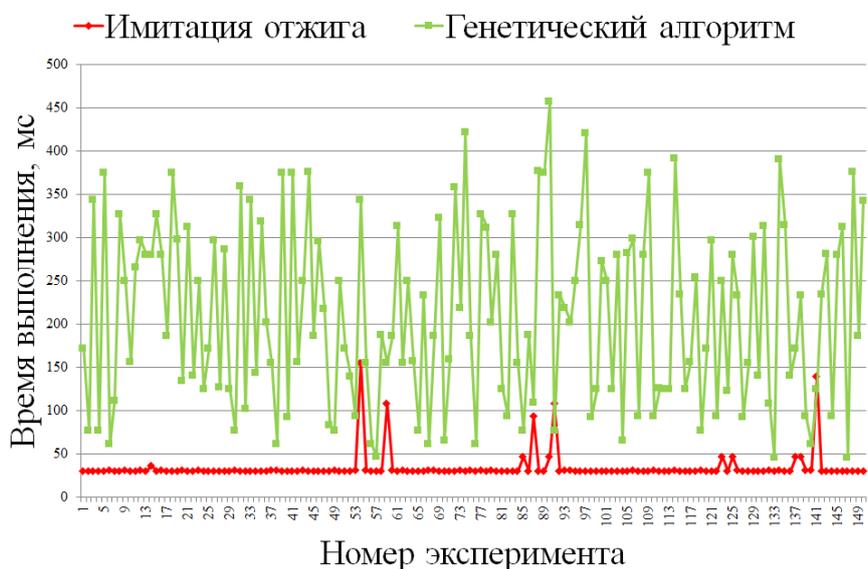


Рисунок 3.5 – Время выполнения работы алгоритмов

Среднее время выполнения алгоритма имитации отжига составило 34.91 мс, генетического алгоритма – 210.04 мс.

Также проведены исследования зависимости количества размещенных заявок от исходного числа поступивших заявок для различных значений максимального количества доступных лицензий программного обеспечения. На рисунке 3.6 представлены графики зависимости количества размещенных заявок пользователей от количества доступного ПО для 100 исходных заявок в РЦКД.

Анализ графиков (рисунок 3.6) показывает, что при увеличении максимального количества лицензий программного обеспечения, доступного региональному центру коллективного доступа, количество размещенных заявок пользователей приближается к исходному числу заявок.

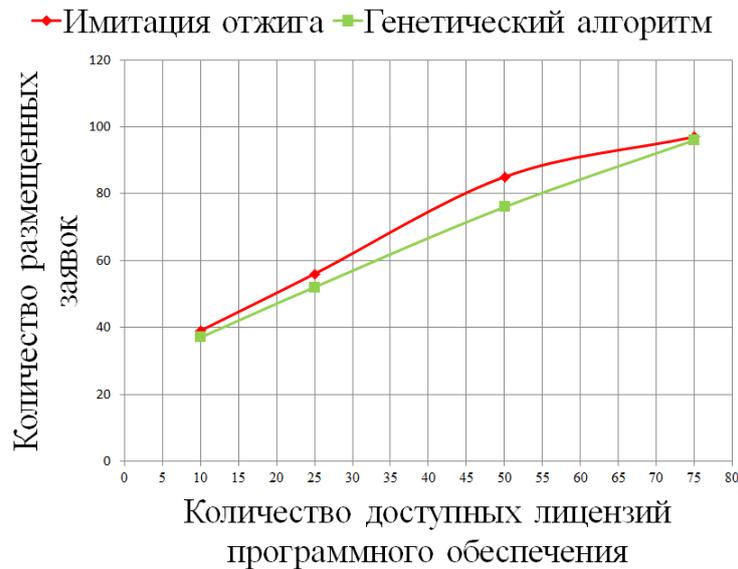


Рисунок 3.6 – Графики зависимости числа размещенных заявок от количества доступного программного обеспечения для 100 исходных заявок

### 3.7 Алгоритм отслеживания использования лицензий РЦКД

Так как количество лицензий на программное обеспечение может быть ограничено, необходимо строго отслеживать наличие возможности поставить данный запрос в расписание. Для решения этой задачи был выбран один из возможных вариантов алгоритма, исследована его асимптотическая сложность и затраты по памяти, а также дополнительно были рассмотрены некоторые способы оптимизации.

На каждую виртуальную машину может быть установлено дополнительное программное обеспечение из списка Software размера SoftwareCount. Значение LicenseCount<sub>i</sub> показывает общее количество лицензий для i-го ( $i \in [1, \text{SoftwareCount}]$ ) программного обеспечения. Если это значение равно -1, то количество лицензий не ограничено.

Пользователи оставляют заявки на Web-портале регионального центра коллективного доступа. Заявки могут быть двух видов:

- 1) одиночная заявка по требованию:

$$\text{ReqOnDemand}_i = (S_i, \text{VMCount}_i, t\text{Start}_i, t\text{End}_i),$$

где  $S_i$  – список необходимого программного обеспечения,  $VMCount_i$  – количество виртуальных машин,  $[tStart_i, tEnd_i]$  – время работы виртуальных машин, с точностью до минут;

2) заявка на еженедельные занятия. В течение четверти или семестра учащиеся должны изучить определённые темы, которые следуют друг за другом и известны заранее, при этом занятия проводятся еженедельно в установленное время и дни недели. Благодаря этому есть возможность составить одну заявку, передав весь список необходимого программного обеспечения для каждой темы, расписание и продолжительность уроков и количество необходимых виртуальных машин в учебном заведении. Кроме того, необходимо сообщить информацию о дате первого урока, так как предмет может начинаться с 1 сентября, второго семестра или даже с середины четверти. Сама заявка будет иметь вид:

$$ReqWeekly_i = (ListS_i, VMCount_i, len_i, tStart_i, Sch_i),$$

где  $ListS_i$  – список необходимого программного обеспечения на каждую тему,  $VMCount_i$  – количество виртуальных машин,  $len_i$  – продолжительность уроков,  $tStart_i$  – дата первого урока,  $Sch_i$  – список с указанием в какие дни недели и время будут проходить занятия.

Как легко можно заметить, заявки на еженедельные занятия представимы в виде списка одиночных заявок, поэтому в дальнейшем будут рассматриваться только запросы вида:

$$Req_i = (S_i, VMCount_i, tStart_i, tEnd_i).$$

Далее представлено описание алгоритма отслеживания использования лицензий, который основан на предварительной оценке значений поступающих запросов, таких как:

а) пользователи могут отправлять заявки на добавление в расписание только для текущего учебного года;

б) продолжительность урока в большинстве школ равна 45 минутам;

в) параметры  $tStart_i$  и  $tEnd_i$  указаны с точностью до минуты.

Рассмотрим временной промежуток, равный одному году. Для високосного года  $T = 527040$  минут. Отслеживание количества лицензий будет происходить с помощью массива `resources` размера  $T \times SoftwareCount$ . Таким образом, идея алгоритма состоит в хранении информации об использовании лицензий для каждой минуты следующего года.

Пусть уже обработано  $m$  запросов, тогда  $(m+1)$ -й запрос возможно добавить в расписание, если выполняется условие:

$$\forall j \in [tStart_{m+1}, tEnd_{m+1}], \forall s \in S_{m+1}, LicenseCount_s \neq -1:$$
$$resources[j][s] + VMCount_{m+1} \leq LicenseCount_s,$$

где  $j$  - отдельная минута временного отрезка,  $S_{m+1}$  - множество программного обеспечения, необходимое для  $(m+1)$ -го запроса,  $LicenseCount_s$  - имеющееся количество лицензий для программного обеспечения  $s$ .

Чтобы добавить запрос в расписание необходимо выполнить следующие шаги:

```
For j = tStartm+1 ... tEndm+1
  For s : Sm+1
    resources[j][s] += VMCountm+1
```

Рассмотрим вычислительную сложность алгоритма:

а) асимптотика проверки возможности добавления запроса в расписание -  $O(t * SoftwareCount)$ , где  $t = tEnd_{m+1} - tStart_{m+1} + 1$ ;

б) асимптотика добавления запроса в расписание -  $O(t * SoftwareCount)$ ;

в) асимптотика расхода памяти –  $O(T * \text{SoftwareCount})$ .

Предположив, что  $t$  и  $T$  – константы, можно утверждать, что сложность алгоритма будет зависеть исключительно от  $\text{SoftwareCount}$ , то есть время выполнения и затрачиваемая память лишь линейно зависят от количества различного программного обеспечения, и не зависят от количества обработанных заявок. Однако, из-за больших констант, использование этого алгоритма для небольшого количества заявок не эффективно, так как существуют подходы, которые более универсальны (не нужно накладывать ограничения по времени), и при этом будут менее требовательны к памяти и более быстрые. Кроме того, для данного алгоритма можно использовать следующие оптимизации:

1) если операции проверки гораздо чаще, чем операции вставки, то можно вместо обычного массива использовать структуру данных дерево отрезков. Тогда асимптотика проверки равна  $O(\log(T) * \text{SoftwareCount})$ , асимптотика вставки равна  $O(t * \log(T) * \text{SoftwareCount})$ , а количество используемой памяти  $O(4 * T * \text{SoftwareCount})$ . Кроме того, можно заметить, что время проверки больше не будет зависеть от  $t$ ;

2) если уменьшить точность с одной минуты до  $k$ , то можно ускорить работу алгоритма и уменьшить количество используемой памяти в  $k$  раз. Это решение не изменяет асимптотику, но позволяет уменьшить константы  $T$  и  $t$ .

### **3.8 Исследование производительности сервиса РЦКД для отслеживания использования лицензий на программное обеспечение**

Для проведения эксперимента Web-сервис и база данных были развёрнуты на одной виртуальной машине (8 ядер 3GHz, 4 GB RAM, 20 GB HDD). Для БД были выделены отдельные четыре ядра, оставшиеся использовались сервисом. Управление ядрами проводилось с помощью программы `taskset`.

С целью имитации клиента было написано два модуля на языке C#: основной (Main) и вспомогательный (VUser). Полная схема работы представлена на рисунке 3.7.

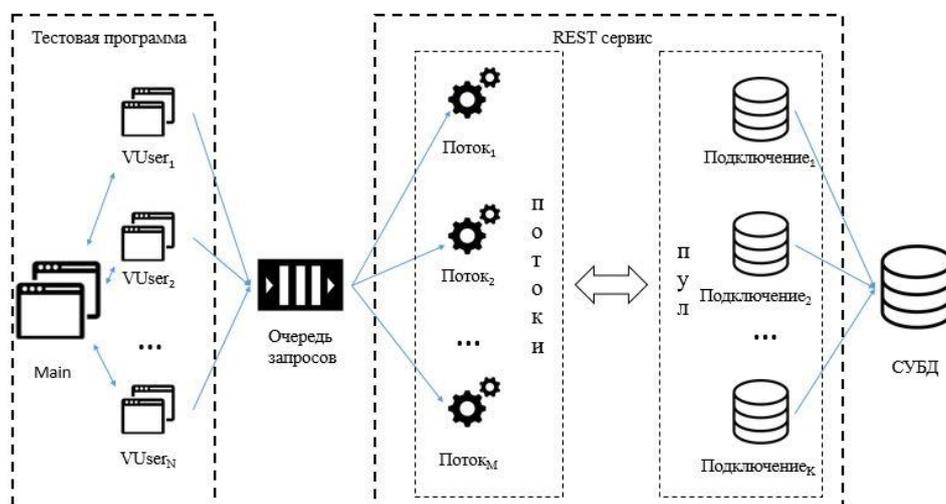


Рисунок 3.7 – Структура тестового стенда

Модуль VUser представляет собой отдельного «виртуального пользователя», который отправляет  $\mu$  запросов в секунду (CheckLicenses или BookLicenses, в зависимости от настройки) в течение T секунд. Каждый запрос, дождавшись своей очереди, начинает обрабатываться одним из M потоков. Получение информации из БД осуществляется с помощью пула, который возвращает любое из свободных в данный момент подключений (общее количество подключений - K).

Основной блок (Main) запускает одновременно N модулей VUser, передавая им параметры  $\mu$  и T, ожидает их завершения, после чего собирает и обрабатывает результаты (среднее количество запросов в секунду, которое обрабатывалось сервисом, среднее время задержки). Параметрами  $\mu$  и N можно контролировать интенсивность генерируемых запросов в секунду:

$$\lambda = \mu * N.$$

Для того, чтобы исключить влияние некоторых факторов (например, неодновременные старт и завершение всех модулей VUser), при обработке не учитываются первые и последние 10 % времени работы.

В первую очередь было проведено исследование производительности и масштабируемости сервиса с целью оценки зависимости количества обрабатываемых в секунду запросов от количества используемых ядер. Для этого был проведён эксперимент со следующими показателями:

- а) интенсивность от 1500 запросов до 6500, с шагом в 500 запросов;
- б) длительность теста - 60 секунд;
- в) количество выделенных потоков и размер пула подключений – 24;
- г) количество ядер для сервиса - от одного до четырёх.

Каждый раз вызывалась проверка с блокировкой – BookLicenses. Результаты представлены на рисунке 3.8

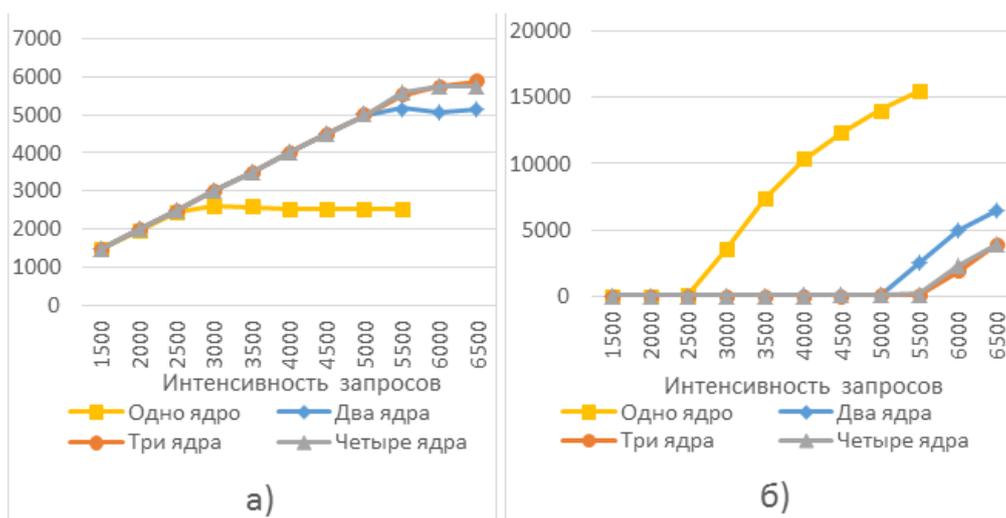


Рисунок 3.8. – Графики зависимости количества обрабатываемых запросов в секунду (а) и среднего времени ожидания (б) от интенсивности запроса для разного количества ядер

На графиках видно, что количество обработанных запросов в секунду линейно возрастает относительно интенсивности до тех пор, пока не достигает границы возможностей сервиса. При превышении этой точки, заявки в очереди сервиса начинают накапливаться, время ожидания начинает расти линейно. Продолжительная нагрузка на сервис может приводить к превышению времени ожидания ответа или к выводу сервиса из работоспособного состояния.

Для того, чтобы сделать вывод о масштабируемости, было проведено сравнение результатов для 1-го, 2-х, 3-х и 4-х ядер. Результаты показали, что на 1-м ядре сервис может обрабатывать до 2500 запросов в секунду, тогда как на 2-х – уже около 5000 (прирост производительности в 2 раза). Но если сравнивать 2 и 3 ядра, то можно заметить, что разница между ними уже не столь велика – всего около 700 запросов в секунду, между 3-мя и 4-мя - её практически нет. Это говорит о том, что данный способ не подходит для наращивания производительности и для этой цели следует искать другие методы.

Основные причины задержек в работе сервиса:

а) блокировки при проверке расписания. Так как потоки могут читать и изменять одни и те же данные, к ним необходим последовательный доступ (пока один запрос обрабатывается, остальные ожидают своей очереди). Время ожидания можно уменьшить, если блокировать ресурсы по частям или использовать потоконезависимые типы данных;

б) оптимальное соотношение количества потоков и размера пула. Очевидно, что при увеличении числа одновременно работающих потоков будет увеличиваться производительность. Но этот рост ограничен, так как, начиная с определенного числа потоков, затраты на использование и переключение между потоками перестанут быть целесообразными. С размером пула всё гораздо сложнее – требуется не только оптимизация времени работы сервиса, но и работы базы данных. Оптимальные значения находятся экспериментальным путем;

в) время ответа от базы данных. Из-за необходимости получения дополнительных данных из БД, нагрузка при каждом запросе создается не только на сервис, но и на базу, что ведет к увеличению её времени отклика. В результате производительность может снижаться именно из-за простоя сервиса во время ожидания данных от БД. Существует два способа решения этой проблемы: увеличение ресурсов для базы данных и сокращение количества обращений сервиса. Эффективной мерой в данной ситуации будет отказ от обращения к базе за редко изменяемыми данными и хранение их у себя (с периодическим обновлением);

г) скорость Интернет-канала. Не смотря на то, что отдельная заявка имеет небольшой размер (HTTP заголовок и параметры в формате JSON), при их большом количестве, они не будут успевать поступать на обработку сервисом. В данном случае, самой эффективной мерой будет увеличение скорости Интернет-соединения у провайдера облачных услуг.

Далее было проведено исследование влияния блокировок потоков на производительность сервиса, в рамках которого вместо функции BookLicenses, использовалась CheckLicenses, так как она не изменяет данные в сервисе. Для выявления возможного повышения уровня масштабируемости эксперимент проводился для 3-х и 4-х ядер. Количество потоков и размер пула подключений к БД был оставлен без изменений. Результаты представлены на рисунке 3.9.

Для лучшей визуализации отсутствия прироста в производительности, на рисунке 3.9 также отражены результаты предыдущего эксперимента. Анализ графиков показал, что внутренние блокировки на данный момент не оказывают заметного влияния на производительность сервиса.

Необходимость следующего эксперимента обусловлена потребностью определения оптимального количества потоков и размера пула подключений к БД. В рамках данного исследования оба параметра изменялись в диапазоне от 4 до 24.

Экспериментальное исследование проводилось на 4 ядрах с интенсивностью потока  $\lambda=6500$  и использованием запросов с временной блокировкой программного обеспечения (BookLicenses). Результаты представлены на рисунке 3.10. При увеличении количества потоков и размера пула с 4 до 8 на графике заметно значительное увеличение производительности. В диапазоне от 8 до 12 оно незначительно, от 12 и выше его нет.

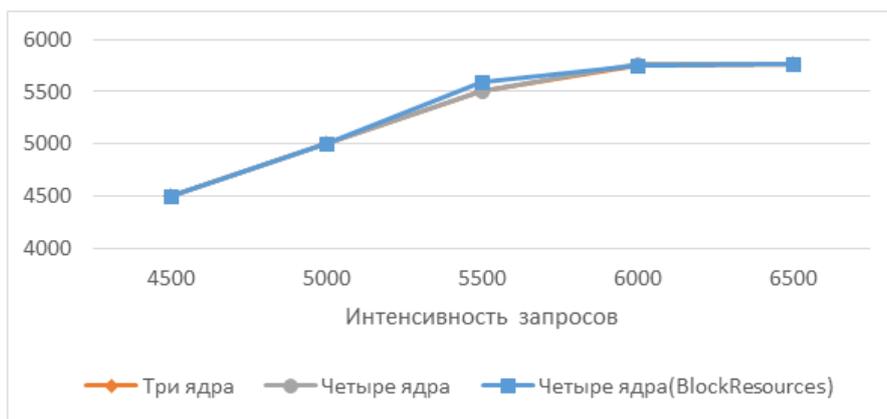


Рисунок 3.9 – Графики зависимости количества запросов в секунду при отсутствии блокировки на бронирование от интенсивности запросов для разного количества ядер

Если количество потоков меньше размера пула – изменения производительности также не наблюдается, так как часть подключений к базе никогда не используется, аналогично не даёт прироста производительности количество потоков, превосходящее число возможных подключений.

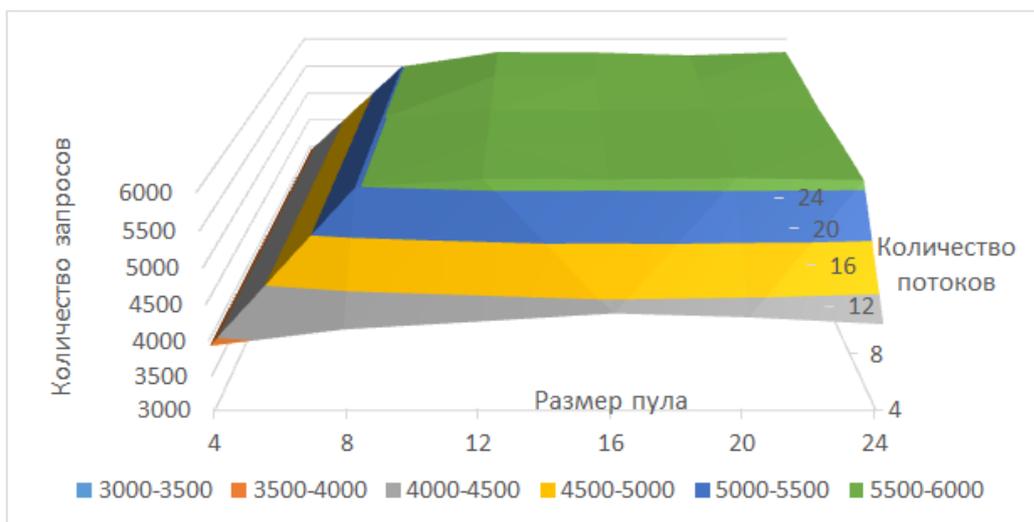


Рисунок 3.10 – График зависимости обрабатываемых запросов в секунду от количества потоков и размера пула

Следовательно, идеальное соотношение количества потоков / размера пула подключений для данной конфигурации 12/12.

Следующее исследование посвящено задержкам при обращении к БД за информацией. Для того, чтобы исключить влияние базы данных, информация об образе будет дополнительно храниться в оперативной памяти сервиса. Параметры будут те же, что и в первом исследовании, представленном на рисунке 3.7 Для лучшей визуализации на рисунке 3.11 представлены только результаты для 3-ех и 4-ех ядер, интенсивность изменяется в пределах 5500-6500 запросов в секунду, и для сравнения добавлены результаты из рисунке 3.7. Как можно заметить, увеличения производительности нет. Таким образом, обращения к БД не являются влияющим фактором.

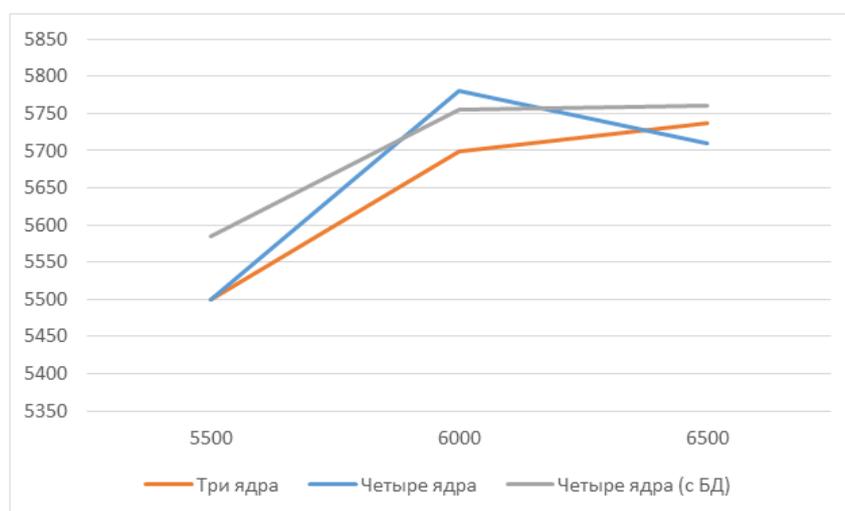


Рисунок 3.11 – График зависимости обрабатываемых запросов в секунду от интенсивности запросов для разного количества ядер и запросов без обращения к базе данных

Исследование, представленное на рисунке 3.12, показывает, что низкая скорость Интернет-соединения сильно отражается на количестве обрабатываемых заявок в секунду. Параметры исследования использованы те же, что и на рисунке 3.7 Если проводить сравнение, то производительность для 3-ех ядер повысилась, в среднем, с 5700 заявок в секунду до 6500. Тогда как для 4-ех ядер прирост составил до 7200 заявок. Таким образом, в данном случае решающее влияние оказывала недостаточно высокая скорость Интернет-соединения.

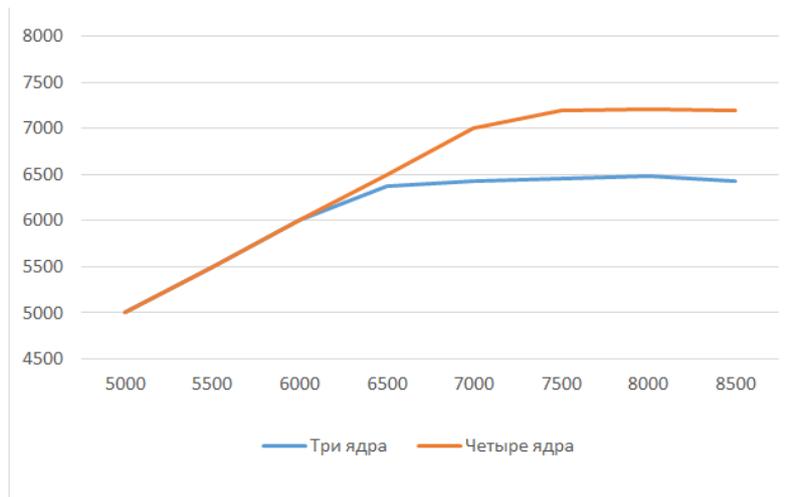


Рисунок 3.12 – График зависимости обрабатываемых запросов в секунду от количества ядер для расширенного Интернет-канала

Следующее исследование совмещает те условия, что использовались в исследованиях, представленных на рисунке 3.11 и рисунке 3.12, то есть Интернет-канал с высокой пропускной способностью и хранение информации в оперативной памяти сервиса. Параметры использовались те же, что и на рисунке 3.7 Результаты представлены для 4-ех ядер. Как можно заметить, исключив влияние БД из результатов, представленных на рисунке 3.12, увеличение производительности выросло с 7200 до, минимум, 9000 заявок в секунду. К сожалению, особенности тестового стенда не позволяют генерировать заявки с интенсивностью более, чем 9000.

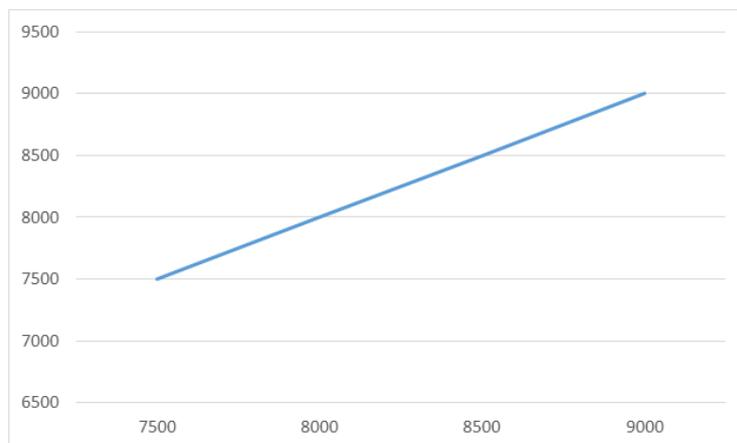


Рисунок 3.13 – График зависимости обрабатываемых запросов в секунду от интенсивности запросов для расширенного Интернет-канала и без запросов к БД

Экспериментальные исследования показали значительное увеличение производительности при использовании двух ядер, вместо одного, при трёх ядрах имелся незначительный дальнейший рост, а при четырёх он отсутствовал. Дальнейшее исследование показало отсутствие влияния блокировок, связанных с синхронизацией потоков. Также было выявлено оптимальное соотношение количества потоков, работающих на выделенных ядрах, и размера пула подключений к базе данных. Было показано, что скорость Интернет-соединения сильно сказывалось на производительности сервиса. Не смотря на то, что обращение к БД в первом исследовании не оказывала влияния, в ходе последнего было установлено, что впоследствии хранение редко изменяемых данных в оперативной памяти сервиса приводит к увеличению производительности.

## 4 Разработка методики оптимизации ресурсного обеспечения РЦКД

Целью оптимизации ресурсного обеспечения РЦКД является экономия средств, затрачиваемых на аренду виртуальных машин и программного обеспечения за счет выбора провайдера облачных услуг, оптимальных параметров виртуальных машин.

Для исследования конкретных экономических показателей был проведен анализ предложений компаний поставщиков облачных ресурсов на российском рынке. Результаты исследования сведены в таблицу 4.1.

Таблица 4.1 – Сравнительная стоимость основных провайдеров в российском сегменте Интернет.

Провайдер	Средние цены на ресурсы (многие провайдеры имеют нелинейную зависимость стоимости от количества)			Цена конфигурации, ядер/ОЗУ/накопитель, руб. в месяц		
	Цена 1 ядра, руб. в месяц	Цена 1Gb ОЗУ, руб. в месяц	Цена 1Gb накопителя, руб. в месяц	1/1/10	2/2/20	4/4/40
1	2	3	4	5	6	7
simplecloud.ru	20	40	10	250	500	1000
ruvds.com	60	140	1	300	500	1000
neoserver.ru	70	240	5	360	720	1440
hostkey.ru	77	147	12	342	682	1350
vpsville.ru	100	150	5	499	899	1499
vscale.io	100	290	1	400	800	1600
lcloud.ru	110	315	5	475	950	1900
flops.ru	125	280	3	500	1000	2000
clodo.ru	153	670	26	930	1860	3720
maxiplace.ru	158	287	13	627	1198	2341
cloudlite.ru	180	238	4	387	1057	2113
lunacloud.ru	180	252	2	450	900	1800
infobox.ru	194	302	12	619	1237	2474
oblotekta.ru	200	590	7	800	-	2200
ispserver.com	210	220	4	640	1090	1950
clod-pro.ru	250	550	5	1100	1700	3400
rtcomm-yug.ru	260	480	4	780	1560	3120
imserver.ru	300	75	25	425	850	1700

Продолжение таблицы 4.1

1	2	3	4	5	6	7
slidebar.ru	300	400	9	880	1580	3160
selectel.ru	402	146	7	767,7	1388,7	2484,7
oncloud.mts.ru	531	531	11	1593	2655	4779
makecloud.ru	576	254	7	1425	2695	5395
cloud.softline.ru	700	250	15	1965	2665	4689
rtcloud.ru	780	480	4	1038	2075	4150
rackstore.ru	1380	180	18	1221	2297	4593
activecloud.ru	1900	874	3	3100	4950	6200

На рисунке 4.1 показано графическое представление цен на одинаковые конфигурации у разных поставщиков.

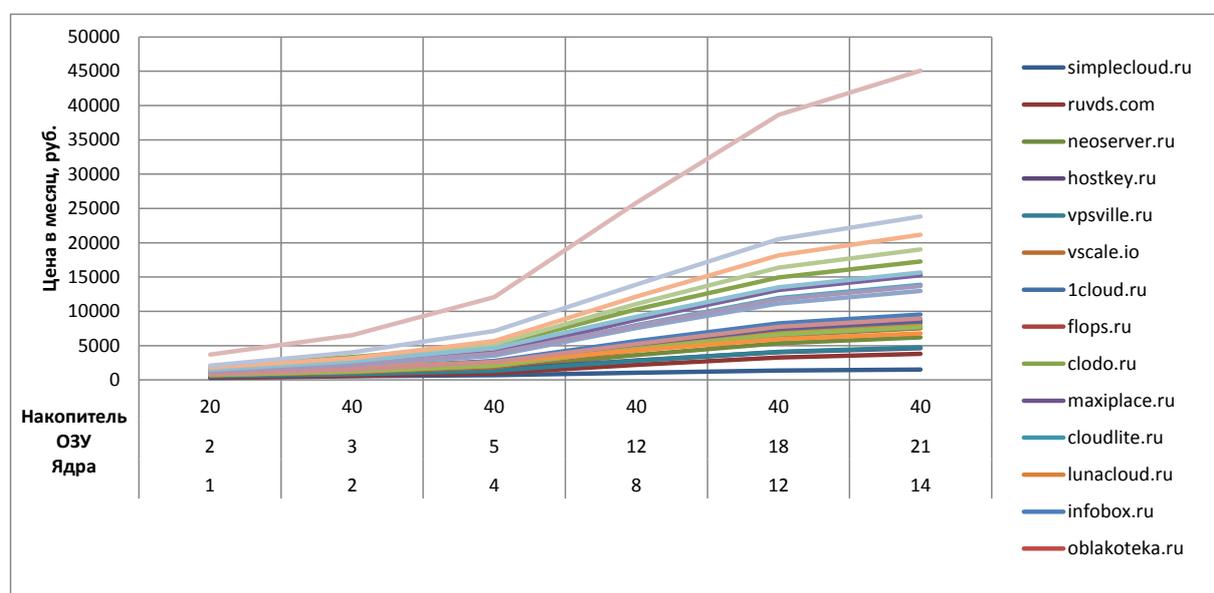


Рисунок 4.1 – Динамика стоимости ресурсов

Как видно из рисунка есть существенные различия в стоимости одинаковых ресурсов разных поставщиков. Но не все поставщики из нижнего ценового сегмента удовлетворяют требованиям размещения РЦКД, а именно:

- а) наличие API для управления созданием серверов;
- б) наличие почасовой или поминутной тарификации серверов;
- в) отсутствие необходимости резервировать мощности заранее.

Для размещения ресурсов в облаке в виде сервера Windows с удаленным доступом существуют две стратегии:

а) размещение большого числа маломощных серверов, по одному на пользователя;

б) размещение меньшего количества серверов мощных конфигураций для одновременной работы многих пользователей.

Для изучения экономически эффективной стратегии были проведены натурные испытания комфортного количества ОЗУ и ядер для работы нескольких пользователей в офисном режиме (работа с офисным пакетом) и в высокопроизводительном режиме (работа со средами программирования и средствами моделирования).

Результаты моделирования показаны на рисунке 4.2 (процессоры) и 4.3 (ОЗУ).

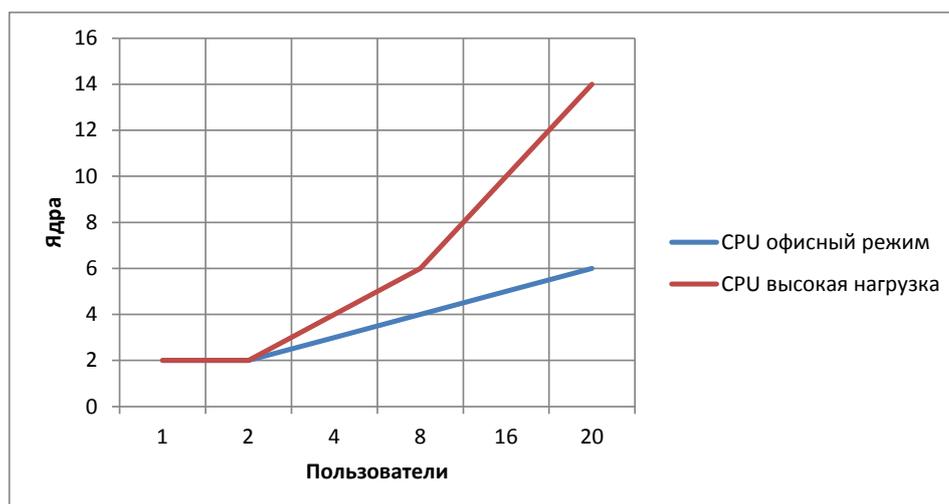


Рисунок 4.2 – Результаты испытаний по процессорам

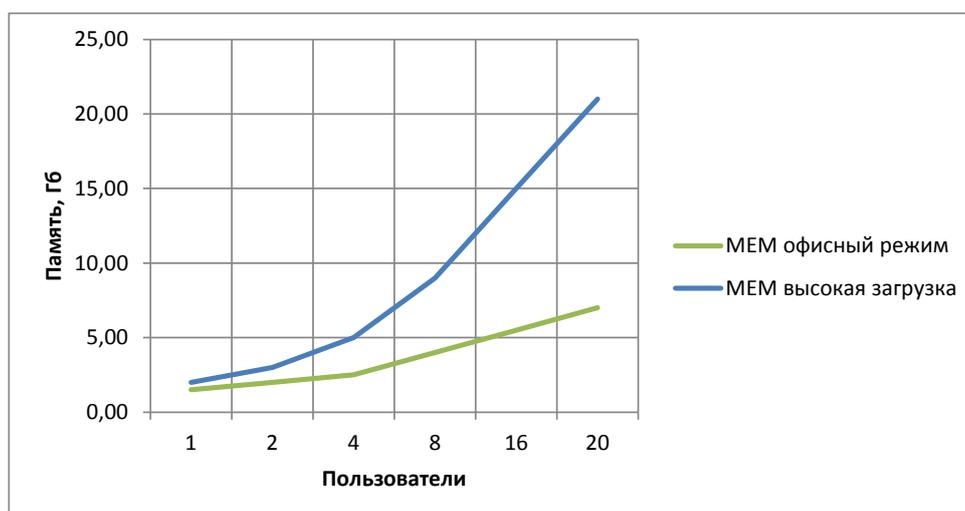


Рисунок 4.3 – Результаты испытаний по памяти

С использованием результатов исследования цен и наружного испытания была составлена модель использования ресурсов по обеим стратегиям. Результаты сведены в таблицу 4.2.

Таблица 4.2 – Результаты моделирования стратегий использования.

Кол-во пользователей на 1 сервер	Процессоров, офисный режим, шт	ОЗУ, офисный режим, Гб	Средняя цена сервера в месяц, высокая	Цена за 1 пользователя, высокая нагрузка	Процессоров, высокая нагрузка, шт	ОЗУ, высокая загрузка, Гб	Средняя цена сервера в месяц, офисная	Цена за 1 пользователя, офисная нагрузка, руб
1,00	2,00	1,50	637,05	637,05	2,00	2,00	743,40	743,40
2,00	2,00	2,00	743,40	371,70	2,00	3,00	956,10	478,05
4,00	3,00	2,50	949,75	237,44	4,00	5,00	1581,50	395,38
8,00	4,00	4,00	1368,80	171,10	6,00	9,00	2632,30	329,04
16,00	5,00	5,50	1787,85	111,74	10,00	15,00	4308,50	269,28
20,00	6,00	7,00	2206,90	110,35	14,00	21,00	5984,70	299,24
30,00	10,00	11,00	3457,70	115,26				
40,00	14,00	16,00	4921,20	123,03				

По вычисленным значениям был построен график стоимости владения сервером суммарно и в пересчете на одного пользователя (рисунок 4.4).

Из графика видно, что при использовании одного сервера для размещения нескольких пользователей даже с учетом дополнительных расходов на лицензии стоимость в пересчете на одного пользователя снижается с количеством задействованных на сервере учетных записей по сравнению с начальной точкой отсчета (минимальным сервером для 1 пользователя). Это означает, что стратегия размещения нескольких пользователей на одном сервере выгодней с экономической точки зрения.

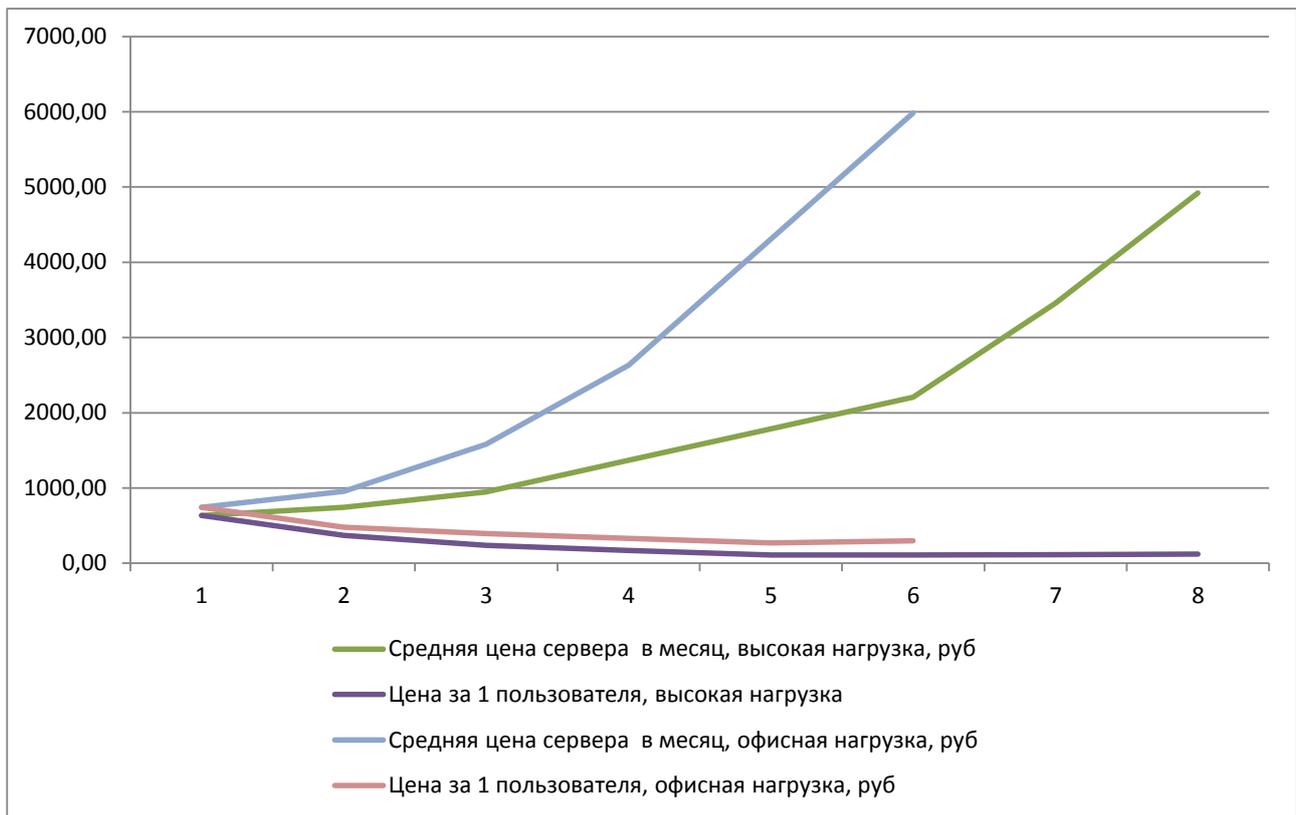


Рисунок 4.4 – Стоимость владения

С учетом того, что типовое количество рабочих столов на один виртуальный класс составляет 10-12, исходя из проведенного исследования, наиболее оптимальным с точки зрения затрат является создание одного виртуального сервера на виртуальный класс.

## 5 Разработка проекта системы управления регионального центра коллективного доступа к образовательным программным продуктам на базе облачных услуг отечественных провайдеров

### 5.1 Описание проекта ресурсного центра коллективного доступа

Архитектура системы управления регионального центра коллективного доступа изображена на рисунке 5.1.

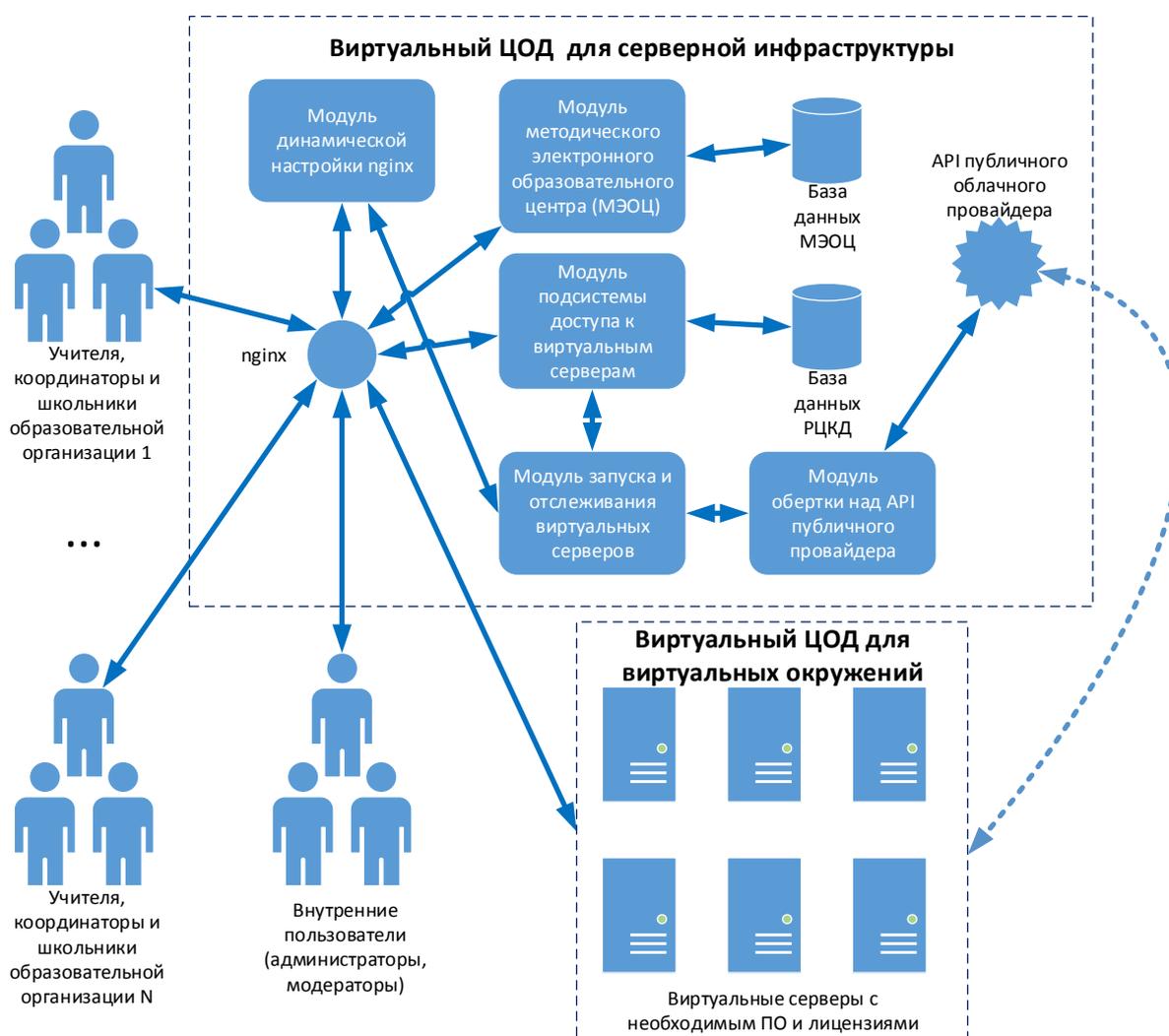


Рисунок 5.1 – Архитектура регионального центра коллективного доступа

Основными пользователями регионального центра коллективного доступа выступают:

а) представители образовательных организаций – учителя, координаторы и школьники;

б) внутренние пользователи – администраторы и модераторы;

Вся система может быть развернута с использованием минимум двух виртуальных ЦОД:

а) виртуальный ЦОД для сетевой инфраструктуры – содержит основные инфраструктурные компоненты регионального центра коллективного доступа;

б) виртуальный ЦОД для виртуальных окружений – используется для запуска виртуальных серверов (виртуальных машин) с виртуальными окружениями для работы учителей и школьников.

Подобное разделение позволяет эффективно изолировать инфраструктурные ресурсы от виртуальных окружений. Это увеличивает безопасность первых в силу отсутствия виртуальной сети между ними, а также снижает шансы случайного отключения инфраструктурной компоненты при ошибочном использовании API облачного провайдера.

Дополнительным преимуществом является отдельный биллинг, что позволяет отделить расходы на инфраструктуру от расходов на работу виртуальных окружений.

Кроме того, возможны варианты, когда используются услуги нескольких облачных провайдеров. В этом случае, например, виртуальный ЦОД для сетевой инфраструктуры может размещаться у одного провайдера, а для виртуальных окружений – у другого. Также предложенная архитектура допускает гибкость в плане запуска нескольких экземпляров каждого из двух типов виртуальных ЦОД у нескольких провайдеров с использованием полного или частичного набора инфраструктурных компонент.

Основными инфраструктурными компонентами, изображенными на рисунке 5.1, являются:

а) модуль динамической настройки nginx – осуществляет настройку доступа пользователей через nginx к виртуальным окружениям, методическому электронному образовательному центру или к подсистеме доступа к виртуальным серверам;

б) модуль методического электронного образовательного центра – предоставляет Web-интерфейс для доступа и размещения учебно-методических материалов, необходимых учителям для проведения занятий;

в) модуль подсистемы доступа к виртуальным серверам – предоставляет Web-интерфейс для задания расписания занятий, формирования сведений об образовательной организации и получения доступа к виртуальным окружениям по расписанию или по требованию;

г) модуль запуска и отслеживания виртуальных серверов – использует обертку над API публичного облачного провайдера для запуска и остановки виртуальных серверов, предоставляющих необходимые виртуальные окружения для проведения занятий;

д) модуль обертки над API публичного облачного провайдера – единственный провайдеро-зависимый модуль системы, представляющий собой прослойку между API провайдера и модулем запуска и отслеживания виртуальных серверов. В случае смены провайдера должен переписываться только данный модуль;

е) база данных МЭОЦ – база данных, содержащая учебно-методические материалы, информацию об их авторстве, названия предметов, классы, темы и разделы;

ж) база данных РЦКД – база данных, содержащая все остальные сведения ресурсного центра коллективного доступа, включая сведения об образовательных организациях, пользователях, координаторах, учителях, расписании и т.п.

Разделение информации по двум БД дает дополнительный выигрыш в производительности работы модулей методического электронного образовательного центра и подсистемы доступа к виртуальным серверам, а также обеспечивает больший уровень безопасности.

Подробное описание инфраструктурных компонент ресурсного центра коллективного доступа приведено в разделе 7.

За счет обертки над API облачного провайдера, предоставляющей единый интерфейс, все остальные компоненты ресурсного центра не привязывают к возможностям и API конкретного провайдера. При необходимости легко сменить

провайдера или добавить поддержку нового провайдера – нужно просто переписать модуль обертки над API.

На рисунке 5.2 изображена диаграмма вариантов использования UML для регионального центра коллективного доступа, она позволяет определить основных пользователей системы и доступные им функциональные возможности.

Основными пользователями (действующими лицами в терминах UML) являются:

а) ученик – пользователь по умолчанию, не имеющий учетной записи в системе. Может работать с методическими материалами, включая их просмотр, поиск и скачивание. Он также может получать доступ в созданные учителем виртуальные окружения;

б) учитель – зарегистрированный пользователь системы, наследуется от ученика (может делать все, что и ученик). Также учитель может входить в систему и выходить из нее, просматривать расписание и работать с занятиями по требованию (создавать и отменять их). Учитель является представителем одной или нескольких образовательных организаций;

в) координатор – зарегистрированный пользователь системы, наследуется от ученика. Координатор может входить в систему и выходить из нее, формировать расписание занятий (шаблон расписания и расписание на текущую неделю), работать с учителями (приглашать учителей, удалять и редактировать записи о них), работать с классами учеников (создавать, изменять и удалять классы), работать с расписанием звонков (создавать, изменять и удалять), редактировать описание образовательной организации. Координатор может курировать одну или несколько образовательных организаций;



г) модератор – зарегистрированный пользователь системы, являющийся потомком ученика. Доступные действия для модератора: входить в систему и выходить из нее, формировать структуру предмета, публиковать, удалять и изменять методические материалы. Под формированием структуры предмета подразумевается работа с темами (создание, изменение и удаление), работа с разделами внутри тем (создание, изменение и удаление);

д) администратор – зарегистрированный пользователь системы, имеющий максимальные права. Наследуется от ученика, также может входить в систему и выходить из нее, управлять пользователями (создавать в системе учетные записи координаторов и модераторов, изменять и удалять их, назначать координаторам школы, а модераторам – права доступа к предметам и классам), работать с образовательными организациями (создавать и удалять их, редактировать описание).

В рамках ресурсного центра коллективного доступа администратор создает координаторов и модераторов, координаторы создают и добавляют учителей. Ученики представляют собой незарегистрированных пользователей системы, что является дополнительным достоинством, т.к. это решает ряд проблем, связанных с отсутствием необходимости учителям добавлять в систему учеников, формировать и часто менять индивидуальные пароли. При этом доступ учеников к виртуальным окружениям осуществляется по шестизначному коду безопасности.

В рамках проекта системы управления региональным центром коллективного доступа к образовательным программным продуктам на базе облачных услуг отечественных провайдеров разработана архитектура системы, иллюстрирующая ее основные компоненты, также была построена диаграмма вариантов использования UML, отражающая основных пользователей системы и их функциональные возможности.

## **5.2 Анализ существующих решений для создания Web-приложений в рамках ресурсного центра коллективного доступа**

Архитектура ресурсного центра коллективного доступа включает в себя два модуля, представляющих собой Web-приложения – модуль методического электронного образовательного центра и модуль подсистемы доступа к виртуальным серверам.

Для ускорения разработки Web-приложений, обеспечения их высокой надежности и безопасности, а также эффективного использования ресурсов, наиболее актуально использование специальных систем создания Web-приложений, так называемых каркасов (фреймворков).

Web-каркас – это базовая платформа, которая задает структуру системы, осуществляет выполнение модулей и определяет общий стандарт кодирования. Основное достоинство использования каркасов – повторное использование кода, использование стандартных библиотек, которые сокращают время разработки Web-порталов.

Большая часть современных Web-каркасов использует шаблон Model View Controller (MVC, Модель-Представление-Контроллер) [8]. Данный шаблон включает в себя три компоненты:

а) модель (Model) – состояние приложения. Это данные, хранящиеся в программе;

б) представление (View) – визуальное представление данных модели в виде Web-страниц;

в) контроллер (Controller) – обрабатывает запросы пользователей, обращаясь к модели, и передает данные представлению для отображения.

В общем случае Web-фреймворк на основе MVC состоит из следующих компонентов:

а) шаблонизатор. Отвечает за независимость вёрстки от программного кода;

б) роутер. Распознаёт URL, по которому произошло обращение к серверу;

в) модуль доступа к базе данных;

г) модуль кэширования. Ускоряет загрузку страниц;

д) модуль безопасности. Аутентификация и авторизация пользователей;

е) файлы конфигурации.

Рассмотрим наиболее известные MVC фреймворки.

ASP.NET MVC Framework [9] – каркас от компании Майкрософт, являющийся частью библиотеки ASP.NET. Ее исходный код имеет лицензию Microsoft Public License (MS-PL). Язык программирования – C#.

Zend Framework (ZF) [10] – объектно-ориентированный каркас компании Zend с открытым исходным кодом. Язык программирования – PHP. Каркас включает в себя набор библиотек для MVC, обеспечения безопасности, интеграции с другими сервисами.

Symfony [11] – простой и гибкий Web-фреймворк с лицензией MIT. Язык программирования – PHP. Содержатся средства для работы с данными, создания представлений.

Ruby On Rails [12] – фреймворк для создания Web-приложений на Ruby. Основан на методах Agile-программирования, хорошо документирован.

Apache Struts [13] – предназначен для создания корпоративных Web-приложений на Java. В его основе лежит Java Servlet API и MVC.

Django [14] – фреймворк на языке Python. Архитектура во многом повторяет Ruby On Rails [12]. В Django для задания и настройки путей URL используются регулярные выражения, не требуется строгая структура контроллеров и моделей. Также Django предоставляет собственный ORM для работы с реляционными данными. Python имеет хорошо документированную библиотеку классов, лаконичный и простой синтаксис [14]. В каркасе Django имеется подробная и исчерпывающая документация. Одной из уникальных особенностей Django является автоматически генерируемая страница для администратора. Django предназначен для работы с высокими нагрузками, имеются средства кэширования и распределения нагрузки [14].

Таким образом, можно сделать вывод, что наиболее предпочтительным из перечисленных каркасов для реализации Web-приложений в рамках РЦКД является Django. Язык Python также был выбран для создания других инфраструктурных компонент РЦКД – модуля запуска и отслеживания виртуальных серверов и модуля обертки над API публичного облачного провайдера.

## 6 Разработка технологии защиты информации в РЦКД

Поскольку с виртуальными рабочими местами будут работать дети, защита в РЦКД должна быть двусторонней – как самой инфраструктуры от внешних воздействий, так и пользователей от нежелательного контента. Для реализации этих степеней защиты предлагается использовать следующие решения:

а) установка и настройка брандмауэра (межсетевого экрана) на все узлы, запрет входящего трафика кроме как от выделенных узлов инфраструктуры (обратных распределяющих прокси-серверов), запрет доступа в Интернет без фильтра;

б) установка антивируса на Windows системы;

в) установка систем мониторинга на все узлы;

г) установка прокси-сервера с фильтрами для доступа в Интернет с узлов;

д) настройка DNS на безопасный фильтрованный DNS Yandex;

е) применение политик ограниченного использования программ;

ж) периодическое восстановление всех Windows узлов из образов;

з) установление защищенного соединения между пользователем и компонентами РЦКД;

и) использование разовых кодов для предоставления доступа к виртуальным окружениям.

Рассмотрим подробно реализацию этих мероприятий. Для Linux систем сетевой фильтр является встроенным и управляется через iptables [15]. Каждый узел настраивается типовым образом для фильтрации нежелательного трафика служб, незадействованных для внешнего обмена.

Также установлены системы Fail2ban [16] для фильтрации злонамеренной активности на сервисах SSH и HTTP. Одни из наиболее атакуемых сервисов – SSH, так как он используется для удаленного входа в систему. Для решения данной проблемы можно использовать программу Fail2ban. Fail2ban – это локальный сервис, который отслеживает файлы регистрации запущенных программ, и на основании различных условий блокирует по IP-адресам найденных нарушителей. Например, если определенный IP-адрес, после определенного количества попыток, не прошел

аутентификацию, что защищает SSH-серверы от атаки методом перебора паролей. Таким образом, Fail2ban будет позволять совершать вход в систему с помощью SSH только авторизованному пользователю, а не удаленному злоумышленнику. Также умеет бороться с различными атаками на \*NIX-сервисы: Apache, Nginx, ProFTPD, vsftpd, Exim, Postfix, named, и т.д.

Для Windows систем используется встроенный межсетевой экран в режиме повышенной безопасности с фильтрацией всего входящего трафика с узлов, отличных от внутренних узлов инфраструктуры. Для исходящего трафика будут разрешены только протоколы HTTP/HTTPS и только на прокси-сервер.

В качестве антивируса выбран Microsoft Security Essential [17], поскольку он совместим с серверным ПО и является бесплатным.

Microsoft Security Essentials (MSE) – служба от компании Microsoft для защиты компьютера. Обеспечивает защиту от вирусов, шпионских программ, троянских программ и т.д. Данное программное обеспечение работает только на активированных копиях операционной системы Windows.

Преимущества данного продукта:

- а) автоматическая проверка и загрузка обновлений вирусной базы данных;
- б) использование службы динамической подписи файлов;
- в) создание точек восстановления системы перед удалением вируса;
- г) возможность осуществлять запланированную проверку системы;
- д) защита в реальном времени;
- е) экономное использование оперативной памяти;
- ж) защита от пакетов программ rootkit;
- з) интеграция в брандмауэр Windows;
- и) поддержка службы динамических сигнатур;
- к) система проверки сети, которая защищает от сетевых эксплойтов.

Системы мониторинга должны помочь выявить нетипичное поведение или сбой узлов. Кроме того, система должна быть совместима с Linux и Windows и, желательно, иметь программных агентов для общения с сервером мониторинга. Одной из лучших систем является Nagios [18] – бесплатная система с агентским

принципом. Для Windows и Linux есть агенты, они могут работать через прокси, резервировать сервера.

В качестве прокси-сервера выбран squid для Linux [19], поскольку к нему можно легко интегрировать любые белые и черные списки сайтов, в том числе и открытые. Совместно с фильтрованным DNS это позволит максимально снизить вероятность доступа к нежелательному контенту.

Для форсирования доступа через прокси используются групповые политики и брандмауэр. Для автоматизации используется скрипт автоконфигурирования сервера при его создании. Он содержит изменение следующих веток реестра:

- Выключение автонастройки прокси

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\MAIN\FeatureControl\FEATURE_AUTOCONFIG_BRANDING]
"iexplore.exe"=dword:0
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings]
```

```
"EnableAutoProxyResultCache" = dword: 0
```

- отключение возможности настройки пользователем

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings]
```

```
"ProxySettingsPerUser"=dword:0
```

- настройка прокси сервера

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Internet Explorer\Control Panel]
```

```
"Proxy" = dword:1
```

```
"AutoConfig" = dword:1
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings]
```

```
"ProxySettingsPerUser" = dword:0
```

```
"DialupUseLanSettings" = dword:1
"WarnOnIntranet" = dword:0
"ProxyEnable" = dword:1
"MigrateProxy" = dword:1
"ProxyServer" = "http://user:paswd@proxy.56bit.ru:8080"
"ProxyOverride" = "<local>"
```

После таких настроек доступ в интернет будет осуществляться только через указанный прокси сервер. Для блокировки других способов доступа необходимо включить и настроить брандмауэр. Для автоматизации – это делается через PowerShell:

- включение брандмауэра

```
Set-NetFirewallProfile -Name Public -Enabled True
```

- запрет всех входящих и исходящих подключений по умолчанию

```
Set-NetFirewallProfile -DefaultOutboundAction Block
```

```
Set-NetFirewallProfile -DefaultInboundAction Block
```

- Разрешить только DNS от Яндекс 77.88.8.7 и 77.88.8.3

```
New-NetFirewallRule -DisplayName '56bit - YandexDNS' -Name 'ya.dns' -Action
Allow -Direction Out -Enabled True -Protocol UDP -RemoteAddress 77.88.8.7,77.88.8.3 -
RemotePort 53
```

- Разрешить только proxy.56bit.ru на порт 8080

Для этого необходимо получить все адреса

```
$ip = Resolve-DNSName -Type=A proxy.56bit.ru
```

Формирование списка адресов

```
$addr=New-Object System.Collections.Generic.List[System.Object]
```

```
$ip | foreach {$addr.Add($_.IPAddress)}
```

Создание правила

```
New-NetFirewallRule -DisplayName '56bit - Proxy' -Name '56bit.proxy' -Action Allow -Direction Out -Enabled True -Protocol TCP -RemoteAddress $addr -RemotePort 8080
```

Разрешить на вход удаленного доступа только r.56bit.ru на порт 80 и 443. Для этого необходимо получить все адреса

```
$ip = Resolve-DNSName -Type=A r.56bit.ru
```

Формирование списка адресов

```
$addr=New-Object System.Collections.Generic.List[System.Object]  
$ip | foreach {$addr.Add($_.IPAddress)}
```

Создание правила

```
New-NetFirewallRule -DisplayName '56bit - Remote' -Name '56bit.r' -Action Allow -Direction Out -Enabled True -Protocol TCP -RemoteAddress $addr -LocalPort 80,443
```

Выключение правил внешнего доступа для RDP и NetBios

```
Get-NetFirewallPortFilter |Where-Object -FilterScript { $_.LocalPort -Eq "3389" -or $_.LocalPort -Eq "137" -or $_.LocalPort -Eq "138" } | Get-NetFirewallRule | Set-NetFirewallRule -Enabled False
```

Разрешение RDP для организации поддержки из ОГУ

```
New-NetFirewallRule -DisplayName '56bit - RDP' -Name '56bit.RDP' -Action Allow  
-Direction IN -Enabled True -Protocol TCP -RemoteAddress 62.76.152.0/24 -LocalPort  
3389
```

Теперь необходимо настроить DNS серверы Яндекса [20] на всех адаптерах

```
Get-NetAdapter | Set-DNSClientServerAddress -ServerAddresses ("77.88.8.7" ,  
"77.88.8.3")
```

В Windows Server, используемом для развертывания виртуальных окружений, также создается учетная запись пользователя с ограниченными правами и работа осуществляется под ней.

Windows Server также позволяет запрещать запуск определенных приложений для пользователей. Для этого используется инструмент «Политика ограниченного использования программ», который является объектом групповой политики (GPO). В начале работы с политикой ограниченного использования программ необходимо создать новый объект [21]. При создании нового объекта групповой политики активируются политики по умолчанию. Затем выбирается уровень необходимой безопасности и настраиваются параметры политики: на какие типы файлов будет накладываться ограничение; будут ли применяться ограничения на локальных администраторов; следует ли проверять библиотеки DLL, выбор доверенных издателей. Затем создаются правила, которые запрещают или разрешают выполнение программ. Существует четыре типа правил: зона, путь, сертификат, хеш. Однако, следует заметить, что политики ограниченного использования не распространяются на: программы, написанные для .NET CLR, программы, запущенные от SYSTEM, макросы документов Microsoft Office, драйверы, будут работать независимо от настройки политик ограниченного использования. Политика ограниченного использования программ позволяет уменьшить риск работы на компьютере

вредоносных программ, разрешает использование программ в соответствии с имеющимся расписанием.

Периодическое восстановление всех Windows узлов из сохраненных образов позволит обезопасить систему от заложенных вирусов, троянов, закладок, сбоя в системе и от изменения настроек.

Еще одним средством защиты РЦКД является шифрование трафика между Web-браузером пользователя и инфраструктурными компонентами РЦКД. Для этого предлагается использовать криптографический протокол TLS (Transport Layer Security). Он обеспечивает шифрование, аутентификацию и целостность данных при работе с браузерами. В 2016 году рекомендованными являются версии протокола TLS 1.1 и TLS 1.2. TLS 1.2 является более предпочтительным, так как там используется другая базовая псевдослучайная функция и появилась поддержка шифров в режиме аутентифицированного шифрования [22]. С целью реализации TLS были получены сертификаты удостоверяющего центра на основные поддомены РЦКД. Это позволяет браузерам пользователей аутентифицировать Web-сервер и исключить возможные MITM-атаки.

Для доступа к виртуальным окружениям незарегистрированным ученикам в рамках РЦКД предлагается использовать разовые коды, представляющие собой пароли из шести цифр. Код доступа создается заново каждый раз при запуске виртуальной машины с виртуальными окружениями и действует до ее удаления или выключения. Он является общим для всех учеников, работающих с виртуальными окружениями на одной виртуальной машине.

Дополнительным средством защиты РЦКД может являться использование контейнеров Docker [23] для размещения основных инфраструктурных компонент системы, его использование возможно в будущем. Docker – это инструмент, предоставляющий удобный интерфейс для работы с LXC. Docker позволяет запускать процессы в изолированном окружении. Процесс, запущенный под Docker, работает в той же операционной системе, что и остальные процессы. Однако данный процесс не видит остальные процессы, файлы и все остальное за пределами своей «песочницы», ему кажется, что он работает в минимальном окружении. Docker позволяет запускать

множество контейнеров с приложениями. При взломе злоумышленником приложения, которое запущено в контейнере, он получает доступ только к контейнеру, но не к виртуальной машине. Это также сможет повысить безопасность системы.

## 7 Реализация модулей системы управления РЦКД

### 7.1 Модуль динамической настройки nginx

Для осуществления масштабирования предлагается использовать инфраструктуру DNS для обнаружения сервисов и для добавления новых и удаления старых адресов. В API провайдера есть DNS раздел для удаления и добавления записей. При запуске нового экземпляра сервера будет добавляться новая А запись, при остановке или отсутствии отклика запись будет удаляться. Параметр TTL таких записей должен быть установлен в 60 для оперативного обновления информации.

Nginx используется с динамической настройкой в двух конфигурациях:

- а) для кеширования запросов к МЭОЦ;
- б) для перенаправления запросов к виртуальным машинам.

Рассмотрим оба варианта использования. При использовании в МЭОЦ nginx работает как кеширующий обратный прокси. Для корректной эффективной работы необходимо:

- а) иметь актуальную информацию о вышестоящих серверах для распределения нагрузки;
- б) актуальную информацию о сессиях пользователей для перенаправления на тот же сервер, на котором началась сессия;
- в) быстродействующий кэш для отдачи кэшированного контента.

Рассмотрим пути поступления информации о вышестоящих серверах:

- а) от DNS службы;
- б) от внешних сервисов.

Так как была выбрана модель обнаружения сервисов через DNS, остановимся на его реализации. На рисунке 7.1 показана схема работы. При запросе от клиента выбирается один из вариантов действия: для разделов модерации важно отсутствие балансировки между серверами, нужно попадать только на первичные сервера, поэтому их адреса поставляются скриптом DNS клиента, который периодически обновляет адреса.

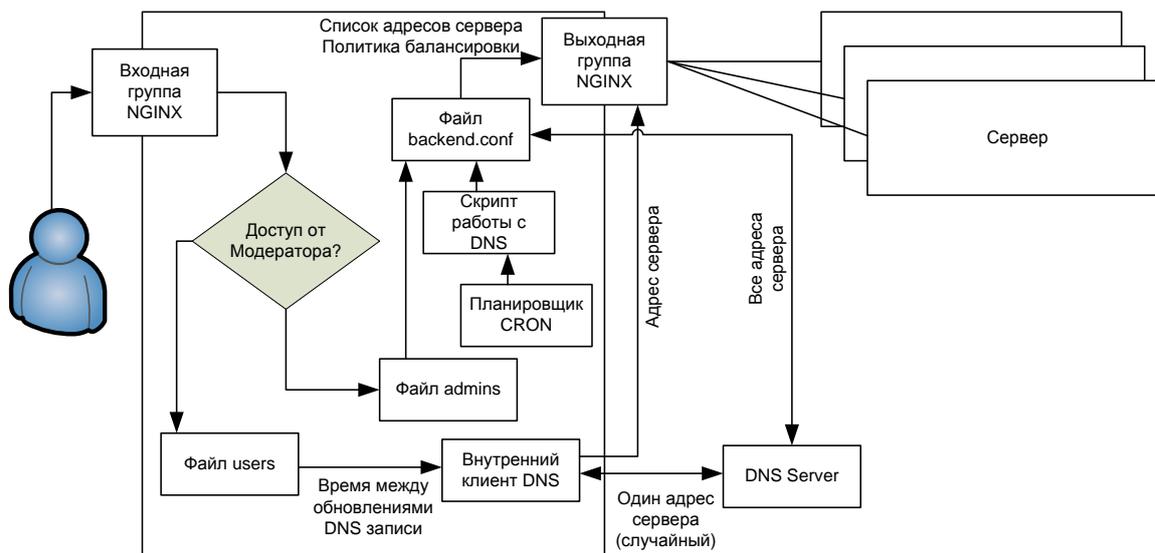


Рисунок 7.1 – Схема работы одного узла кеширующего обратного прокси

Для работы с конечными пользователями требуется распределять нагрузку равномерно, для этого используется механизм Round-Robin и постоянно обновляемый список адресов серверов. Для этого в настройках устанавливается параметр «resolver 8.8.8.8 valid=5s;», что дает обновление списка каждые 5 секунд.

Сама система масштабирования работает по принципу наблюдения. На всех  $N$  конечных серверах установлен параметр `location /nginx_status { stub_status on;}` для получения статистики по использованию nginx. При превышении среднего количества активных сессий максимального количества пользователей  $U_{max}$  (определяется экспериментально) в течение более 10 минут дается команда на запуск нового сервера. При запуске сервер регистрируется в DNS через обертку API и автоматически попадает в распределение нагрузки и в мониторинг. При существенном снижении нагрузки на серверах до уровня  $2U_{max}/3N$  в течении 10 минут последний запущенный сервер получает команду на выключение, при которой удаляет регистрацию в DNS и выключается.

Если сервер не отвечает на запрос о статусе в течении 2 минут, посылается команда на его перезапуск, а запись в DNS удаляется.

Реализация быстродействующего кэша построена на кэшировании проху запросов nginx на диск, созданный в ОЗУ, что дает самый высокий прирост

производительности. Для этого были экспериментально подобраны следующие параметры nginx:

```
Создан диск в оперативной памяти, для этого была добавлена строка в /etc/fstab  
tmpfs /tmp/ram tmpfs rw, size=1G, nr_inodes=5k, noexec, nodev, nosuid, uid=648,  
gid=648,mode=1700 0 0
```

после чего диск был подключен

```
mkdir /tmp/ram
```

```
mount -a
```

Затем был перенастроен Nginx

Активация кэша на 1Гб

```
proxy_cache_path /tmp/ram levels= keys_zone=cache:10m max_size=1G;
```

```
proxy_ignore_headers Expires Cache-Control;
```

```
proxy_cache_use_stale error timeout invalid_header http_502;
```

В настройках основного сервера был включен gzip

```
gzip on;
```

```
gzip_min_length 10240;
```

```
gzip_proxied expired no-cache no-store private auth;
```

```
gzip_types text/plain text/css text/xml text/javascript application/x-javascript
```

```
gzip_disable "msie6";
```

В настройках перенаправления был включен кэш со сложносоставным ключем хеширования запроса

```
location / {
```

```
proxy_cache cache;
```

```
proxy_cache_valid any 10m;
```

```
proxy_cache_key
```

```
"$request_method|$http_if_modified_since|$http_if_none_match|$host|$request_uri";
```

```
add_header X-Cache $upstream_cache_status;
```

```
proxy_cache_methods GET HEAD POST;
```

```
proxy_no_cache $http_pragma $http_authorization;
```

```
proxy_cache_bypass $cookie_nocache $arg_nocache$arg_comment;
```

```

proxy_pass http://backend;
}

```

Реализация сервера перенаправления на виртуальные машины состоит из двух частей: часть nginx и часть API для управления списком серверов. На рисунке 7.2 показана схема работы узла.

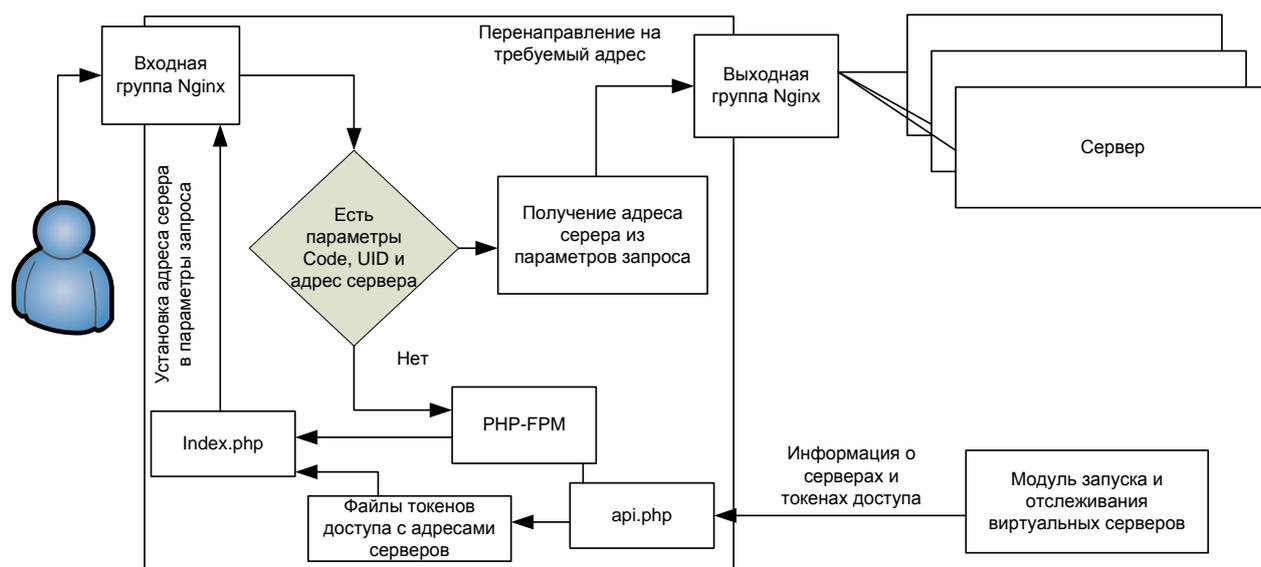


Рисунок 7.2 – Сервер перенаправления на виртуальные машины

Сервер nginx отслеживает параметры запроса, и, если в них есть данные токена доступа и адреса сервера, то происходит перенаправление на нужный сервер, иначе происходит перенаправление на скрипт index.php, который служит для установки этих параметров. Если в запросе присутствует код одноразового запуска или токен сервера, то по этим данным происходит поиск в базе данных токенов и установка параметров запроса. Если данных нет, то сервер перенаправляет пользователя обратно на сайт РЦКД.

Для отслеживания параметров запроса используется встроенный скриптовый язык nginx LUA.

Для различных вариантов развития запроса предусмотрены несколько вариантов перенаправления.

При недоступности сервера (уже или еще выключен, неправильные параметры) происходит перенаправление на скрипт очистки параметров доступа, который после очистки перенаправляет пользователя обратно на сайт РЦКД.

```
error_page 502 =200 /index.php?clear=on;
```

Проверяются параметры одноразовых кодов

```
if ($http_cookie ~* "code=([^;]+)(?::|$)") {  
    set $code $1;  
}
```

А также параметров доступа к серверу

```
if ($http_cookie ~* "sa=([^;]+)(?::|$)") {  
    set $s1 $1;  
}
```

```
if ($http_cookie ~* "ma=([^;]+)(?::|$)") {  
    set $s2 $1;  
}
```

```
if ($http_cookie ~* "na=([^;]+)(?::|$)") {  
    set $s3 $1;  
}
```

```
if ($http_cookie ~* "sc=([^;]+)(?::|$)") {  
    set $s4 $1;  
}
```

Затем после вызова скрипта поиска параметров одноразового кода или токена возвращается \$id сервера и его адрес \$s. После этого происходит перенаправление на нужный сервер

```

if ( $id != " ) {
proxy_pass http://$s;
}

```

Иначе происходит перенаправление на index.php, который решает, что дальше делать.

```

root /var/www/html
location ~ /\.php$ {
    try_files $uri = 404;
    include fastcgi_params;
    fastcgi_pass unix:/var/run/php5-fpm.sock;
    fastcgi_index index.php;

    fastcgi_param SCRIPT_FILENAME $document_root $fastcgi_script_name;
}

```

Скрипт Index.php имеет следующие входные (таблица 7.1) и выходные (таблица 7.2) параметры

Таблица 7.1 – Входные параметры index.php

Параметр	Тип	Возможные значения	Описание
1	2	3	4
clear	Текст, 2 знака	on	Очищает сессию, сбрасывает параметры доступа
code	Цифра, 6 знаков	000000-999999	Выполняет поиск в базе одноразовых кодов, при успехе устанавливает параметры сессии и адрес сервера в запрос
Пустой запрос			Перенаправляет пользователя обратно на РЦКД

Таблица 7.2 – Выходные параметры index.php

Параметр	Тип	Возможные значения	Описание
1	2	3	4
Code	Cookie, число	000000-999999	Одноразовый код
Sa	Cookie, число	0-255	Первый октет сервера
Ma	Cookie, число	0-255	Второй октет сервера
Na	Cookie, число	0-255	Третий октет сервера
Sc	Cookie, число	0-255	Четвертый октет сервера
C	Cookie, строка		Токен доступа
Location	Заголовок, URL		Перенаправление

Для api.php доступны следующие входные (таблица 7.3) и выходные (таблица 7.4) параметры.

Таблица 7.3 – Входные параметры api.php

Параметр	Тип	Возможные значения	Описание
1	2	3	4
Ip	Строка, 15 знаков	0.0.0.0-255.255.255.255	Адрес сервера
Act	Строка, 3 знака	add,del	Режим операции
Code	Число, 6 знаков	000000-999999	Одноразовый код
Id	Строка		Токен
Key	Строка		Ключ авторизации API

Таблица 7.4 – Выходные параметры api.php

Параметр	Тип	Возможные значения	Описание
1	2	3	4
Тип HTTP ответа	Заголовок HTTP	200, 404, 400	Ответы REST сервиса о результатах выполнения
Файл токена	Файл		Файл с сохраненными параметрами токена и сервера

## **7.2 Модуль методического электронного образовательного центра**

Основное назначение модуля – предоставление доступа к учебно-методическим материалам через Web-интерфейс.

Пользователи модуля:

а) ученики и учителя – чтение, загрузка, скачивание материалов без авторизации;

б) модераторы – создание структуры тем и разделов, подготовка и публикация материалов;

в) администратор – управление модулем, работа с БД МЭОЦ.

Модуль реализован на языке Python 3.5 с использованием каркаса Django 1.10.

## **7.3 Модуль подсистемы доступа к виртуальным серверам**

Основное назначение модуля – предоставление Web-интерфейса для формирования и работы с виртуальными окружениями.

Основные пользователи и их функциональные возможности:

а) ученики – получение доступа к виртуальным окружениям по коду. Сами виртуальные окружения запускаются учителями по требованию или по заранее сформированному расписанию;

б) учителя – просмотр расписания, получения кода доступа, запуск и отмена занятия по требованию. Один и тот же учитель может работать в нескольких образовательных организациях и при необходимости переключаться между ними;

в) координаторы – формирование списка учителей, классов школьников, расписания звонков, шаблона расписания и расписания на текущую неделю, заполнение данных об образовательной организации. Координатор может управлять несколькими образовательными организациями и переключаться между ними;

г) администраторы – создание координаторов, образовательных организаций, работа с БД.

Модуль реализован на языке Python 3.5 с использованием каркаса Django 1.10

## 7.4 Модуль запуска и отслеживания виртуальных серверов

Основное назначение модуля – запуск, выключение и удаление виртуальных серверов в облачном ЦОД публичного облачного провайдера с целью проведения занятий по требованию или по расписанию.

Пользователи не работают непосредственно с данным модулем, с ним через систему Celery взаимодействует модуль подсистемы доступа к виртуальным серверам.

Модуль написан на языке Python 3.5, для реализации своих функций он использует модуль обертки над API публичного облачного провайдера.

Основным классом модуля является CloudLauncher, который содержит методы, описанные в таблице 7.5.

Таблица 7.5 – Описание основных статических методов класса CloudLauncher.

Описание статического метода	Назначение
1	2
<code>def __get_sha1(s: str) -&gt; str:</code>	Вспомогательный приватный метод вычисления значения хеш-функции для SHA1 для строки s
<code>def __notify_changes_to_nginx(action: str, ip: str, code: str, session_id: int = 1) -&gt; None</code>	Вспомогательный приватный метод, используемый для оповещения модуля динамической настройки nginx с помощью GET-запроса о запуске или выключении/удалении виртуального сервера с виртуальными окружениями. Расшифровка параметров: action – действие (“add” или “del”); ip – IP-адрес виртуальной машины; code – код доступа к виртуальной машине; session_id – идентификатор сессии.

Продолжение таблицы 7.5

1	2
<pre>def __notify_added_server_to_nginx(ip: str, code: str) -&gt; None</pre>	<p>Вспомогательный приватный метод, используемый для оповещения модуля динамической настройки nginx с помощью GET-запроса о запуске виртуального сервера.</p> <p>Расшифровка параметров:  ip – IP-адрес виртуальной машины;  code – код доступа к виртуальной машине.</p>
<pre>def __notify_deleted_or_stopped_server_to_nginx(ip: str, code: str) -&gt; None</pre>	<p>Вспомогательный приватный метод, используемый для оповещения модуля динамической настройки nginx с помощью GET-запроса об остановке или удалении виртуального сервера.</p> <p>Расшифровка параметров:  ip – IP-адрес виртуальной машины;  code – код доступа к виртуальной машине.</p>
<pre>def __send_json_to_nginx(json_str: str) -&gt; None:</pre>	<p>Вспомогательный приватный метод отправки POST-запроса с телом, содержащим строку json_str модулю динамической настройки nginx.</p>
<pre>def __notify_all_virtual_server_states_to_nginx() -&gt; None</pre>	<p>Вспомогательный приватный метод отправки POST-запроса с телом, содержащим строку json модулю динамической настройки nginx. В теле описывается состояние всех виртуальных серверов.</p>
<pre>def __update_db_server_from_provider(db_server: VirtualServer, provider_server: Server) -&gt; None</pre>	<p>Вспомогательный приватный метод обновления статуса виртуального сервера db_server в БД на основе объекта provider_server, описывающего состояние сервера в облачном ЦОД провайдера.</p>

Продолжение таблицы 7.5

1	2
<pre>def __create_server(disk_image: DiskImage, pc_count: int, school_class: SchoolClass = None, subject: Subject = None) -&gt; Server</pre>	<p>Вспомогательный приватный метод создания виртуального сервера в ЦОД провайдера и его представления в БД на основе образа disk_image. pc_count задает кол-во виртуальных окружений, school_class – класс школьников, для которых создается образ и subject – предмет. Результатом работы метода будет объект сервера провайдера.</p>
<pre>def __launch_on_demand_request(request: OnDemandRequest) -&gt; None</pre>	<p>Вспомогательный приватный метод запуска запроса на проведение занятий по требованию. Параметром request является запрос.</p>
<pre>def __update_all_on_demand_requests() -&gt; None</pre>	<p>Вспомогательный приватный метод, осуществляющий обновление статусов всех запросов по требованию и задействованных виртуальных серверов.</p>
<pre>def __update_all_schedule_items() -&gt; None</pre>	<p>Вспомогательный приватный метод, осуществляющий обновление статусов запланированных пунктов расписания и задействованных виртуальных серверов.</p>
<pre>def set_provider(api_name: str) -&gt; None</pre>	<p>Публичный метод, задающий облачного провайдера по его наименованию api_name.</p>
<pre>def synchronize_images() -&gt; None</pre>	<p>Публичный метод, осуществляющий синхронизацию информации о дисковых образах между ЦОД и БД.</p>
<pre>def synchronize_servers_from_provider_ to_db()</pre>	<p>Публичный метод, осуществляющий применение изменений с виртуальными серверами в ЦОД к БД.</p>
<pre>def launch_on_demand_request(request_id: int) -&gt; None</pre>	<p>Публичный метод запуска запроса на проведения занятия по требованию по его идентификатору request_id.</p>
<pre>def cancel_on_demand_request(request_id: int) -&gt; None</pre>	<p>Публичный метод отмены запроса на проведение занятия по требованию по его идентификатору request_id.</p>
<pre>def update()-&gt; None</pre>	<p>Публичный метод обновления статусов заявок на проведение занятий по требованию и запланированных пунктов расписания.</p>
<pre>def delete_all_launched_servers() -&gt; None</pre>	<p>Публичный метод, удаляющий все запущенные сервера в ЦОД.</p>

## **7.5 Модуль обертки над API публичного облачного провайдера**

Основное назначение модуля – работа с API облачного провайдера.

Содержит следующие основные классы:

- а) Image – абстрактный класс дискового образа виртуального сервера;
- б) OneCloudImage – реализация класса Image для OneCloud API;
- в) Server – абстрактный класс виртуального сервера;
- г) OneCloudServerAction – класс действия сервера для OneCloud API;
- д) OneCloudServer – реализация класса Server для OneCloud API;
- е) CloudAPI – абстрактный класс клиента для работы с API провайдера;
- ж) OneCloudApi – реализация класса CloudAPI для OneCloud API.

## 8 Создание подсистемы совместного доступа к виртуальным урокам, включающим учебные, текстовые и видеоматериалы

Создана подсистема совместного доступа к виртуальным урокам, включающим учебные, текстовые и видеоматериалы, которая носит название Методического электронного образовательного центра (МЭОЦ), Описание модуля МЭОЦ представлено выше в разделе 7.2.

Методический электронный образовательный центр (МЭОЦ) – портал, направленный на повышение качества образования, через который предоставляется доступ к учебно-методическим материалам, рекомендованным Министерством образования Оренбургской области и ОГУ. Схема доступа к portalу представлена на рисунке 8.1.

Особенности портала:

- доступ к материалам для проведения уроков по всем предметам для всех классов;
- наглядная визуализация учебных материалов – видео, графика, инфографика, схемы, текст, презентация;
- простота работы – получение материалов в три клика, умный поиск.

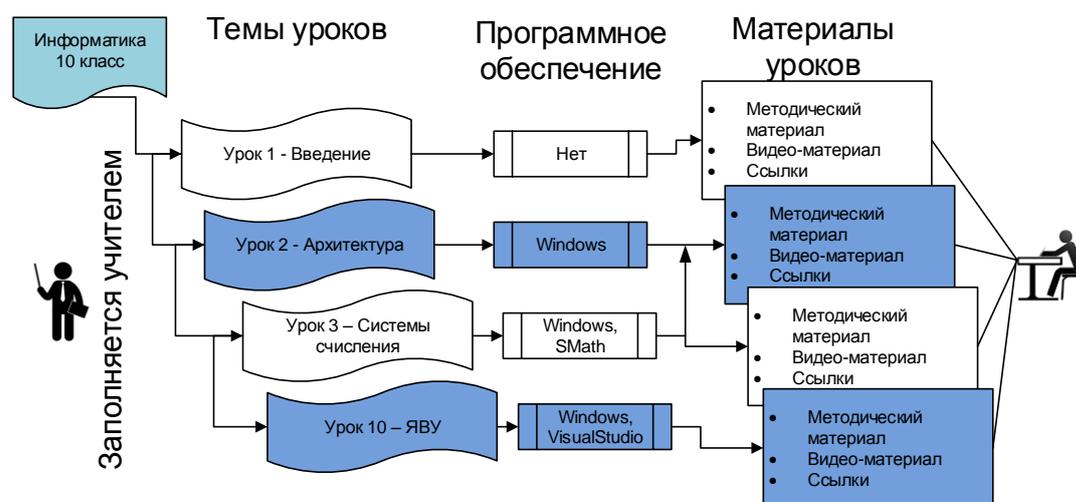


Рисунок 8.1 – Схема доступа к ресурсам МЭОЦ

Вся методическая информация, представленная на портале, распределяется по учебным предметам для каждого года обучения. Основной точкой входа служит таблица методических материалов (рисунок 8.2).

Таблица методических материалов		<a href="#">Добавить методический материал</a>										
	1 класс	2 класс	3 класс	4 класс	5 класс	6 класс	7 класс	8 класс	9 класс	10 класс	11 класс	
<a href="#">Русский язык</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	
<a href="#">Математика</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>						
<a href="#">Алгебра</a>							<a href="#">Открыть</a>					
<a href="#">Физика</a>							<a href="#">Открыть</a>					
<a href="#">Геометрия</a>							<a href="#">Открыть</a>					
<a href="#">Химия</a>								<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	
<a href="#">География</a>					<a href="#">Открыть</a>							
<a href="#">Биология</a>					<a href="#">Открыть</a>							
<a href="#">История</a>					<a href="#">Открыть</a>							
<a href="#">Литература</a>					<a href="#">Открыть</a>							
<a href="#">Информатика</a>		<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	
<a href="#">Обществознание</a>								<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	<a href="#">Открыть</a>	

Рисунок 8.2 – Таблица методических материалов

Внутри каждого предмета для заданного года обучения с первого по одиннадцатый класс формируются разделы, каждый из которых разбивается на темы (рисунок 8.3).

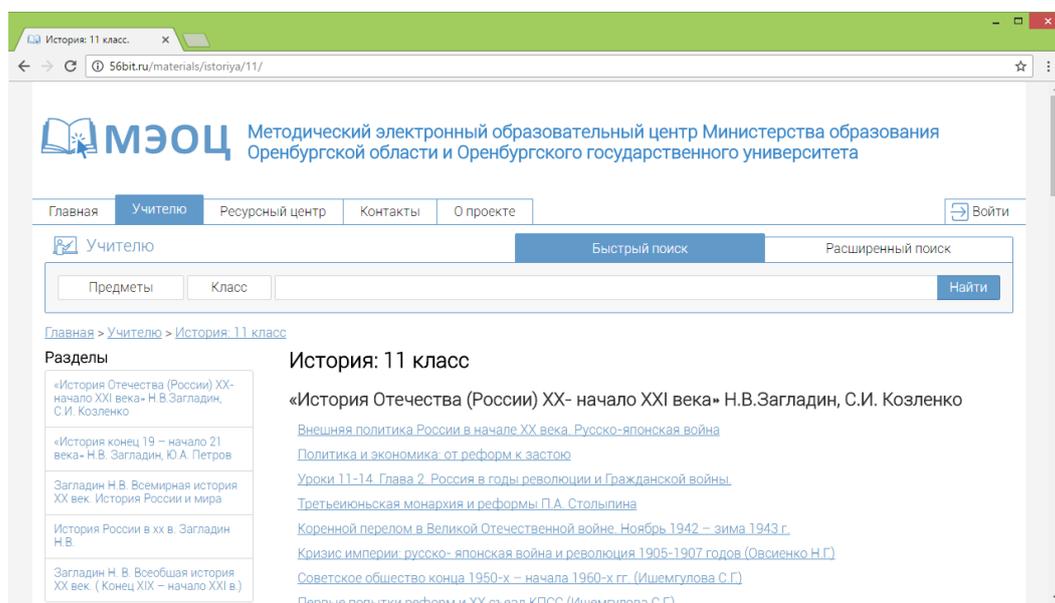


Рисунок 8.3 – Списки разделов и тем

Раздел обычно представляет собой главу учебника, в имени раздела указываются фамилии авторов учебника.

Каждая тема обычно соответствует одному уроку в соответствии с типовым поурочным планированием.

Внутри каждой темы доступен определенный набор методических материалов, классифицированных по типу: текст, презентация, видео, вопросы, задания и другие материалы, по тем или иным причинам не попавшие ни в один раздел материалов. Для удобной навигации между различными типами можно воспользоваться левым меню (рисунок 8.4). Допустимые форматы размещаемых файлов представлены в таблице 8.1.

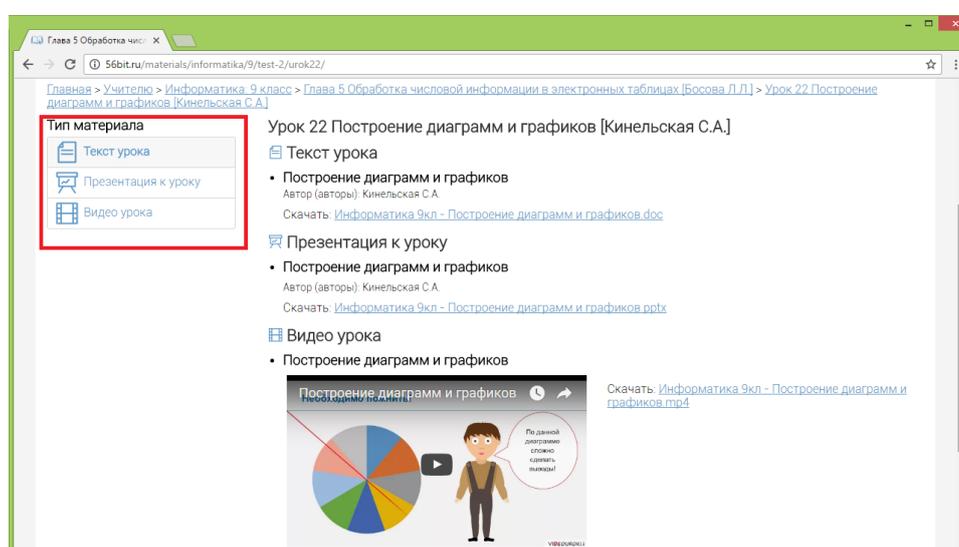


Рисунок 8.4 – Тема урока и методические материалы

Таблица 8.1 – Допустимые форматы файлов

Тип материала	Формат файла	Максимальный размер
Текст урока Вопросы к уроку Задания к уроку	doc, docx, rtf, pdf, odt, txt, xls,xlsx, djvu	100 мегабайт
Видео урока	mov, mp4, mpg4, mpeg4, avi, mkv, wmv, ps, flv, 3gp, webm	700 мегабайт
Презентация к уроку	ppt, pptx, pdf	300 мегабайт
Другой	zip, 7z, rar, mp3, wav, wma	500 мегабайт

Поиск на портале реализован как по названию, так и по полному тексту материалов. Поддерживается технология умного поиска, сразу после ввода слова

выводится список тем, материалы которых содержат заданное слово. Ввод следующих слов уточняет поиск. Поддерживается технология ранжирования, которая выдвигает вперед наиболее релевантные результаты.

Для удобства пользователей портал запоминает несколько последних просмотренных тем, так что к ним всегда можно быстро вернуться.

Видеоматериалы после загрузки на сайт портала автоматически размещаются на видеохостинге youtube.com для удобного просмотра на странице.

Портал поддерживает автоматическое переименование размещаемых материалов в соответствии с названием темы.

Отдельные вопросы реализации и использования портала связаны с интеллектуальными правами на размещаемые методические материалы.

Во-первых, авторы материалов в письменном виде подтверждают свое согласие на обработку персональных данных и размещение материалов в сети Интернет.

Во-вторых, в соответствии со статьей 1253.1 Гражданского кодекса администрация портала является информационным посредником и не несет ответственность за нарушение интеллектуальных прав, поскольку не является инициатором передачи информации и не определяет получателя указанного материала; не изменяет указанный материал при оказании услуг связи, за исключением изменений, осуществляемых для обеспечения технологического процесса передачи материала; а также не знает и не должна знать о неправомерности использования интеллектуальных прав.

В-третьих, в соответствии со статьей 1274 Гражданского кодекса РФ использование материалов в образовательных целях при сохранении авторства не требует согласия автора.

Администраторы портала имеют возможность в любой момент скрыть или удалить материалы, принадлежащие определенному автору, например, если последний отзовет свое согласие на размещение материалов в сети Интернет.

Таким образом, разработанный портал полностью соответствует требованиям Министерства образования Оренбургской области.

## **9 Разработка типовых конфигураций виртуальных классов для предмета «Информатика» на основе используемых УМК с учетом необходимого ПО**

Для опытной эксплуатации РЦКД необходима разработка типовых конфигураций (дисковых образов) по школьным предметам. В рамках первой версии проекта было принято решение создать образы для использования на уроках информатики в 10-11 классах.

Выбор предмета «информатика» обусловлен следующими причинами. Во-первых, большинство программных продуктов используется именно на уроках информатики. Во-вторых, координаторами образовательных организаций при тестовой эксплуатации портала стали именно учителя информатики. В-третьих, для преподавания информатики имеются качественные бесплатные программные продукты, что позволило на этапе тестовой эксплуатации не иметь проблем с закупкой и использованием платных лицензий.

В качестве УМК по базовому курсу информатики для 10-11 классов выбран УМК И.Г. Семакина, Е.К. Хеннера, Т.Ю. Шеиной [25]. Данный УМК разработан в полном соответствии с требованиями Федерального государственного образовательного стандарта среднего (полного) общего образования (ФГОС). Согласно разделу ФГОС 18.3.1. в состав обязательной для изучения предметной области «Математика и информатика» входит учебный предмет «Информатика» (базовый и углубленный уровни).

Кратко представим основные разделы УМК. Задачи из разных предметных областей решаются в разделе «Информационное моделирование». Содержательную линию «информационные технологии» развивают темы, связанные с проектированием и разработкой многотабличных баз данных и приложений к ним, на основе реальных производственных информационных систем. В разделе, посвященном сетям и Интернету, знакомятся с основами сайтостроения. Значительное место в содержании курса занимает линия алгоритмизации и программирования. Новым элементом является знакомство с основами теории

алгоритмов. Углубляются знания учеников языка программирования (в учебнике рассматривается язык Паскаль), развиваются умения и навыки решения на компьютере типовых задач обработки информации путем программирования.

Содержание учебника инвариантно к типу ПК и программного обеспечения. Из 18 работ практикума для 10 класса непосредственную ориентацию на тип ПК и ПО имеют лишь две работы: «Выбор конфигурации компьютера» и «Настройка BIOS». Для выполнения практических заданий по программированию может использоваться любой вариант свободно распространяемой системы программирования на Паскале (Pascal ABC, Free Pascal и др.).

Для выполнения практических заданий на работу с информационными технологиями в 11 классе могут использоваться различные варианты программного обеспечения: свободного/ из списка приобретаемых школами бесплатно/ другого. В учебнике, в разделе, посвященном разработке сайтов, дается описание конструктора сайтов KompoZer (СПО). Непосредственно в практикуме присутствует описание работы с реляционной СУБД LibreOffice Base, также относящейся к свободно распространяемому программному обеспечению. В качестве ПО для моделирования используется табличный процессор Microsoft Excel. При необходимости задания этих двух разделов могут быть выполнены с использованием других аналогичных программных средств: реляционной СУБД и табличного процессора.

УМК И.Г. Семакина по углубленному курсу информатики дополнительно включает изучение объектно-ориентированного программирования, что возможно с использованием среды Lazarus или Pascal ABC.Net.

Альтернативный УМК по углубленному курсу информатики – УМК Полякова К.Ю., Еремина Е.А. [26]. Целевая аудитория УМК – «школьники старших классов, которые планируют связать свою будущую профессиональную деятельность с информационными технологиями» [26]. Учебник ориентирован на получение фундаментальных знаний, умений и навыков, которые не зависят от операционной системы и другого программного обеспечения, применяемого на уроках. УМК предназначен для углубленного изучения всех основных разделов курса

информатики учащимися информационно-технологического и физико-математического профилей.

УМК включает в себя следующие содержательные линии:

- а) информация и информационные процессы;
- б) кодирование информации;
- в) основы логики;
- г) устройство и программное обеспечение компьютеров;
- д) компьютерные системы телекоммуникации;
- е) моделирование;
- ж) алгоритмизация и программирование;
- з) информационные технологии.

Необходимое программное обеспечение – офисный пакет LibreOffice, СУБД LibreOffice Base, система 3d моделирования Blender, система математического моделирования GeoGebra.

В УМК предлагается широкий выбор языков и сред программирования. Обязательный раздел посвящен объектно-ориентированному программированию. Поэтому для языка Pascal необходимо использовать Lazarus или Pascal ABC.Net. Для языков C# и C++ используется среда программирования Microsoft Visual Studio Express Edition или Microsoft Visual Studio Community. Для python используется Python 3.4 и WingIDE.

Анализ необходимого ПО позволил выделить две типовые конфигурации виртуальных машин для использования на уроках информатики.

Офисная конфигурация включает в себя нетребовательные к ресурсам приложения:

- а) свободно распространяемые офисные пакеты Open Office и Libre Office;
- б) свободно распространяемые системы программирования Lazarus, Pascal ABC.Net, Microsoft Visual Studio Community, Wing IDE (Python 3.4);
- в) учебные среды программирования Кумир, Scratch;
- г) свободно распространяемые редакторы растровой и векторной графики Paint.NET, Gimp, InkSpace;

д) свободный редактор HTML KompoZer;

е) мультимедиа конфигурация включает в себя дополнительно ресурсоемкие приложения;

ж) свободно распространяемая система трехмерной графики и анимации Blender;

з) система для компьютерного черчения Компас 3D;

и) видеоредактор Киностудия Windows;

к) система для математического моделирования GeoGebra.

Офисная конфигурация виртуального сервера, рассчитанного на 8 рабочих столов одновременно, требует 4 ГБ памяти и 4 ядра CPU. Мультимедиа конфигурация требует 8 ГБ памяти и 6 ядер CPU.

Разработанные конфигурации успешно использовались на этапе тестовой эксплуатации РЦКД.

## 10 Развертывание и опытная эксплуатация базовой конфигурации РЦКД

Для запуска системы необходимы сервисы, которые будут обслуживать как клиентские запросы, так и инфраструктуру. Первоначально предполагалось использовать только облачные ресурсы, но в дальнейшем от хранения основной информации в облачном расположении было решено отказаться в пользу локального сервера ввиду недостаточных гарантий сохранности и работоспособности при отсутствии пополнения баланса.

Для развертывания системы был реализован план поэтапного запуска и конфигурирования всех сервисов системы (таблица 10.1). Для тестовой эксплуатации МЭОЦ проведено размещение методических материалов, собранных Региональным центром развития образования (РЦРО) в сентябре-октябре 2016 г.

Таблица 10.1 – Этапы развертывания

Этап развертывания	Развертываемые сервисы
1	2
0	Создание учетной записи на стороне поставщика ресурсов. Запрос ключей системы управления (API). Настройка системы автоматического формирования счета для пополнения баланса. Запуск сервера для размещения МЭОЦ. Запуск сервера для резервного копирования материалов МЭОЦ. Размещение службы поиска сервисов масштабирования МЭОЦ. Прикрепление DNS записей МЭОЦ. Формирование и запрос SSL/TLS сертификатов МЭОЦ. Тестовая эксплуатация МЭОЦ.

Продолжение таблицы 10.1

1	2
1	Создание второй учетной записи на стороне поставщика ресурсов для РЦКД. Запрос ключей системы управления (API) на вторую учетную запись. Настройка системы автоматического формирования счета для пополнения баланса. Запуск сервера для размещения обратного прокси и API РЦКД. Размещение службы поиска сервисов масштабирования РЦКД. Прикрепление DNS записей РЦКД. Формирование и запрос SSL/TLS сертификатов РЦКД. Начало наполнения информацией МЭОЦ.
2	Формирование образов с программным обеспечением для РЦКД. Тестирование нагрузки и формирование профилей масштабирования РЦКД. Тестирование нагрузки и формирование профилей масштабирования МЭОЦ. Внедрение системы кеширования CDN для МЭОЦ Тестирование МЭОЦ и РЦКД в выбранных школах. Исправление найденных недочетов, корректировка профилей масштабирования и образов программного обеспечения.

Первая очередь размещения материалов включала разработки по следующим предметам:

- а) информатика 9, 11 классы;
- б) обществознание 9, 10, 11 классы;
- в) история 9, 10, 11 классы;
- г) геометрия 9, 10, 11 классы;
- д) алгебра 9, 11 классы;
- е) физика 9, 11 классы.

Всего за время тестовой эксплуатации с 5 октября по 18 ноября 2016 г. на портале МЭОЦ размещено 1103 урока, которые включают 1301 текстовый документ, 513 презентаций, 100 заданий, 24 видеоролика, 49 других материалов (архивов, рабочих файлов, тестов в формате Flash и др.)

До 1 ноября в рамках тестовой эксплуатации проводилось альфа тестирование, было выявлено и исправлено более 40 ошибок в функционировании системы.

С 1 ноября по 30 ноября проходил этап бета тестирования, когда ссылка на ресурс [www.56bit.ru](http://www.56bit.ru) была предоставлена РЦРО и районным отделам образования. Во время бета тестирования также выявлено и исправлено несколько ошибок.

Для тестовой эксплуатации РЦКД с 1 ноября по 30 ноября 2016 г. подготовлены тестовые образы виртуальных машин, представленные в разделе 9. В систему добавлены координаторы школ (учителя информатики) г. Оренбурга и Оренбургской области в соответствии со следующим перечнем:

- а) физико-математический лицей г. Оренбурга (А.А. Горелик);
- б) губернаторский многопрофильный лицей-интернат для одаренных детей Оренбуржья (О.В. Кондрашова);
- в) лицей №8 г. Оренбурга (И.В. Минина);
- г) гимназия №5 г. Оренбурга (Т.Н. Анисина);
- д) лицей №1 г. Оренбурга (И.В. Войнова).

Всего в процессе тестирования РЦКД было создано 29 занятий по требованию, 62 урока в постоянном расписании. Общее количество потенциальных пользователей системы превысило 300 человек.

В целом, тестовая эксплуатация системы показала ее соответствие выдвинутым функциональным и нефункциональным требованиям, что позволяет считать разработку завершенной и перейти к эксплуатации системы с 1.12.2016 г.

## 11 Экспериментальные исследования эффективности, сбалансированности и производительности сервисов РЦКД

Объектом экспериментальных исследований является РЦКД. Целью экспериментальных исследований является:

- а) подтверждение пригодности РЦКД для опытной эксплуатации;
- б) подтверждение соответствия технических характеристик требованиям эксплуатации;
- в) оценка производительности разработанных программных систем в совокупности с используемым программно-аппаратным обеспечением поставщика ресурсов.

Экспериментальные исследования проводятся на базе Оренбургского государственного университета. Условия проведения экспериментальных исследований должны максимально приближаться к воспроизводимым условиям эксплуатации.

Состав средств, используемых для проведения экспериментальных исследований, показан в таблице 11.1.

Таблица 11.1 – Средства экспериментальных исследований

№ п/п	Обозначение	Наименование	Количество
1	2	3	4
Программные компоненты			
1		Обертка над API публичного провайдера	1
2		Модуль запуска и отслеживания виртуальных серверов	1
3		Модуль динамической настройки nginx	1
4		Методический электронный образовательный центр	2
5		Модуль РЦКД	2
6		Подсистема доступа к виртуальным серверам	2
7		Модуль составления расписания	1

Продолжение таблицы 11.1

1	2	3	4
8		Средство тестирования нагрузочной способности apache ab	1
9		Средство локального сбора статистики использования sar	1
10		Средство удаленного сбора статистики использования snmpd	1
11		Система сбора статистики с удаленных хостов mrtg	
12		Система подготовки статистики на удаленных хостах rrdtool	
13		Обратный прокси-сервер nginx	4
Аппаратные компоненты			
14		Инфраструктура поставщика ресурсов	2
15		Локальный сервер	1
16		Рабочая станция тестировщика	2

Определяемые показатели показаны в соответствии с таблицей 11.2.

Таблица 11.2 – Показатели для экспериментальных исследований.

№ показателя	Наименование показателя
1	2
1	Время выполнения выбранных операций при определенных интенсивностях выполнения этих операций
2	Максимальное количество пользователей, одновременно работающих с экземпляром приложения
3	Время выполнения выбранных операций на высоких, предельных, стрессовых нагрузках
4	Работоспособность приложения при длительном тестировании со средним уровнем нагрузки
5	Коэффициент эффективности использования процессора на одного пользователя системы
6	Коэффициент эффективности использования ОЗУ на одного пользователя системы
7	Коэффициент баланса использования серверов при распределении нагрузки

Методы экспериментальных исследований для определения показателей описаны ниже.

Время выполнения выбранных операций при определенных интенсивностях выполнения этих операций служит для измерения производительности выбранных операций. Для исследования требуется инструмент ab, одна рабочая станция, набор модулей программы. В качестве операций предлагается использовать следующие типовые операции:

- а) просмотр главной страницы;
- б) просмотр таблицы навигации по материалам;
- в) просмотр произвольного списка материалов по дисциплине;
- г) скачивание файла 2.5 Мб (среднее значение);
- д) перенаправление на удаленный рабочий стол.

Для тестирования использовалась следующая методика:

На сервер, где установлены требуемые модули тестирования, формируется POST или GET запрос, который повторяется требуемое количество раз в секунду на протяжении заданного времени. При этом утилита ab измеряет статистические показатели времени отклика (времени выполнения команды). Для запросов GET используется следующий вид запроса:

```
ab -n <Count1> -c <Count2> -C <Cookie> -H "Accept-Encoding: gzip" -r <URL>
```

где <Count1> - общее количество запросов за тестирование,

<Count2> - количество запросов в секунду,

<Cookie> - поля для авторизации при необходимости,

<URL> полный путь до тестируемой функции.

Например, для тестирования вывода таблицы предметов запрос будет выглядеть для пользователя как

```
ab -n 1000 -c 100 -H "Accept-Encoding: gzip" -r http://56bit.ru/materials/
```

а для доступа в закрытые разделы так:

```
ab -n 1000 -c 100 -H "Accept-Encoding: gzip" -r -C im=50 -C ma=227 -C na=16 -C sa=188 -C sc=72 -C code=112233 -C c=1 https://r.56bit.ru/
```

Каждое тестирование будет проводится в двух вариантах – при выключенном и включенном кешировании nginx (там где это применимо).

Максимальное количество пользователей, одновременно работающих с приложением будем определять в расчете на один запущенный экземпляр сервиса с одним процессором и достаточным количеством памяти. Для этого также будем использовать данные тестирования инструмента ab, полученные при тестировании предыдущего параметра.

Время выполнения выбранных операций на высоких, предельных, стрессовых нагрузках замеряется аналогично первому параметру, за исключением того, что запускается до 10 процессов тестирования параллельно и с использованием SSL шифрования, с принудительным выключением кэша. В результате мы получаем существенную нагрузку на процессор (до 50 %) и результаты выполнения в предельных режимах работы.

Работоспособность приложения при длительном тестировании со средним уровнем нагрузки выполняется при количестве пользователей, равном половине от параметра № 2 на время не менее 2 часов. В результате замеряется средний отклик в начале и в конце тестирования, а также размер потребляемой оперативной памяти в начале и в конце тестирования.

Коэффициент эффективности использования процессора на одного пользователя системы определяется при помощи утилит sar и snmp в момент тестирования максимального количества пользователей. Рассчитывается, как отношение загрузки процессора (от 0 до 1) к количеству активных пользователей в системе.

Коэффициент эффективности использования ОЗУ на одного пользователя системы измеряется при помощи утилит sar и snmp в момент тестирования максимального количества пользователей. Рассчитывается, как отношение занятой памяти (в Мб) к количеству активных пользователей в системе.

Коэффициент баланса использования серверов при распределении нагрузки возможно получить только при масштабировании системы. Каждый модуль системы, который может быть масштабирован горизонтально, имеет связь с внешним

трафиком через сервера обратного прокси nginx. Нагрузка на сервера определяется через утилиту snmp в виде количества трафика в секунду. Коэффициент баланса каждого сервера определяется как отношение трафика этого сервера к суммарному трафику всех серверов и является весовым коэффициентом сервера в трафике. Для тестирования необходимо использовать две тестовые станции, на каждой из которой необходимо запускать несколько процессов тестирования, предварительно очистив DNS кеш. Для этого перед запуском теста надо на Linux выполнить команду `/etc/init.d/dns-clean`, на Windows `ipconfig /flushdns`. В этом случае каждый следующий запрос уйдет на другой сервер, как это бы произошло с большим количеством реальных пользователей.

Результаты измерений показаны в таблице 11.3. Все значения является средними показателями по результатам серии измерений из 10 повторений по 1000 тестов в каждом.

Таблица 11.3 – Показатели для экспериментальных исследований.

№	Наименование показателя	Измеренное значение показателя
1	2	3
1	Время выполнения выбранных операций при определенных интенсивностях выполнения этих операций	По операциям без кэша: 1) 6.744 мс 2) 73.393 мс 3) 46.621 мс 4) 1.264 с 5) 3.605 мс По операциям с кэшем: 1) 3.157 мс 2) 3.389 мс 3) 3.662 мс 4) 261,523 мс 5) не применимо

Продолжение таблицы 11.3

1	2	3
2	Максимальное количество пользователей, одновременно работающих с экземпляром приложения	По операциям без кэша: 1) 155 2) 13 3) 22 4) не применимо 5) 282 По операциям с кэшем: 1)1321 2)295 3)270 4) не применимо 5) не применимо
3	Время выполнения выбранных операций на высоких, предельных, стрессовых нагрузках	По операциям без кэша: 1) 28.454 мс 2) 144.272 мс 3) 88.356 мс 4) 281,985 мс 5) 6.825 мс По операциям с кэшем: 1) 15.543 мс 2) 3.389 мс 3) 3.662 мс 4) 261,523 мс 5) не применимо
4	Работоспособность приложения при длительном тестировании со средним уровнем нагрузки	Время выполнения операции №2 начале тестирования 76.031 мс Время выполнения операции №2 конце тестирования 74.344 мс Загрузка памяти в начале тестирования 3980564 Б Загрузка памяти в конце тестирования 3980936 Б
5	Коэффициент эффективности использования процессора на одного пользователя системы	По операциям без кэша: 0,16 По операциям с кэшем: 0,001
6	Коэффициент эффективности использования ОЗУ на одного пользователя системы	По операциям без кэша: 30,27 По операциям с кэшем: 0,122
7	Коэффициент баланса использования серверов при распределении нагрузки	При количестве серверов 2 0,52; 0,48 При количестве серверов 4 0,24; 0,26;0,25; 0,25

В результате экспериментального исследования можно сделать вывод о работоспособности системы, а также о высокой эффективности системы кеширования и балансировки нагрузки.

План вычислительного эксперимента РЦКД с использованием базового набора ПО для школ:

а) установка офисных пакетов (установка производилась в максимальной конфигурации для выбранной версии продуктов Microsoft и для последней версии свободно распространяемого ПО с использованием всех доступных программных модулей и компонентов);

б) установка графических пакетов (для коммерческих графических продуктов использовались ограниченные по времени пробные версии программных продуктов последних версий, для свободно распространяемых продуктов использовались последние актуальные версии);

в) установка браузеров (использовались последние актуальные версии доступные для операционной системы);

г) для каждого пункта плана после установки ПО произведен замер объема дискового пространства, задействованного для размещения.

Для всех видов ПО в каждом эксперименте в процессе работы снятие данных о потребляемых ресурсах проводилось с интервалом 5 минут. Общее время каждого эксперимента 30 минут. Полученные значения усреднялись. Отдельно записывались минимальные и максимальные значения.

Тестовые эксперименты для офисных пакетов:

а) запуск без открытия существующих документов (новый пустой документ);

б) создание эталонных файлов объемом 1, 5, 20 Мб с последующей записью на диск полученного файла;

в) в каждом офисном пакете файлы создавались в трех специализированных ПО (для работы с документами, для работы с таблицами, для работы с презентациями);

г) каждый файл содержал текст, таблицы, формулы, графические изображения;

д) для каждого редактора использовался «родной» формат данных, специфичный для данной версии ПО;

е) открытие одного файла каждого размера и типа для редактирования и последующее сохранение результата;

ж) создание на основе эталонных файлов 10 копий и последующее одновременное открытие для редактирования 10 файлов одного размера (тест на одновременную работу в нескольких документах).

Тестовые эксперименты для графических редакторов для двумерной графики:

а) запуск без открытия существующих документов (новый проект);

б) открытие эталонных файлов объемом 1, 5, 20 Мб с последующей записью на диск полученного файла после редактирования;

в) для каждого графического редактора использовался «родной» формат данных для сохранения результата проекта. В качестве эталонных файлов использовались графические файлы в форматах JPG и PNG;

г) открытие одного файла каждого размера и типа для редактирования и последующее сохранение результата проекта;

д) создание на основе эталонных файлов 10 копий и последующее одновременное открытие для редактирования 10 файлов одного размера (тест на одновременную работу в нескольких документах).

Тестирование проводилось с использованием пакетов Microsoft Office, Open Office, Libre Office, графических редакторов GIMP, Corel Draw, Adobe Photoshop.

Таблица 11.4 - Результаты эксперимента

Эксперимент	Память (МБ)	I/O read (Мб)	I/O write (Мб)	Общее использование памяти (%)	CPU (%)	
1	2	3	4	5	6	
Office 2003 HDD= 986 Мб						
Word 2003						
Запуск нового документа		Min 10,6 Max 20,6 Avg 14,5	2.3	2.3	7%	0.3%
Работа с новым файлом		Min 13,5 Max 25,6 Avg 14,5	2.4	2.4	8%	0.5%
Открытие эталонного файла	1МБ	Min 11,4 Max 21,2 Avg 13,5	2.4	2.4	8%	0.5%
	5МБ	Min 13,3 Max 22,2 Avg 16,5	2.4	2.4	8%	0.5%
	20МБ	Min 12,5 Max 35,7 Avg 18,5	2.7	2.5	8%	0.5%
Одновременное открытие файлов	1МБ	Min 13,5 Max 25,6 Avg 14,5	2.4	2.4	9%	2.5%
	5МБ	Min 13,8 Max 39,5 Avg 16,3	7.2	36	10%	2,6%
	20МБ	Min 16,5 Max 49,5 Avg 21,5	15.3	50	9%	2.7%
Excel 2003						
Запуск нового документа		Min 5,4 Max 20,6 Avg 14,5	6.2	5.3	5%	0.5%
Работа с новым файлом		Min 5,4 Max 20,6 Avg 14,5	6.3	5.4	5%	0.6%

Продолжение таблицы 11.4

1		2	3	4	5	6
Открытие эталонного файла	1МБ	Min 6,4 Max 21,3 Avg 18,3	6.3	5.4	5%	0.6%
	5МБ	Min 6,4 Max 22,3 Avg 16,3	7.3	8.2	5%	0.6%
	20МБ	Min 5,4 Max 32,6 Avg 18,9	12.2	14.2	5%	0.6%
Одновременное открытие файлов	1МБ	Min 5,4 Max 30,6 Avg 18,9	10	33	7%	1.6%
	5МБ	Min 6,4 Max 40,6 Avg 21,2	12	45	7%	1.6%
	20МБ	Min 6,4 Max 50,6 Avg 28,6	20	53	9%	2.6%
PowerPoint 2003						
Запуск нового документа		Min 5,4 Max 20,6 Avg 14,5	15	18	5%	0.7%
Работа с новым файлом		Min 5,4 Max 20,6 Avg 14,5	15.6	19	5%	0.7%
Открытие эталонного файла	1МБ	Min 5,5 Max 21,6 Avg 15,2	16	18	5%	0.7%
	5МБ	Min 6,5 Max 21,6 Avg 17,3	16	18	5%	0.7%
	20МБ	Min 7,7 Max 24,6 Avg 18,2	16	18	6%	0.7%
Одновременное открытие файлов	1МБ	Min 5,5 Max 54,6 Avg 38,2	10	33	7%	6%
	5МБ	Min 5,5 Max 61 Avg 52	20	43	7%	6%
	20МБ	Min 5,5 Max 61 Avg 52	30	52	8%	9%
Office 2007 HDD=1437Mб						

Продолжение таблицы 11.4

1		2	3	4	5	6
Word 2007						
Запуск нового документа		Min 5,4 Max 20,6 Avg 14,5	18	76.3	5%	0.3%
Работа с новым файлом		Min 5,4 Max 20,6 Avg 14,5	18	76.3	5%	0.3%
Открытие эталонного файла	1МБ	Min 6,3 Max 21,2 Avg 15,5	19,1	78.2	5%	0.5%
	5МБ	Min 6,3 Max 21,2 Avg 15,5	19,1	78.2	5%	0.5%
	20МБ	Min 6,3 Max 21,2 Avg 15,5	19,1	78.2	5%	0.5%
Одновременное открытие файлов	1МБ	Min 18,3 Max 29,5 Avg 25,5	18,5	32.3	5%	3.5%
	5МБ	Min 18,3 Max 29,5 Avg 28,3	28.5	36	5%	4.4%
	20МБ	Min 18,3 Max 56,5 Avg 32,3	38.5	36	5%	5.6%
Excel 2007						
Запуск нового документа		Min 5,4 Max 12,6 Avg 14,5	5	6.8	5%	0.3%
Работа с новым файлом		Min 5,4 Max 12,6 Avg 14,5	6	12.4	5%	0.3%

Продолжение таблицы 11.4

1		2	3	4	5	6
Открытие эталонного файла	1МБ	Min 6,3 Max 15,2 Avg 11,5	19,1	12.4	5%	0.5%
	5МБ	Min 6,3 Max 18,2 Avg 13,5	13,1	28.2	5%	0.5%
	20МБ	Min 6,3 Max 21,2 Avg 15,5	29,6	48.2	5%	0.5%
Одновременное открытие файлов	1МБ	Min 6,3 Max 30,6 Avg 21,2	11	33	6%	6%
	5МБ	Min 6,3 Max 21,2 Avg 25,5	31	43	6%	7%
	20МБ	Min 6,3 Max 21,2 Avg 35,5	41	53	7%	8%
PowerPoint 2007						
Запуск нового документа		Min 5,4 Max 20,6 Avg 14,5	15	18	6%	0,6%
Работа с новым файлом		Min 5,4 Max 20,6 Avg 14,5	15	18	6%	0,6%
Открытие эталонного файла	1МБ	Min 6,4 Max 21,6 Avg 15,2	15,1	19.2	6%	0,6%
	5МБ	Min 8,2 Max 23,6 Avg 16,2	20,1	19.2	6%	0,6%
	20МБ	Min 9,4 Max 28,2 Avg 18,9	25,1	29.5	7%	0,6%
Одновременное открытие файлов	1МБ	Min 26,6 Max 46 Avg 36	29,2	39.5	6%	6%
	5МБ	Min 26,6 Max 61 Avg 36	29,2	39.5	6%	6%
	20МБ	Min 36,6 Max 69 Avg 42	32,2	39.5	6%	6%
Openfiice 4.1.3 986						
Документы						

Продолжение таблицы 11.4

1		2	3	4	5	6
Запуск нового документа		Min 27 Max 45 Avg 40	2.3	2.3	8%	0.3%
Работа с новым файлом		Min 69 Max 100 Avg 99	20.3	15.6	8%	4.3%
Открытие эталонного файла	1МБ	Min 69 Max 100 Avg 99	20.3	15.6	8%	4.3%
	5МБ	Min 69 Max 100 Avg 99	21.2	17.6	8%	4.3%
	20МБ	Min 69 Max 100 Avg 99	25.1	14.6	8%	4.3%
Одновременное открытие файлов	1	Min 69 Max 100 Avg 99	22.1	13.6	8%	4.3%
	5	Min 69 Max 100 Avg 99	22.3	17.6	8%	7.4%
	20	Min 69 Max 140 Avg 129	20.3	15.6	8%	9.3%
Таблицы						
Запуск нового документа		Min 5,4 Max 20,6 Avg 14,5	20.3	15.6	8%	0.3%
Работа с новым файлом		Min 5,4 Max 20,6 Avg 14,5	20.3	15.6	8%	4.5%
Открытие эталонного файла	1МБ	Min 5,4 Max 20,6 Avg 14,5	24.1	16.6	8%	4.6%
	5МБ	Min 5,4 Max 20,6 Avg 14,5	25.1	16.6	8%	4.6%
	20МБ	Min 5,4 Max 20,6 Avg 14,5	28.1	17.6	8%	4.6%

Продолжение таблицы 11.4

1		2	3	4	5	6
Одновременное открытие файлов	1МБ	Min 15,4 Max 30,6 Avg 22,5	10	33	8%	6%
	5МБ	Min 15,4 Max 30,6 Avg 22,5	16	54	8%	6,8%
	20МБ	Min 15,4 Max 30,6 Avg 22,5	12	69	9%	7.4%
Презентации						
Запуск нового документа		Min 5,4 Max 20,6 Avg 14,5	12	58	9%	4.6%
Работа с новым файлом		Min 5,5 Max 21,6 Avg 15,2	14	16	9%	4.6%
Открытие эталонного файла	1МБ	Min 5,5 Max 21,6 Avg 15,2	14	18	9%	4.6%
	5МБ	Min 15,5 Max 25,6 Avg 17,2	14	18	9%	4.6%
	20МБ	Min 18,5 Max 29,6 Avg 35,2	14	26	9%	6%
Одновременное открытие файлов	1МБ	Min 25,5 Max 54,6 Avg 30,2	10	33	9%	6%
	5МБ	Min 51,5 Max 64,6 Avg 50,2	14	26	9%	6%
	20МБ	Min 53,5 Max 64,6 Avg 50,2	14	26	9%	6%
LibreOffice 1537						
Документы						

Продолжение таблицы 11.4

1		2	3	4	5	6
Запуск нового документа		Min 15,4 Max 30,6 Avg 22,5	1.8	765.3	8%	0.3%
Работа с новым файлом		Min 15,4 Max 30,6 Avg 22,5	1.8	765.3	8%	0.3%
Открытие эталонного файла	1МБ	Min 15,4 Max 22,6 Avg 18,5	24.1	16.6	8%	8%
	5МБ	Min 16,4 Max 31,6 Avg 26,3	25.1	16.6	8%	8%
	20МБ	Min 5,4 Max 20,6 Avg 14,5	28.1	17.6	8%	8%
Одновременное открытие файлов	1МБ	Min 15,4 Max 20,6 Avg 14,5	24.1	16.6	8%	4.6%
	5МБ	Min 16,4 Max 31,6 Avg 26,3	25.1	16.6	8%	4.6%
	20МБ	Min 16,4 Max 31,6 Avg 26,3	28.1	17.6	8%	4.6%
Таблицы						
Запуск нового документа		Min 5,4 Max 20,6 Avg 14,5	20.3	15.6	8%	0.3%
Работа с новым файлом		Min 5,4 Max 20,6 Avg 14,5	20.3	15.6	8%	4.5%
Открытие эталонного файла	1МБ	Min 5,4 Max 20,6 Avg 14,5	24.1	16.6	8%	4.6%
	5МБ	Min 5,4 Max 20,6 Avg 14,5	25.1	16.6	8%	4.6%
	20МБ	Min 5,4 Max 20,6 Avg 14,5	28.1	17.6	8%	4.6%

Продолжение таблицы 11.4

1		2	3	4	5	6
Одновременное открытие файлов	1МБ	Min 15,4 Max 30,6 Avg 22,5	10	33	8%	6%
	5МБ	Min 15,4 Max 30,6 Avg 22,5	16	54	8%	6,8%
	20МБ	Min 15,4 Max 30,6 Avg 22,5	12	69	9%	7.4%
Презентации						
Запуск нового документа		Min 5,4 Max 20,6 Avg 14,5	12	58	9%	4.6%
Работа с новым файлом		Min 5,5 Max 21,6 Avg 15,2	14	16	9%	4.6%
Открытие эталонного файла	1МБ	Min 5,5 Max 21,6 Avg 15,2	14	18	9%	4.6%
	5МБ	Min 15,5 Max 25,6 Avg 17,2	14	18	9%	4.6%
	20МБ	Min 18,5 Max 29,6 Avg 35,2	14	26	9%	6%
Одновременное открытие файлов	1МБ	Min 25,5 Max 54,6 Avg 30,2	10	33	9%	6%
	5МБ	Min 51,5 Max 64,6 Avg 50,2	14	26	9%	6%
	20МБ	Min 53,5 Max 64,6 Avg 50,2	14	26	9%	6%
GIMP						
Запуск нового документа		Min 158,4 Max 187,6 Avg 164,5	10	33	9%	6%
Работа с новым файлом		Min 158,4 Max 187,6 Avg 164,5	15	23	9%	6%

Продолжение таблицы 11.4

1		2	3	4	5	6
Открытие эталонного файла	1МБ	Min 158,4 Max 187,6 Avg 164,5	15	23	9%	6%
	5МБ	Min 158,4 Max 187,6 Avg 164,5	20	43	9%	6%
	20МБ	Min 158,4 Max 187,6 Avg 164,5	20	43	9%	6%
Одновременное открытие файлов	1МБ	Max 30,6	10	33	9%	6%
	5МБ	Min 158,4 Max 187,6 Avg 164,5	14	153	9%	6%
	20МБ	Min 118,4 Max 137,6 Avg 124,5	50	233	9%	6%
Corel Draw						
Запуск нового документа		Min 58,4 Max 87,6 Avg 64,5	15	18	9%	6%
Работа с новым файлом		Min 58,4 Max 87,6 Avg 64,5	15	18	9%	6%

Продолжение таблицы 11.4

1		2	3	4	5	6
Открытие эталонного файла	1МБ	Min 58,4 Max 87,6 Avg 64,5	15	18	9%	6%
	5МБ	Min 59,4 Max 82,6 Avg 68,5	15	18	9%	6%
	20МБ	Min 63,4 Max 97,6 Avg 74,5	15	18	9%	6%
Одновременное открытие файлов	1МБ	Min 78,4 Max 154,6 Avg 104,5	10	33	9%	6%
	5МБ	Min 78,4 Max 154,6 Avg 104,5	18	36	9%	6%
	20МБ	Min 78,4 Max 154,6 Avg 104,5	20	43	9%	6%
Paint.NET						
Запуск нового документа		Min 5,4 Max 20,6 Avg 14,5	1.8	1.8	3%	0.3%
Работа с новым файлом		Min 5,4 Max 21,6 Avg 16,5	2.8	2.8	3%	0.3%
Открытие эталонного файла	1МБ	Min 5,4 Max 22,2 Avg 17,2	3.4	2.4	3%	0.3%
	5МБ	Min 5,4 Max 22,2 Avg 17,2	13.4	23.4	3%	0.3%
	20МБ	Min 5,4 Max 22,2 Avg 17,2	11.4	22.4	3%	0.3%

Продолжение таблицы 11.4

1		2	3	4	5	6
Одновременное открытие файлов	1МБ	Min 5,4 Max 22,2 Avg 17,2	14.4	22.4	3%	2%
	5МБ	Min 4,4 Max 29,2 Avg 27,2	23.4	25.4	3%	2%
	20МБ	Min 8,4 Max 32,2 Avg 27,2	32.4	22.4	3%	2%
Photoshop						
Запуск нового документа		Min 32,4 Max 72,6 Avg 44,5	25.1	16.6	8%	6%
Работа с новым файлом		Min 20,3 Max 72,6 Avg 42,3	25.1	16.6	8%	6%
Открытие эталонного файла	1МБ	Min 20,4 Max 66,6 Avg 44,5	25.1	16.6	8%	6%
	5МБ	Min 30,4 Max 66,6 Avg 44,5	28.1	36.6	8%	6%
	20МБ	Min 34,4 Max 68,6 Avg 44,5	28.1	36.6	8%	6%
Одновременное открытие файлов	1МБ	Min 29,4 Max 72,5 Avg 48,5	28.1	36.6	8%	6%
	5МБ	Min 122,4 Max 172,6 Avg 141,5	28.1	36.6	8%	6%
	20МБ	Min 131,4 Max 179,6 Avg 149,1	28.1	36.6	8%	6%
Adobe Illustrator						
Запуск нового документа		Min 50,4 Max 72,6 Avg 54,5	25.1	16.6	8%	6%

Продолжение таблицы 11.4

1		2	3	4	5	6
Работа с новым файлом		Min 20,3 Max 72,6 Avg 42,3	25.1	16.6	8%	6%
Открытие эталонного файла	1МБ	Min 20,4 Max 66,6 Avg 44,5	25.1	16.6	8%	6%
	5МБ	Min 30,4 Max 72,6 Avg 44,5	25.1	35.6	8%	6%
	20МБ	Min 54,4 Max 72,6 Avg 44,5	26.2	36.7	8%	6%
Одновременное открытие файлов	1МБ	Min 24,4 Max 72,6 Avg 42,5	25.1	32.6	8%	6%
	5МБ	Min 111,4 Max 132,6 Avg 124,5	23.1	32.6	8%	6%
	20МБ	Min 131,4 Max 162,6 Avg 142,5	22.1	31.6	8%	6%

Результаты экспериментальных исследований позволяют по набору программных продуктов определить параметры ресурсов виртуальных машин.

## 12 Рекомендации по использованию ресурсов РЦКД для органов управления образованием и образовательных организаций

### 12.1 Общая информация

Методический электронный образовательный центр (МЭОЦ) – портал для учителей, предоставляющий доступ к учебно-методическим материалам, рекомендованным Министерством образования Оренбургской области (рисунок 12.1).

Методические материалы, размещенные на портале, включают конспекты уроков, вопросы и задания, презентации, видеоролики и другие авторские материалы, разработанные опытными учителями и прошедшие методическую экспертизу.

Раздел портала «Ресурсный центр коллективного доступа» (РЦКД) позволяет обращаться через сеть Интернет к виртуальному рабочему столу и образовательным программным продуктам, размещенным в облачной системе.

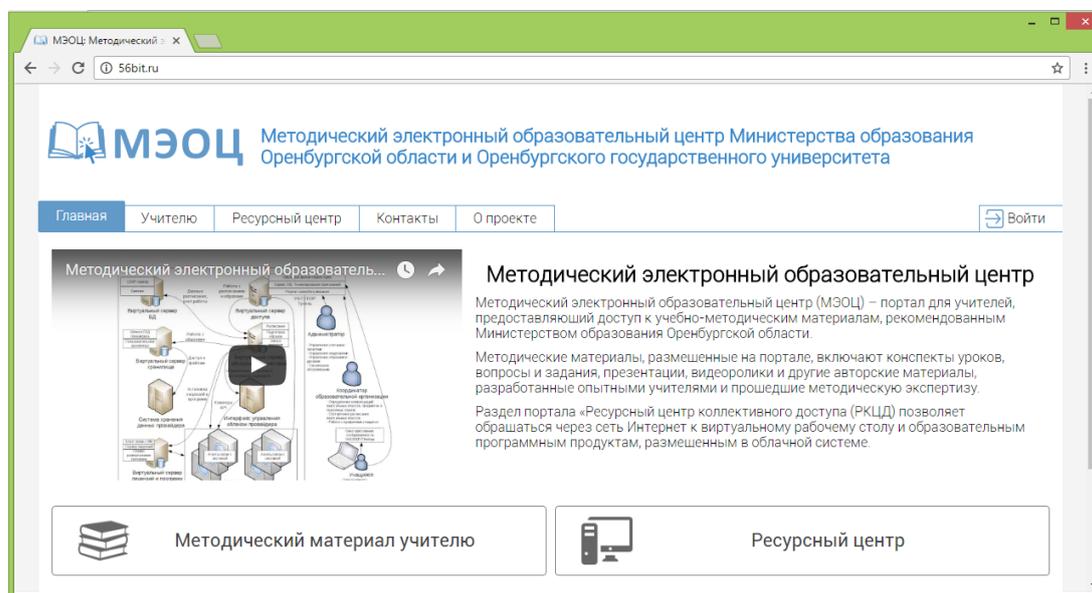


Рисунок 12.1 – Методический электронный образовательный центр

## 12.2 Методический электронный образовательный центр

### 12.2.1 Просмотр методических материалов на портале

Для перехода к разделу необходимо воспользоваться соответствующей ссылкой на главной странице или выбрать пункт меню «Учителю» (рисунок 12.2). Публичный доступ к методическим материалам осуществляется без авторизации и доступен всем желающим.

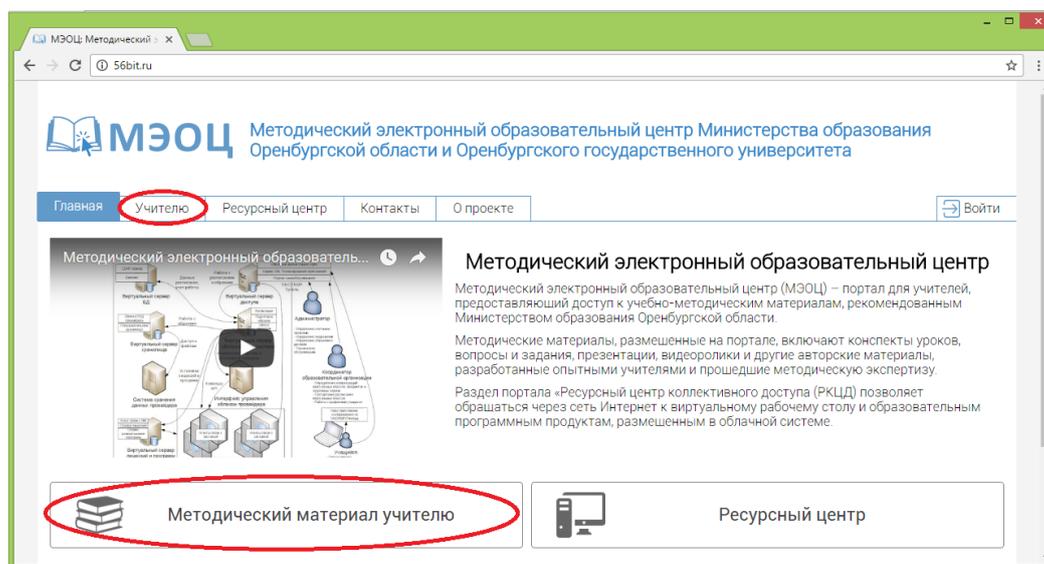


Рисунок 12.2 – Доступ к методическим материалам

Для навигации внутри раздела «Учителю» на портале МЭОЦ предусмотрено несколько вариантов: по предмету с последующим выбором класса, по классу с последующим выбором предмета или с использованием таблицы методических материалов (рисунок 12.3).

Таблица методических материалов позволяет перейти к нужному предмету и классу для ознакомления с доступными ресурсами портала. Переход осуществляется по активной ссылке «Открыть» на пересечении предмета и класса (рисунок 12.4).

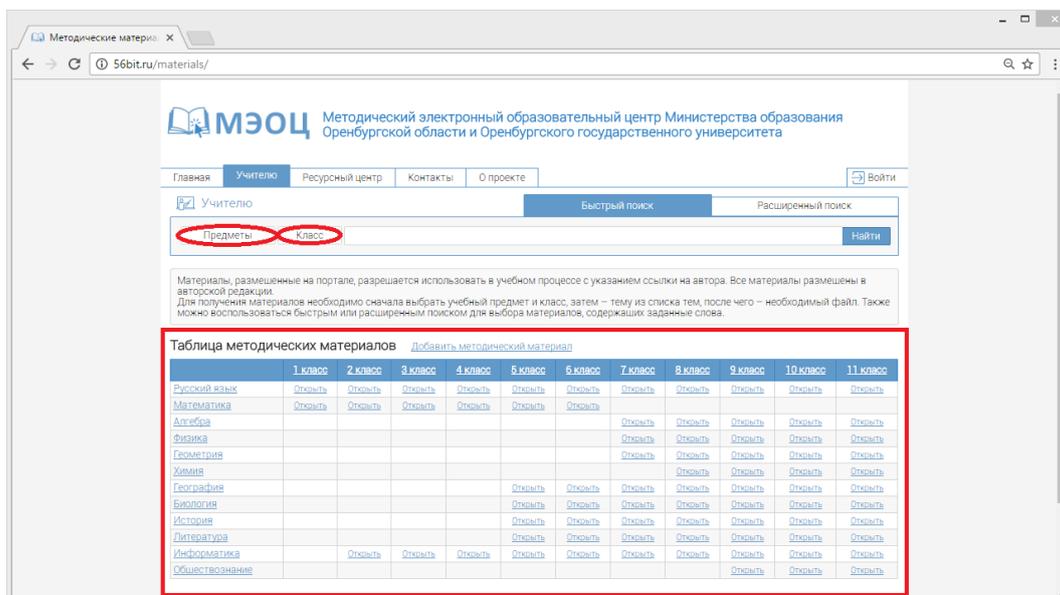


Рисунок 12.3 – Навигация по методическим материалам

Таблица методических материалов позволяет перейти к нужному предмету и классу для ознакомления с доступными ресурсами портала. Переход осуществляется по активной ссылке «Открыть» на пересечении предмета и класса (рисунок 12.4).

Таблица методических материалов

	1 класс	2 класс	3 класс	4 класс	5 класс	6 класс	7 класс	8 класс	9 класс	10 класс	11 класс
Русский язык	Открыть	Открыть									
Математика	Открыть	Открыть									
Алгебра							Открыть	Открыть	Открыть	Открыть	Открыть
Физика							Открыть	Открыть	Открыть	Открыть	Открыть
Геометрия							Открыть	Открыть	Открыть	Открыть	Открыть
Химия							Открыть	Открыть	Открыть	Открыть	Открыть
География					Открыть	Открыть	Открыть	Открыть	Открыть	Открыть	Открыть
Биология					Открыть	Открыть	Открыть	Открыть	Открыть	Открыть	Открыть
История					Открыть	Открыть	Открыть	Открыть	Открыть	Открыть	Открыть
Литература					Открыть	Открыть	Открыть	Открыть	Открыть	Открыть	Открыть
Информатика		Открыть	Открыть								
Обществознание									Открыть	Открыть	Открыть

Рисунок 12.4 – Навигация по таблице методических материалов

После выбора предмета и класса с использованием одного из возможных способов предоставляется доступ к соответствующим методическим материалам. Доступные материалы сгруппированы в разделы, разбитые на темы (рисунок 12.5).

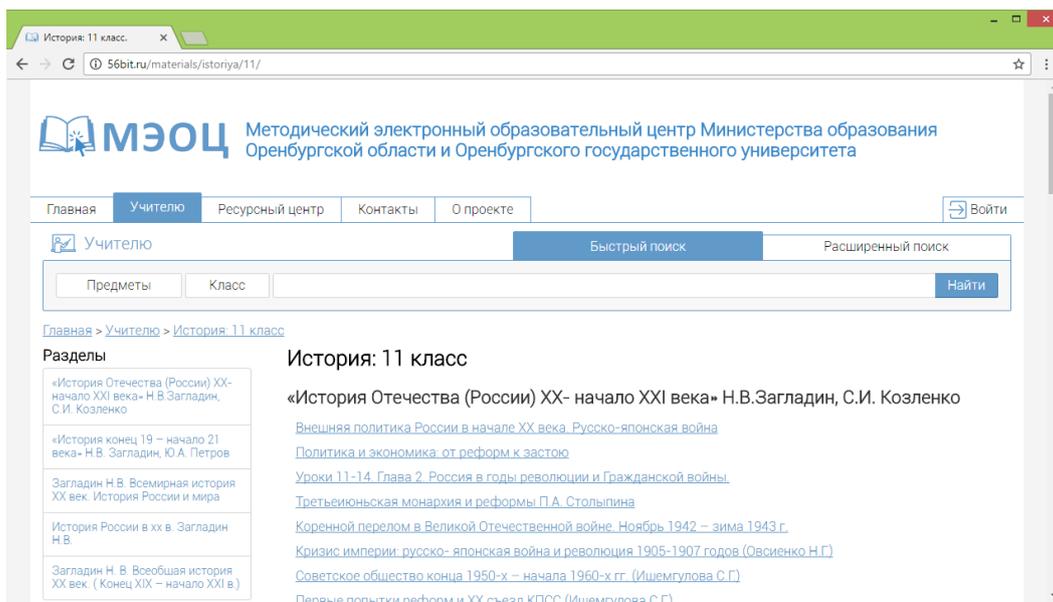


Рисунок 12.5 – Методические материалы

Внутри каждой темы доступен определенный набор методических материалов, классифицированных по типу: текст, презентация, видео, вопросы, задания и другие материалы, по тем или иным причинам не попавшие ни в один раздел материалов. Для удобной навигации между различными типами можно воспользоваться левым меню (рисунок 12.6).

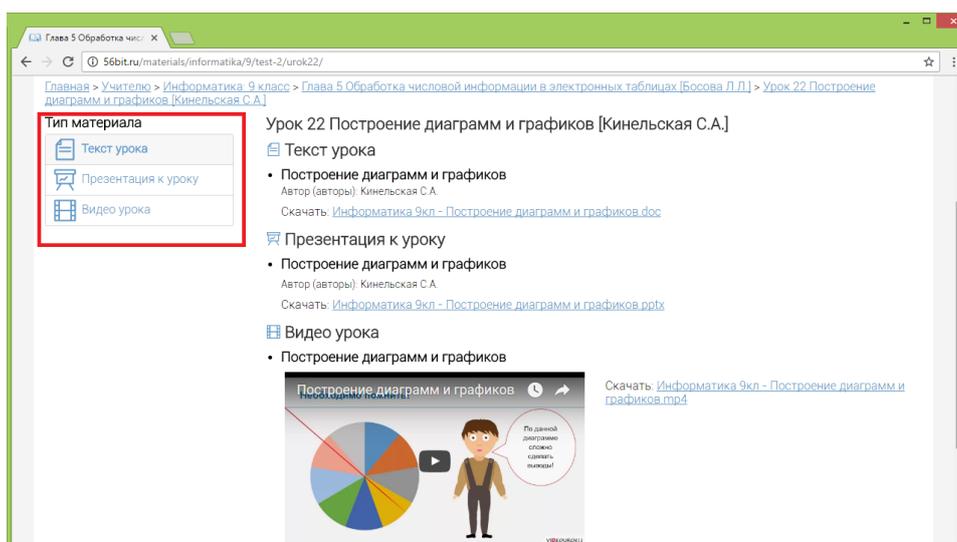


Рисунок 12.6 – Тема урока и методические материалы

Для удобства пользователей на портале МЭОЦ организован поиск по названию темы урока, позволяющий найти необходимый материал без указания предмета и

класса (рисунок 12.7). Расширенный поиск позволяет уточнить критерии и задать тип методического материала, предмет, класс (рисунок 12.8).

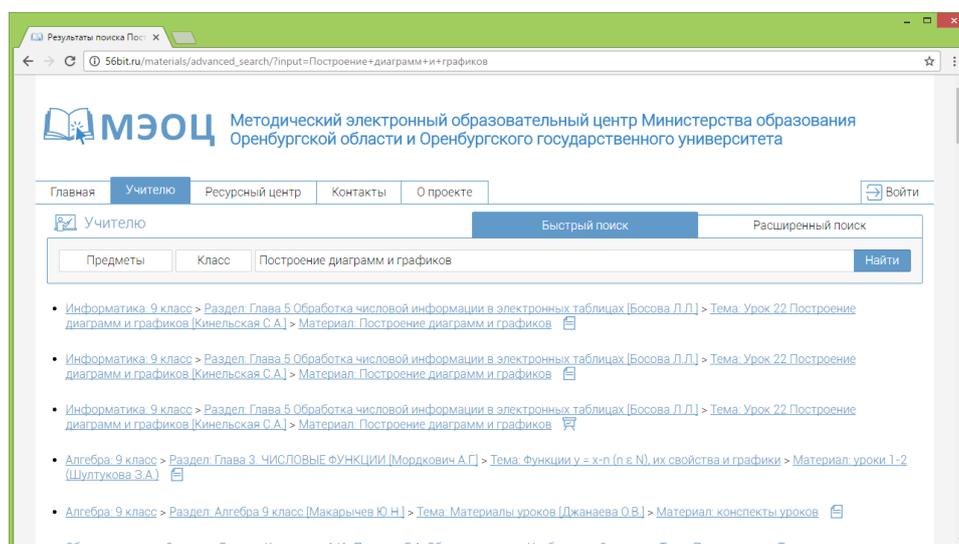


Рисунок 12.7 – Быстрый поиск по portalу МЭОЦ

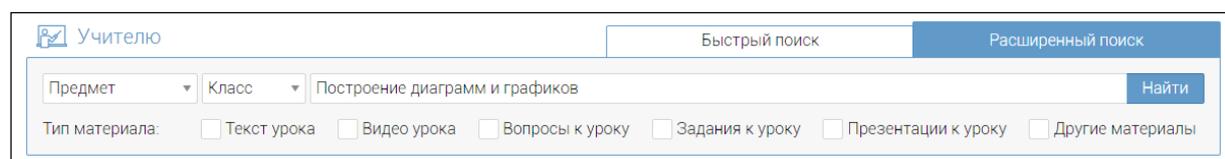


Рисунок 12.8 – Расширенный поиск по portalу МЭОЦ

### 12.2.3 Управление (модерация) методическими материалами на портале

Процедура управления методическими материалами на портале МЭОЦ осуществляется по авторизованному доступу в закрытом разделе портала. Права доступа назначаются администратором портала. Ответственным исполнителям направляется уведомление на адрес электронный почты с указанием логина и пароля.

Для входа в режим модератора необходимо нажать на ссылку «Войти» (рисунок 12.9).

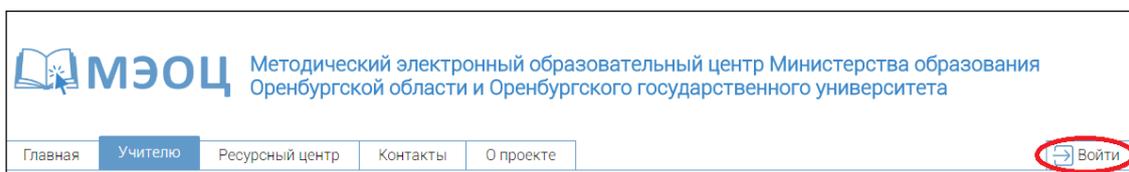


Рисунок 12.9 – Вход в режим модерации методических материалов

Для успешной авторизации необходимо ввести логин и пароль, полученные в электронном письме от администратора портала.

Если Вы забыли пароль или не можете найти письмо, то воспользуйтесь функцией восстановления пароля (рисунок 12.10).

Рисунок 12.10 – Вход в систему портала МЭОЦ

Для восстановления доступа требуется указать адрес электронной почты, которая использовалась для работы на портале (рисунок 12.11).

Рисунок 12.11 – Восстановление доступа

После нажатия кнопки «Отправить сообщение» на указанный адрес электронной почты придет уведомление со ссылкой, по которой необходимо будет

перейти для смены пароля. Обращаем ваше внимание, ссылка действует в течение 30 минут.

#### 12.2.4 Управление структурой методических материалов

Модератор имеет право создавать, редактировать и удалять элементы структуры методических материалов в пределах прав назначенных администратором на предмет и класс.

Для создания раздела необходимо выбрать требуемый предмет и класс. Далее следует выбрать соответствующую пиктограмму в меню «Разделы». В открывшемся окне следует ввести требуемую информацию и нажать кнопку «Создать» (рисунок 12.12).

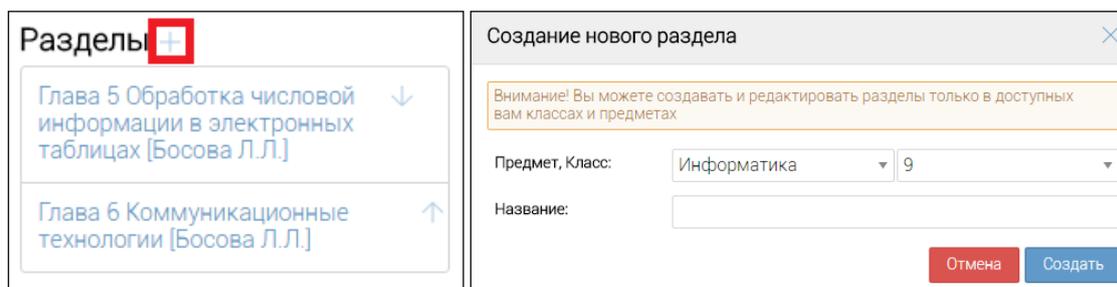


Рисунок 12.12 – Создание раздела

Новые разделы добавляются в конец списка. Если требуется изменить порядок разделов, воспользуйтесь пиктограммами стрелок рядом с названием раздела. Для редактирования, удаления раздела или создания темы воспользуйтесь соответствующими пиктограммами рядом с названием раздела в основной части экрана (рисунок 12.13).

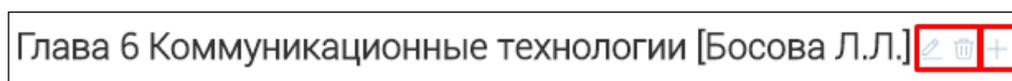
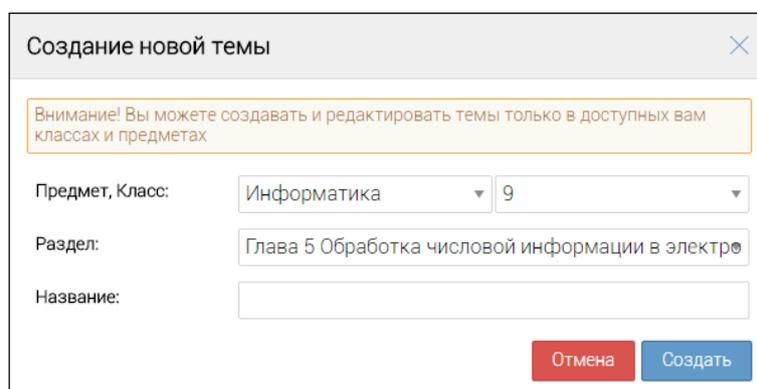


Рисунок 12.13 – Управление разделом, создание темы

Для создания темы заполните требуемую информацию и нажмите кнопку «Создать» (рисунок 12.14).



Создание новой темы

Внимание! Вы можете создавать и редактировать темы только в доступных вам классах и предметах

Предмет, Класс: Информатика 9

Раздел: Глава 5 Обработка числовой информации в электро

Название:

Отмена Создать

Рисунок 12.14 – Создание темы

Редактирование темы происходит аналогичным образом. Для редактирования, удаления или изменения порядка тем в разделе воспользуйтесь соответствующими пиктограммами рядом с названием темы.

### 12.4.3 Управление контентом методических материалов

Модератор имеет право создавать, редактировать и удалять элементы контента методических материалов в пределах прав назначенных администратором на предмет и класс.

Для создания материала необходимо выбрать требуемый предмет и класс. Далее следует выбрать раздел и тему, в которой требуется осуществить добавление материала. Для добавления следует выбрать соответствующую пиктограмму в меню «Тип материала» или рядом с типом материала в основной части окна. В открывшемся окне следует ввести требуемую информацию и нажать кнопку «Создать материал» (рисунок 12.15).

Обязательные поля отмечены звездочкой. Обращайте внимание на подсказки на форме, они помогут осуществить корректный ввод данных.

Редактирование, удаление и изменение порядка следования материала в теме происходит по нажатию на соответствующие пиктограммы рядом с названием материала (рисунок 12.16).

Создание нового материала

Внимание! Вы можете создавать и редактировать материалы только в доступных вам классах и предметах

Название: \*

Аннотация:

Предмет, Класс: \* Информатика 9

Раздел: \* Глава 6 Коммуникационные технологии [Босова Л.Я]

Тема: \* Урок 30 Содержание и структура сайта. Практическое

Тип материала: \* Текст урока

Файл: \*  
 Доступные форматы: doc, docx, rtf, odt, txt, xls, xlsx, djvu  
 Максимальный размер файла: 100 мегабайт  
 + Загрузить файл

Формат добавления нового автора: Фамилия Имя Отчество Email.  
 Наличие пробелов между данными обязательно.

Автор (авторы):

Отмена Создать материал

Рисунок 12.15 – Создание нового материала

- Технологическая карта      
 Автор (авторы): Блинова Т.В.  
 Скачать: [Информатика 9кл - Технологическая карта.docx](#)
- Карта ученика     
 Автор (авторы): Блинова Т.В.  
 Скачать: [Информатика 9кл - Карта ученика.docx](#)
-  Презентация к уроку 

Рисунок 12.16 – Управление контентом материалов

### 12.3 Ресурсный центр коллективного доступа

Для перехода к данному разделу необходимо воспользоваться соответствующей ссылкой на главной странице или выбрать пункт меню «Ресурсный центр» (рисунок 12.17).

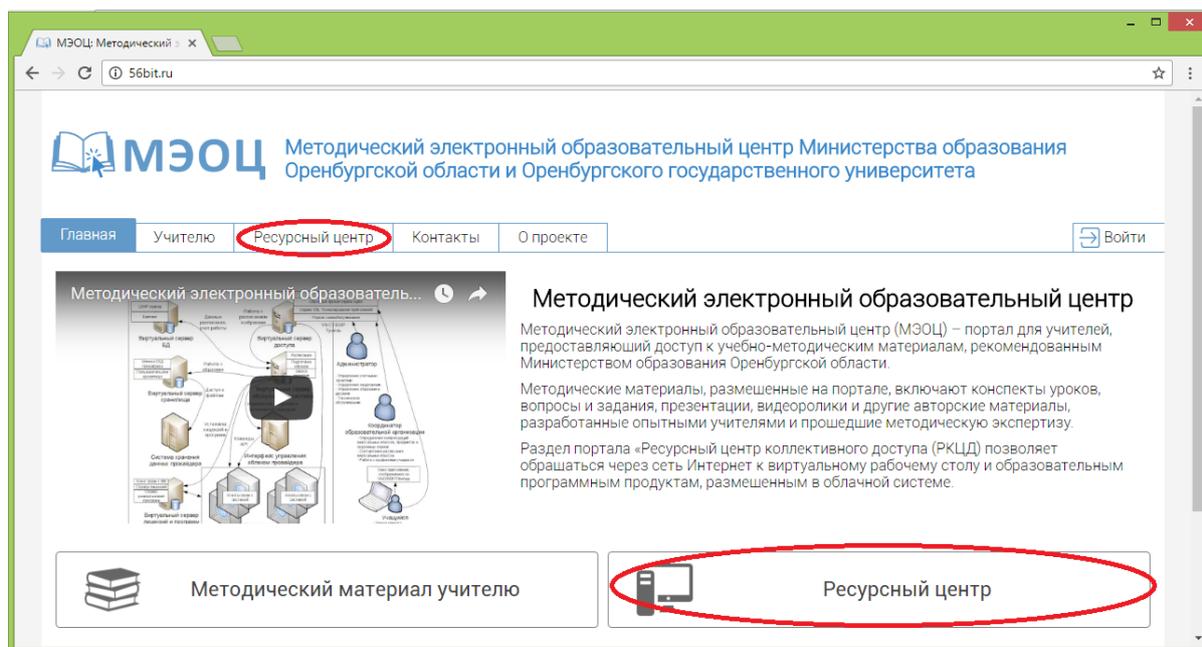


Рисунок 12.17 – Доступ к ресурсному центру коллективного доступа

Для работы на портале РЦКД предусмотрено несколько уровней доступа, соответствующих определенным задачам.

### 12.3.1 Координатор ресурсного центра коллективного доступа

Работа координатора осуществляется на портале РЦКД по авторизованному доступу в закрытом разделе портала. Права доступа назначаются администратором портала. Ответственным исполнителям направляется уведомление на адрес электронный почты с указанием логина и пароля. Для входа в режим координатора необходимо нажать на ссылку «Войти» (рисунок 12.18).

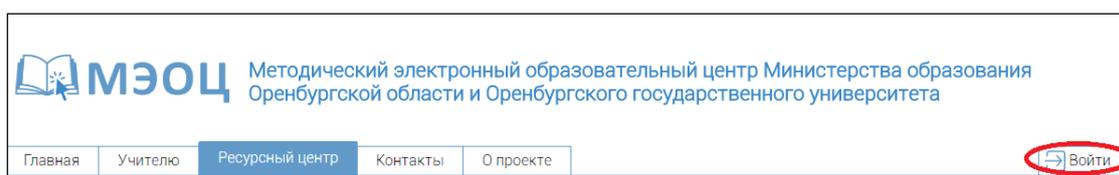


Рисунок 12.18 – Вход в режим координатора РЦКД

Для успешной авторизации необходимо ввести логин и пароль, полученные на электронную почту от администратора портала. Если Вы забыли пароль или не

можете найти письмо, то можете воспользоваться функцией восстановления пароля (рисунок 12.19).

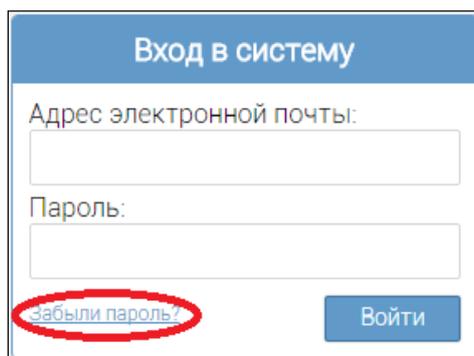


Рисунок 12.19 – Вход в систему портала МЭОЦ

Для восстановления доступа требуется указать адрес электронной почты, которая использовалась для работы на портале (рисунок 12.20).

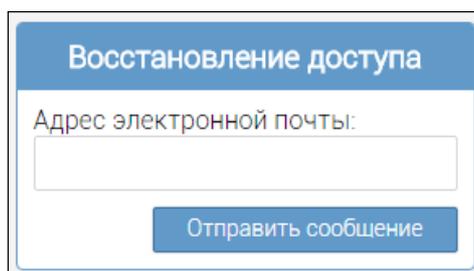


Рисунок 12.20 – Восстановление доступа

После нажатия кнопки «Отправить сообщение» на указанный адрес электронной почты придет уведомление со ссылкой, по которой необходимо будет перейти для смены пароля. Обращаем ваше внимание, ссылка действует в течение 30 минут.

Координатор РЦКД осуществляет администрирование данных на уровне школ. В пределах доступных школ координатор может создавать, редактировать и удалять классы, кабинеты, расписание звонков, учетные записи учителей, составлять шаблоны расписаний для школы в соответствующих разделах портала РЦКД (рисунок 12.21).

### 12.3.1.1 Управление шаблоном расписания для школы

Шаблон расписания составляется для планирования ресурсов РЦКД для конкретной школы. Координатору рекомендуется вносить данные о планируемых уроках в шаблон расписания в начале учебного года (рисунок 12.22). Обратите внимание, все корректировки, внесенные в шаблон расписания, применяются только со следующей недели.

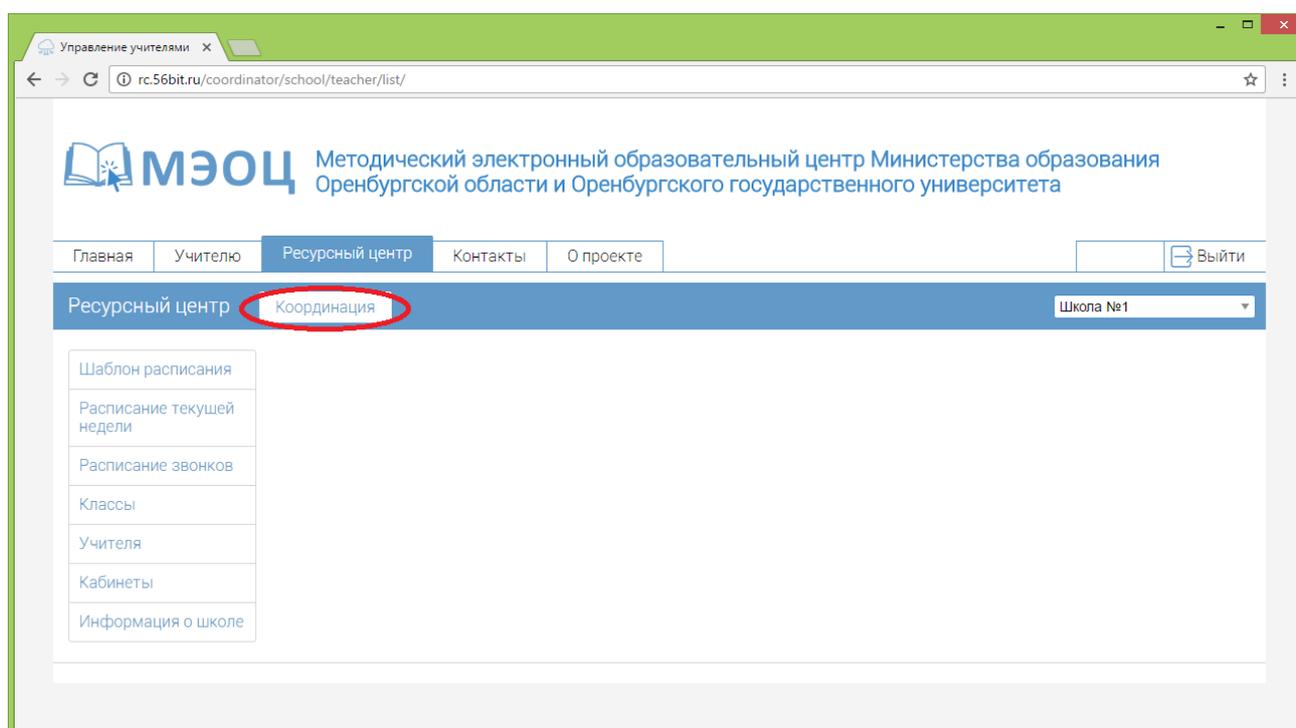


Рисунок 12.21 – Кабинет координатора РЦКД

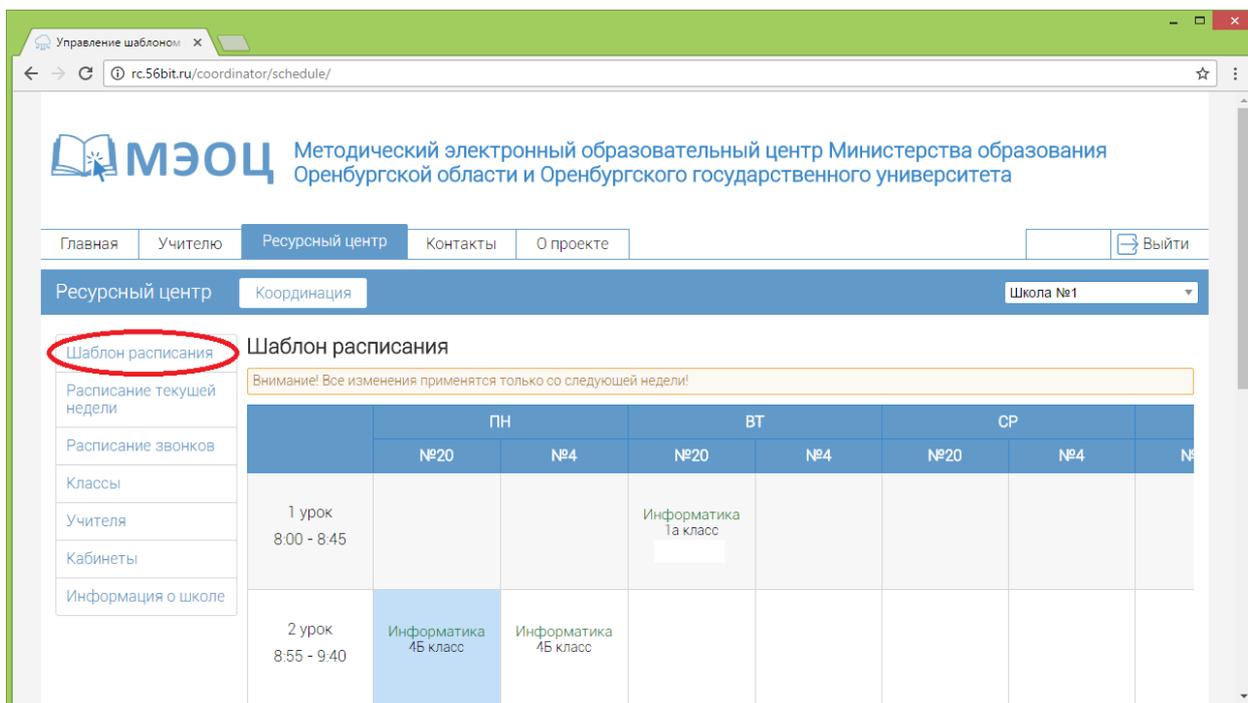


Рисунок 12.22 – Шаблон расписания

Для ввода информации о планируемом уроке в расписании выбирается свободное «окно» на пересечении номера урока и кабинета. По каждому уроку вносится следующий набор данных: предмет, класс, учитель, дисковый образ, требуемое количество компьютеров (рисунок 12.23).

Рисунок 12.23 – Создание урока в шаблоне расписания

Редактирование информации по планируемому уроку происходит аналогичным образом. Для этого в шаблоне расписания выбирается соответствующий урок, и

вносятся необходимые изменения. Если нужно отменить занятие в расписании для этого в режиме редактирования нужно нажать «Удалить урок» (рисунок 12.24).

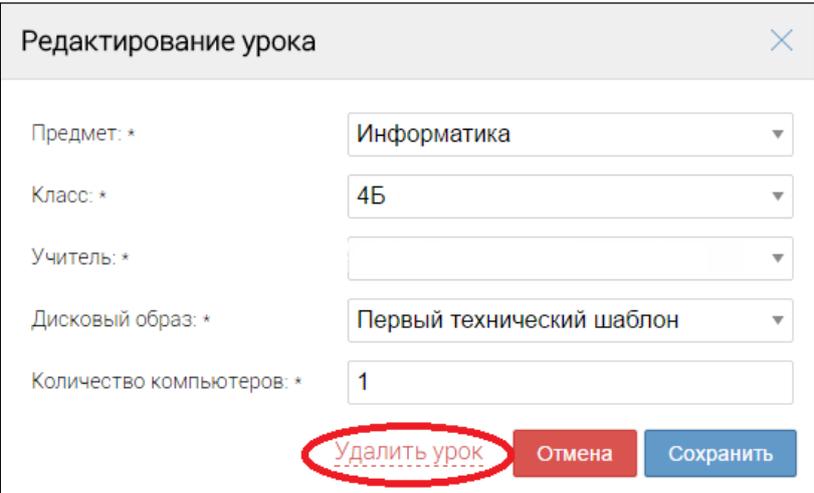


Рисунок 12.24 – Редактирование урока в шаблоне расписания

Если урок был ошибочно назначен или его нужно перенести на другую дату, то следует его удалить, а затем создать новый.

### 12.3.1.2 Управление расписанием занятий для текущей недели

Координатор может вносить корректировки в расписание текущей недели (рисунок 12.25). Обратите внимание, все изменения применяются незамедлительно и действительны только для текущей недели.

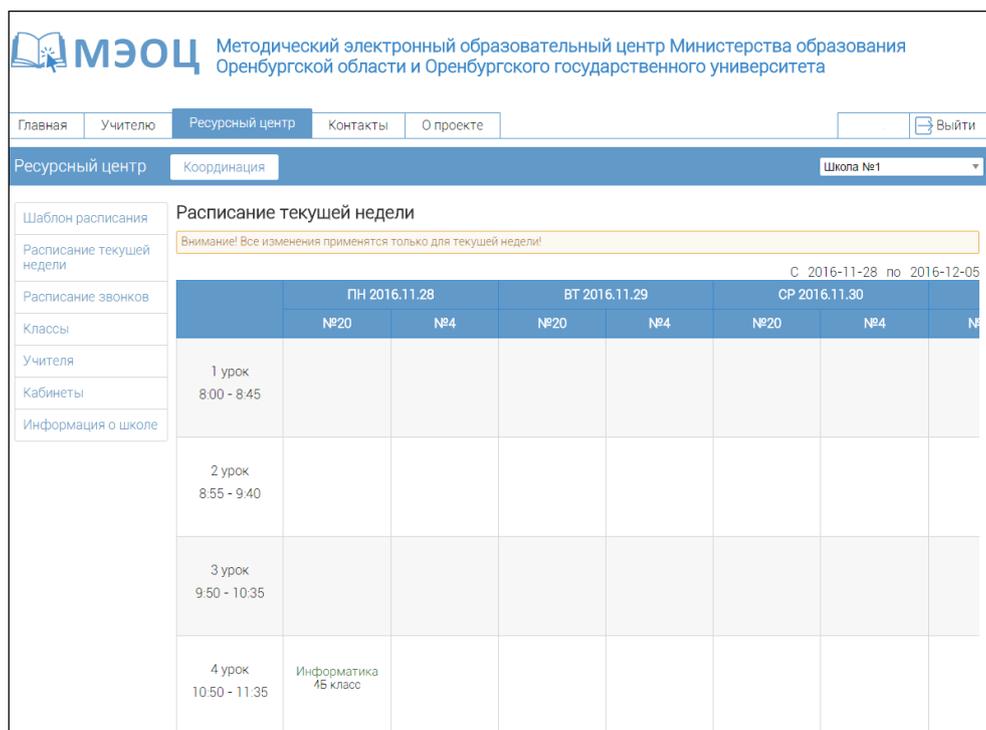


Рисунок 12.25 – Редактирование расписания текущей недели

### 12.3.1.3 Управление расписанием звонков школы

Координатор может просматривать расписание звонков планируемых уроков. Для его редактирования следует нажать на ссылку «Изменить расписание» (рисунок 12.26).

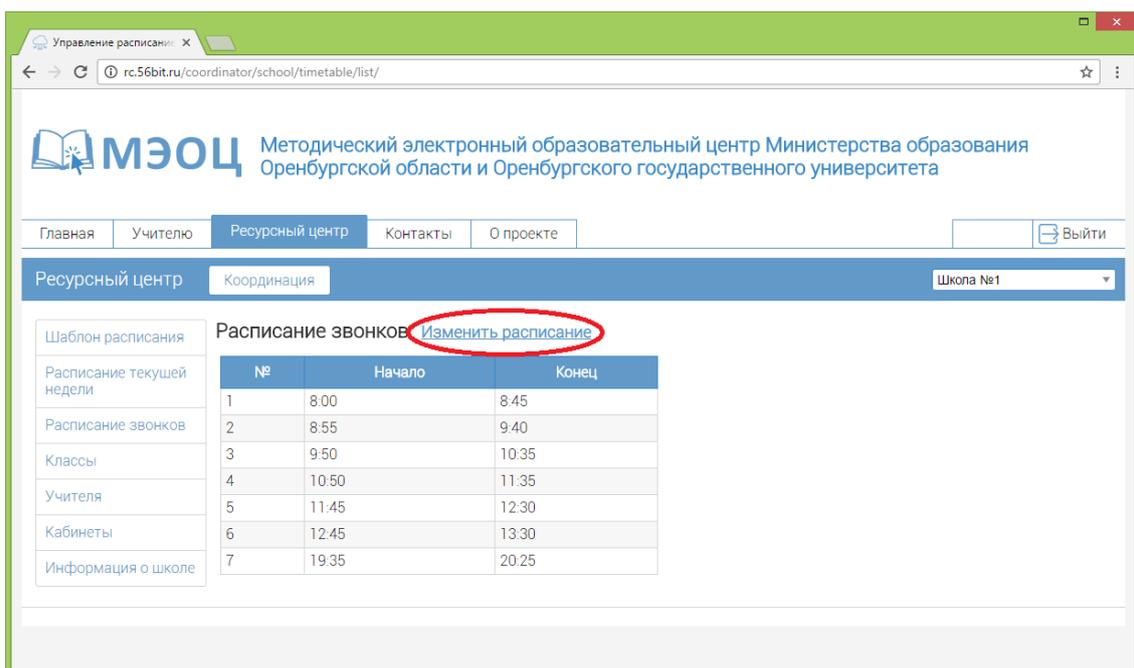


Рисунок 12.26 – Просмотр расписания звонков

На странице редактирования расписания звонков можно удалить существующий или добавить новый элемент расписания (рисунок 12.27).

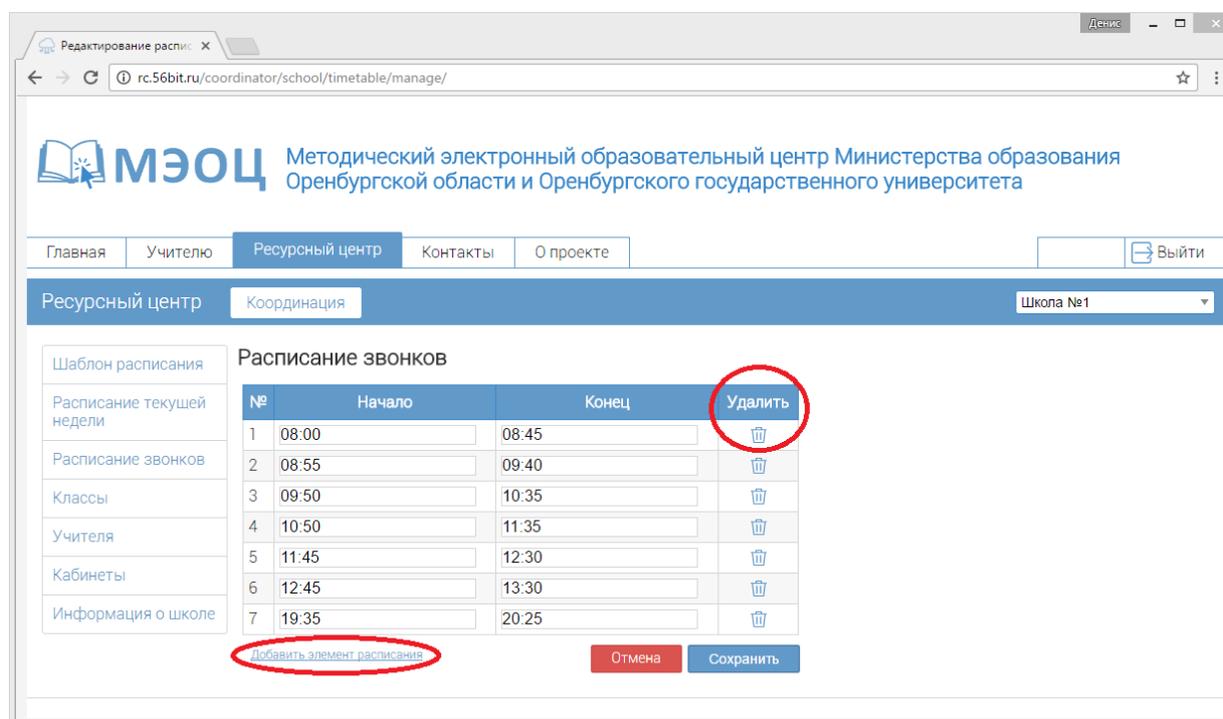


Рисунок 12.27 – Внесение изменений в расписание звонков

Внесенные изменения требуют подтверждения нажатием на кнопку «Сохранить». Для отмены несохраненных изменений и возврата в режим просмотра следует нажать кнопку «Отмена».

#### 12.3.1.4 Управление классами школы

Координатор может просматривать, редактировать и удалять информацию по всем классам в пределах школы. Для редактирования или удаления класса школы следует в соответствующей колонке выбрать пиктограмму напротив нужного класса. Для создания нового класса школы следует нажать на ссылку «Добавить класс» (рисунок 12.28).

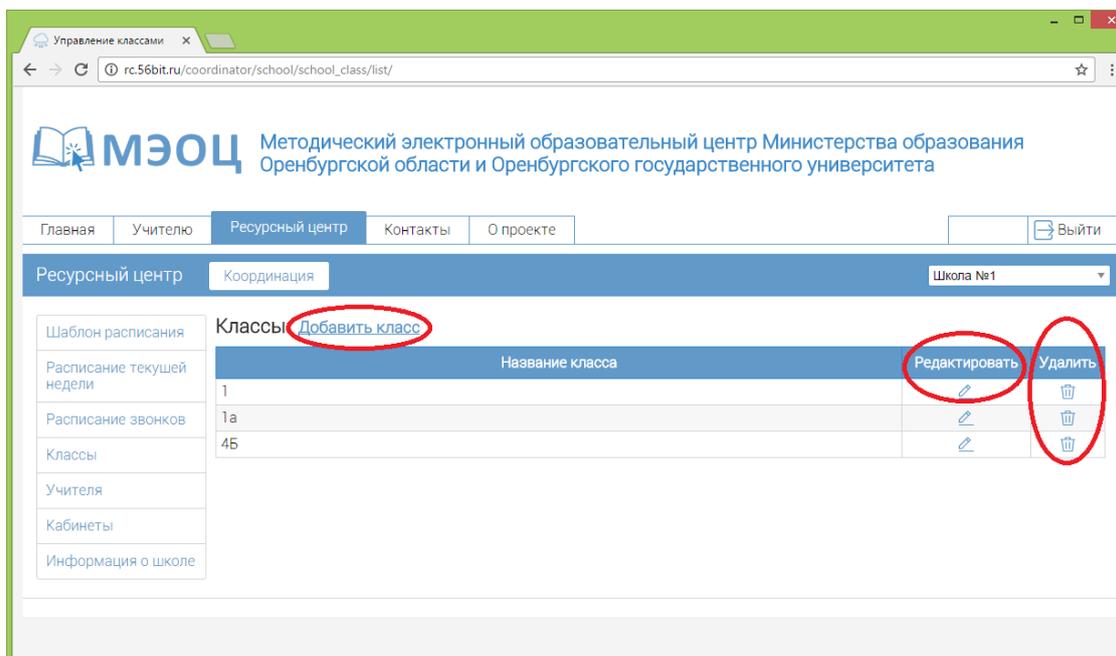


Рисунок 12.28 – Управление классами школы

В появившемся окне следует указать буквенно-цифровое обозначение нового класса. Для внесения информации следует нажать кнопку «Добавить». Если вы хотите отменить действие и вернуться в режим просмотра, следует нажать кнопку «Отмена» (рисунок 12.29). Процесс редактирования осуществляется аналогичным способом в соответствующем разделе.

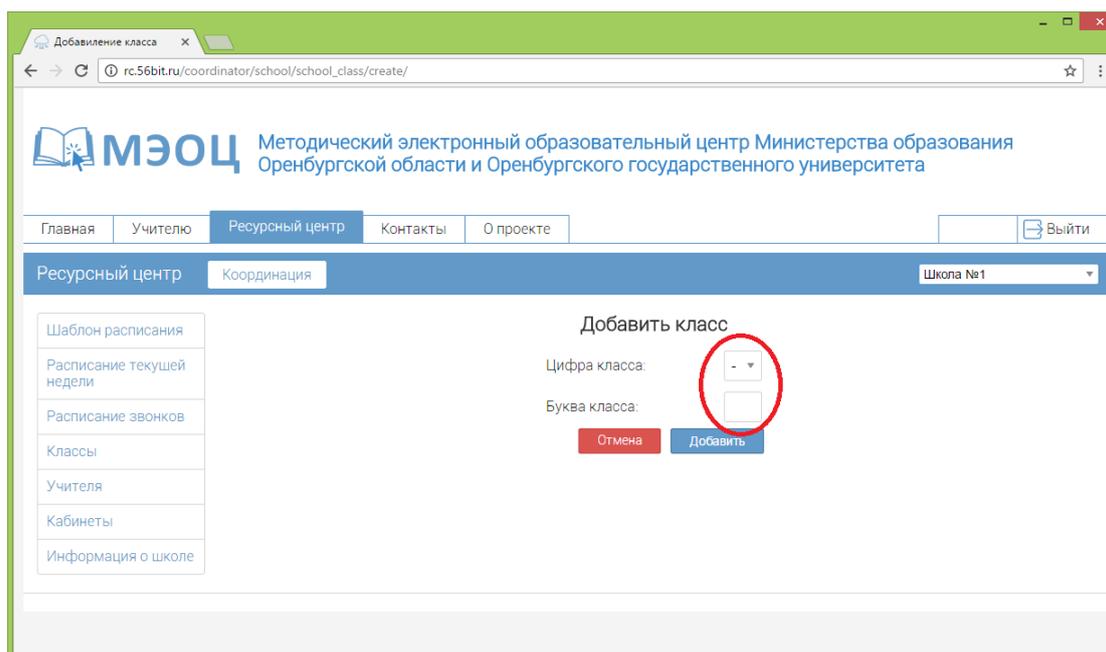


Рисунок 12.29 – Создание нового класса

### 12.3.1.5 Управление учетными записями учителей

Координатор может управлять учетными записями учителей (создавать, редактировать, удалять). Для создания новой учетной записи следует нажать на ссылку «Добавить учителя» (рисунок 12.30).

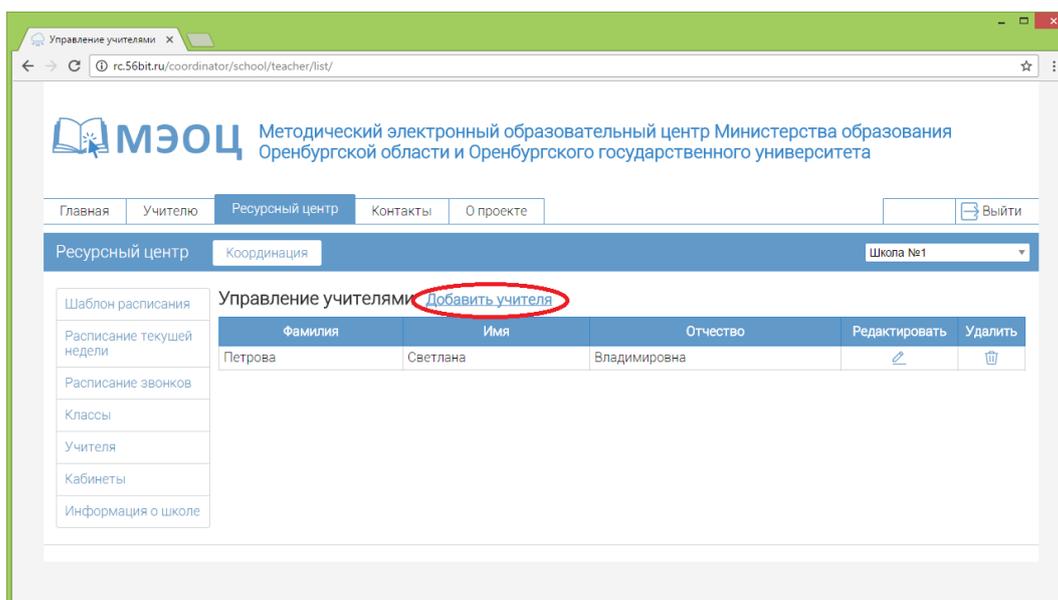


Рисунок 12.30 – Управление учетными записями учителей

В появившемся окне необходимо заполнить поля информацией об учителе (рисунок 12.31).

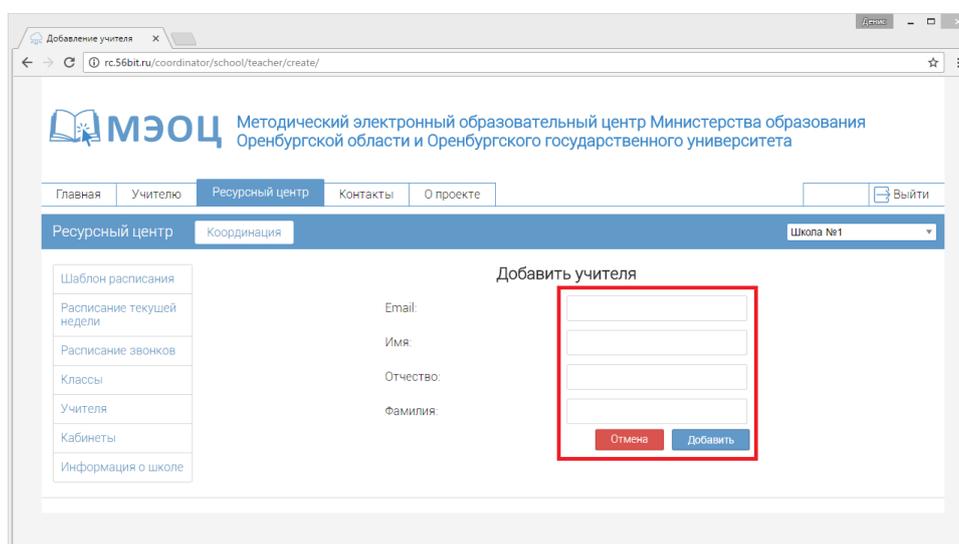


Рисунок 12.31 – Создание новой учетной записи учителя

При необходимости можно внести изменения в параметры созданной учетной записи учителя (рисунок 12.32), нажав на кнопку «Редактировать» на странице управления учетными записями (рисунок 12.30).

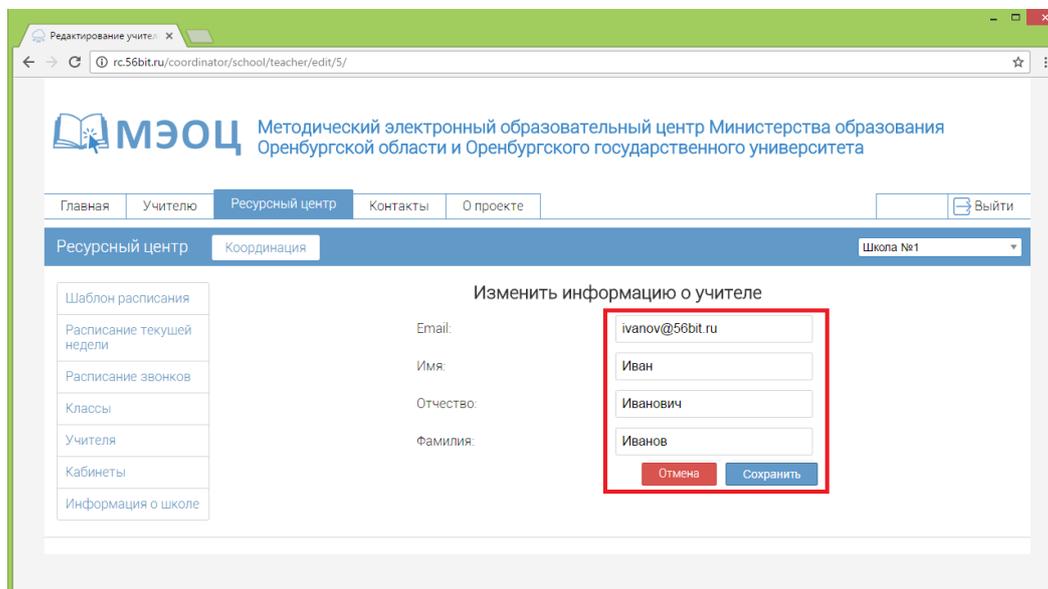


Рисунок 12.32 – Редактирование параметров созданной учетной записи учителя

### 12.3.1.6 Управление кабинетами

Координатор может управлять кабинетами (создавать, редактировать, удалять). Для создания нового кабинета следует нажать на ссылку «Добавить кабинет» (рисунок 12.33).

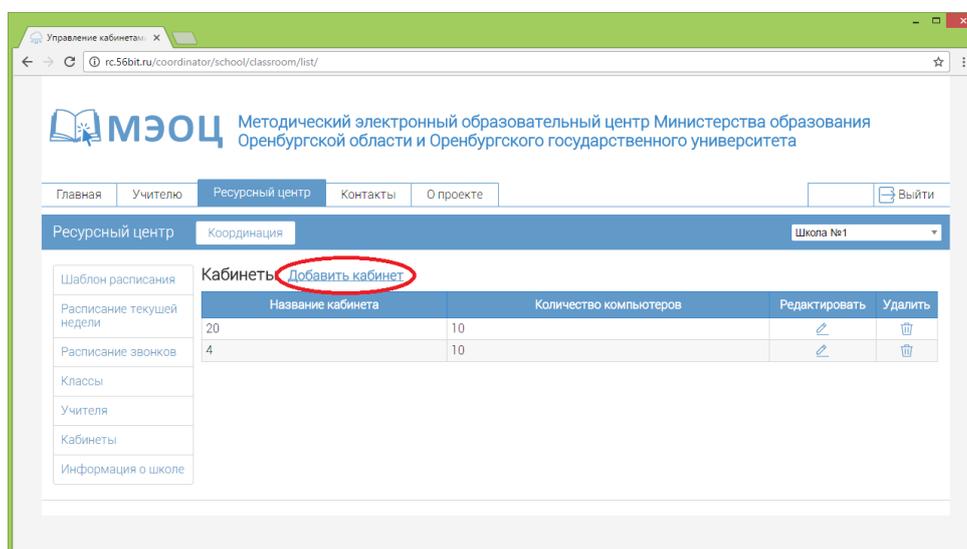


Рисунок 12.33 – Управление кабинетами

В появившемся окне следует заполнить поля сведениями о кабинете (рисунок 12.34).

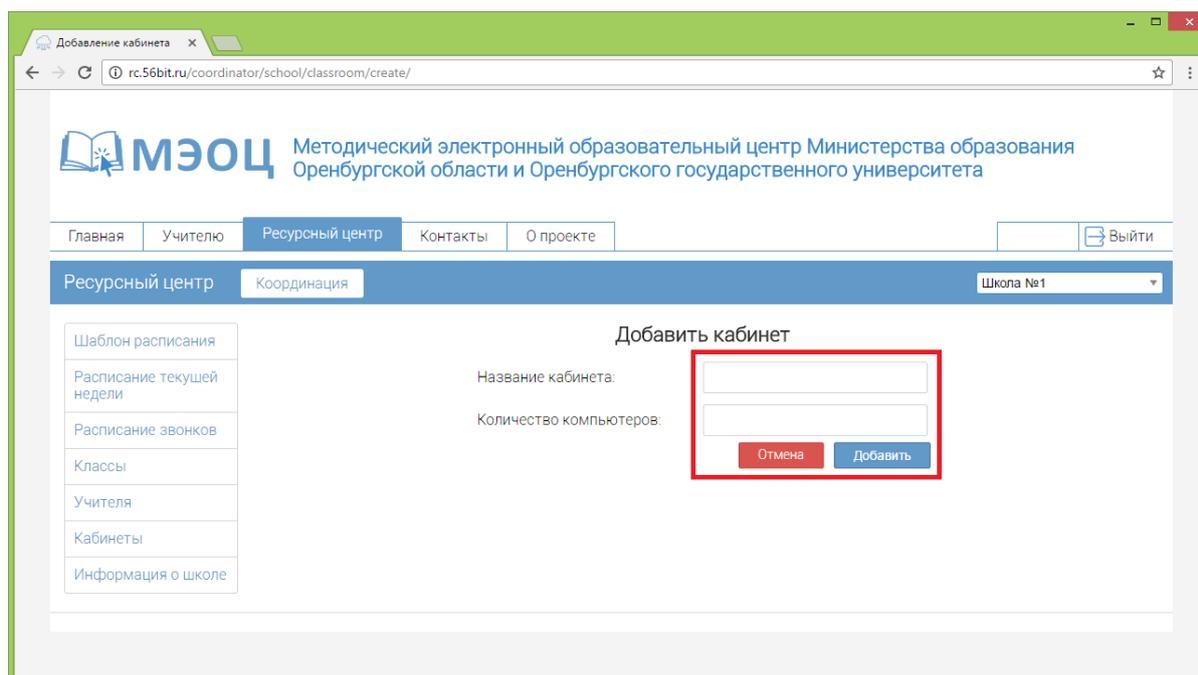


Рисунок 12.34 – Создание нового кабинета

При необходимости можно внести изменения в сведения о кабинете (рисунок 12.35), нажав на кнопку «Редактировать» на странице управления кабинетами (рисунок 12.33).

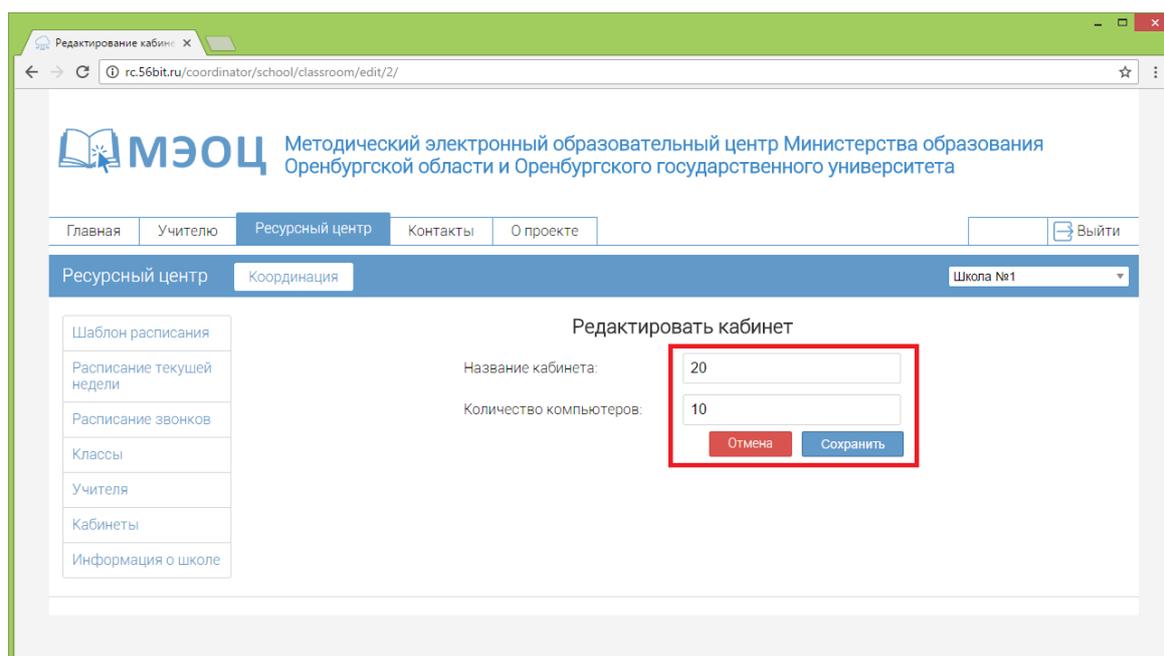


Рисунок 12.35 – Редактирование информации о кабинете

### 12.3.1.7 Редактирование информации о школе

Координатор может вносить изменения в сведения о школе (рисунок 12.36).

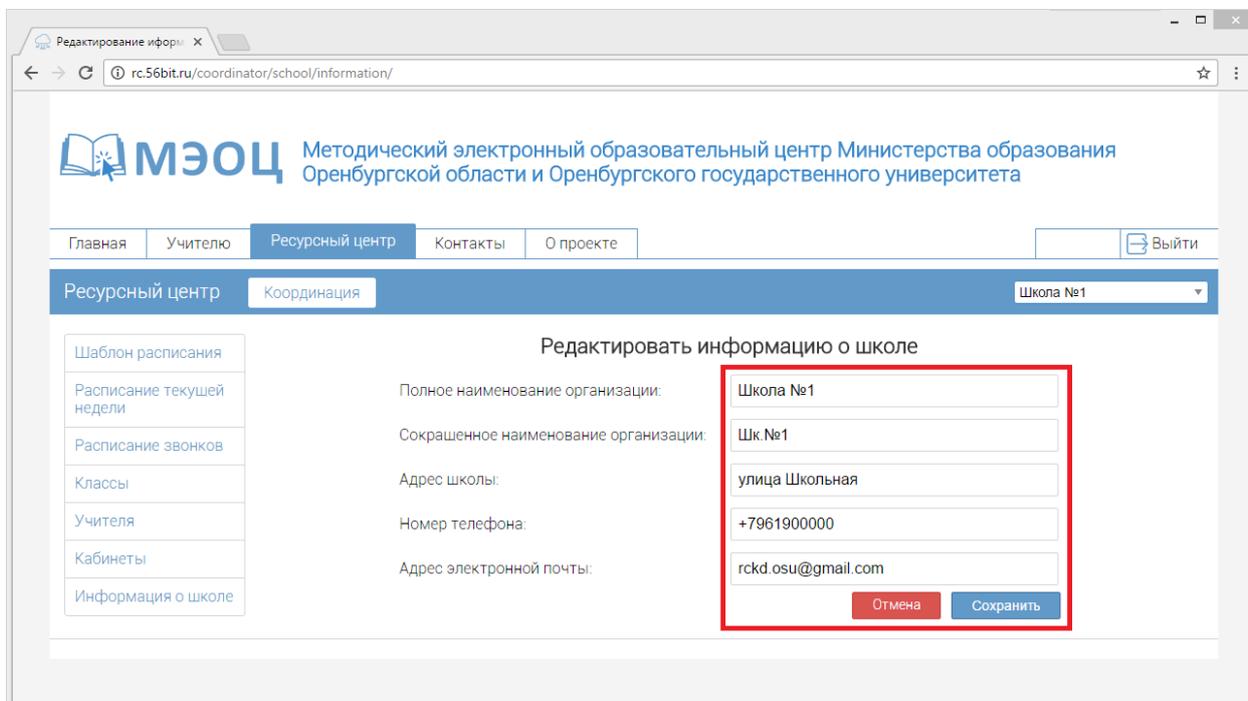


Рисунок 12.36 – Редактирование информации о школе

### 12.3.2 Учитель ресурсного центра коллективного доступа

Работа учителя осуществляется на портале РЦКД по авторизованному доступу в закрытом разделе. Права доступа назначаются координатором школы или администратором портала. Ответственным исполнителям направляется уведомление на адрес электронной почты с указанием логина и пароля. Для входа в режим учителя необходимо нажать на ссылку «Войти» (рисунок 12.37).

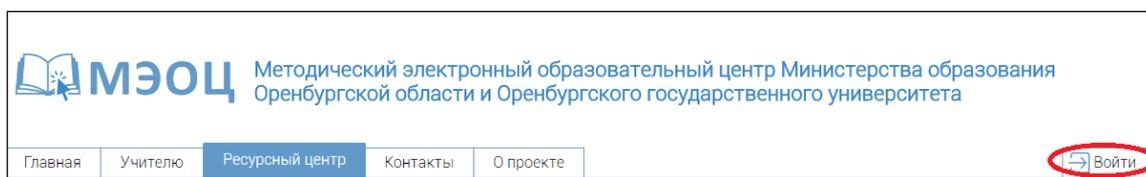


Рисунок 12.37 – Вход в режим учителя РЦКД

Для успешной авторизации на портале необходимо ввести логин и пароль, полученные на электронную почту. Если Вы забыли пароль или не можете найти письмо, то можете воспользоваться функцией восстановления пароля (рисунок 12.38).

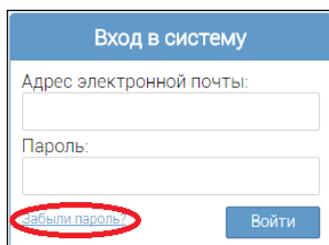


Рисунок 12.38 – Вход в систему портала МЭОЦ

Для восстановления доступа требуется указать адрес электронной почты, которая использовалась для работы на портале (рисунок 12.39).

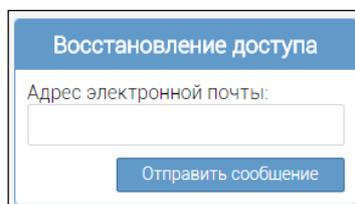


Рисунок 12.39 – Восстановление доступа к portalу

После нажатия кнопки «Отправить сообщение» на указанный адрес электронной почты придет уведомление со ссылкой, по которой необходимо будет перейти для смены пароля. Обращаем ваше внимание, ссылка действует в течение 30 минут.

Учитель РЦКД осуществляет работу согласно расписанию школы, заданному координатором, а также имеет возможность проводить внеплановые занятия по требованию при наличии ресурсов РЦКД.

### 12.3.2.1 Занятия по расписанию

Для проведения занятия по расписанию учителю необходимо открыть раздел «Мои уроки» и выбрать доступный урок (рисунок 12.40).

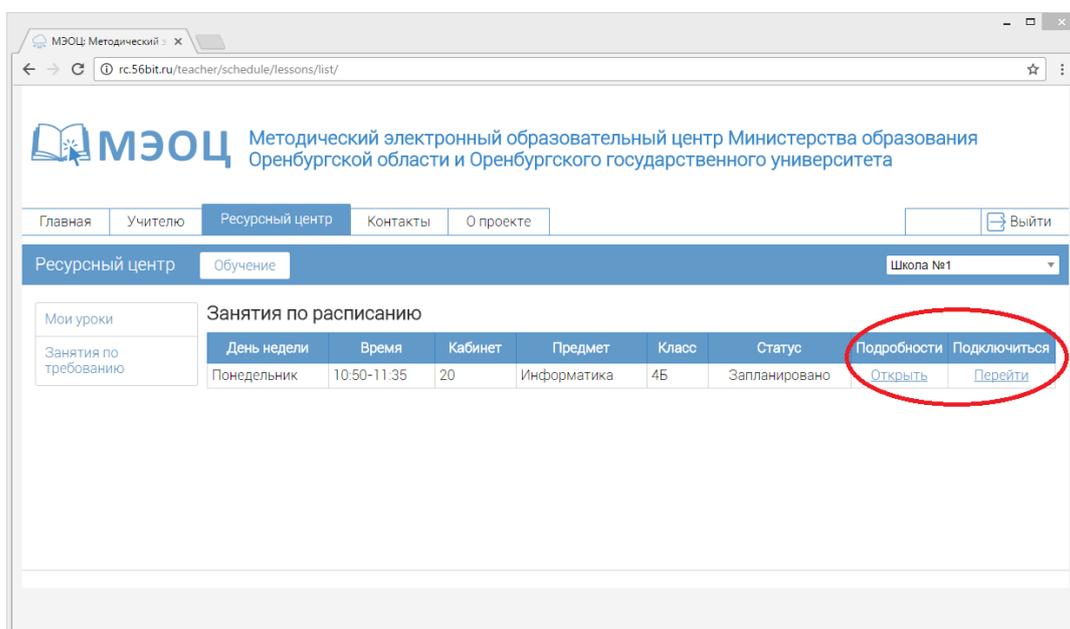


Рисунок 12.40 – Занятия по расписанию

Учитель может просматривать детальную информацию о доступном уроке (рисунок 12.41).

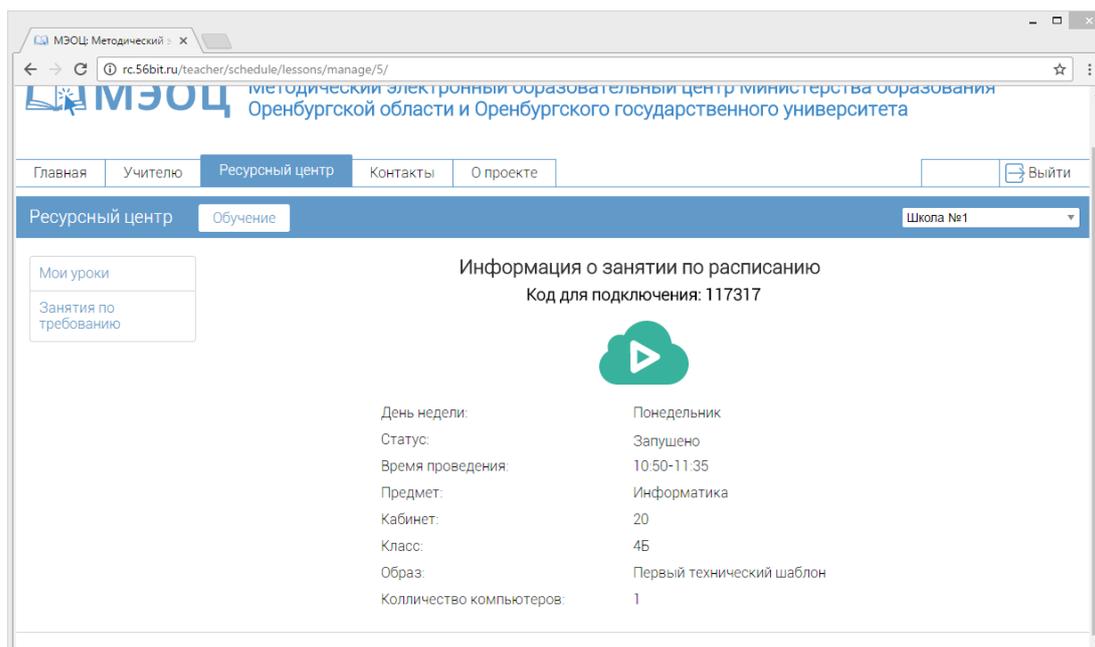


Рисунок 12.41 – Информация о занятиях по расписанию

Для работы с виртуальной машиной в РЦКД учителю необходимо нажать на ссылку «Перейти» в столбце «Подключиться» напротив выбранного урока. В новом окне откроется панель доступа к удаленному рабочему столу. Для того чтобы приступить к работе, необходимо нажать на кнопку «Войти» (рисунок 12.42).

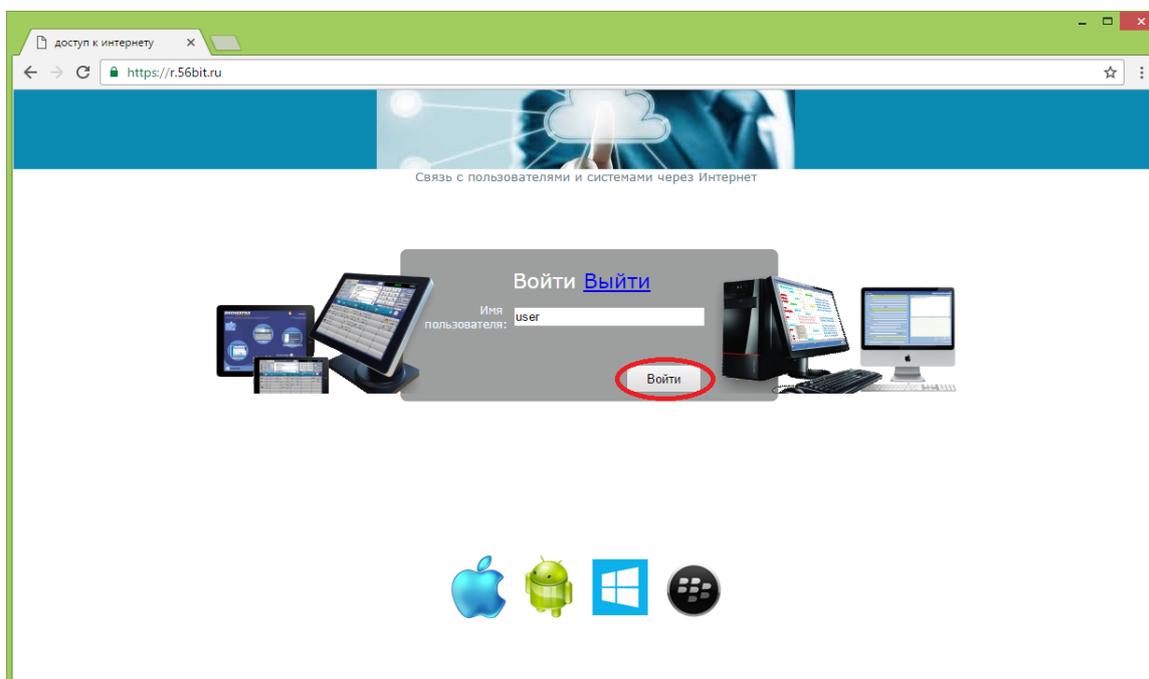


Рисунок 12.42 – Панель доступа к удаленному рабочему столу

### 12.3.2.2 Занятия по требованию

РЦКД позволяет учителю проводить занятия по требованию. Для доступа к данной функциональной возможности необходимо перейти в соответствующий раздел. Для добавления занятия следует нажать на ссылку «Создать занятие» (рисунок 12.43).

Далее следует заполнить требуемые данные, и подтвердить действие кнопкой «Создать» (рисунок 12.44), а в противном случае – «Отмена».

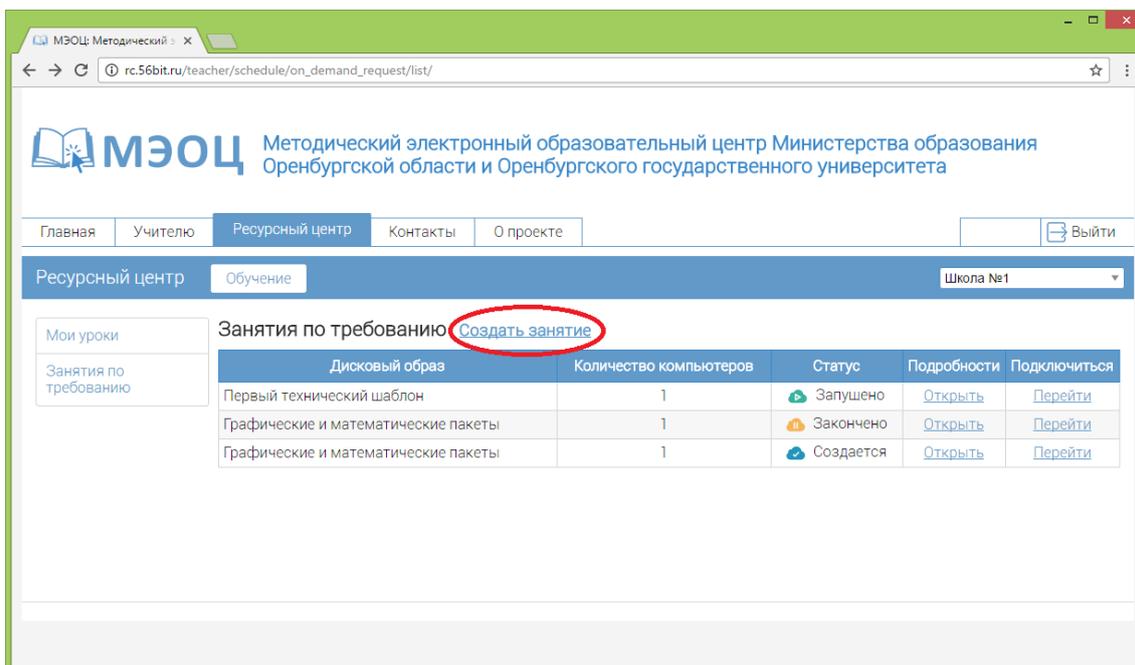


Рисунок 12.43 – Занятия по требованию

Далее следует заполнить требуемые данные, и подтвердить действие кнопкой «Создать» (рисунок 12.44), а в противном случае – «Отмена».

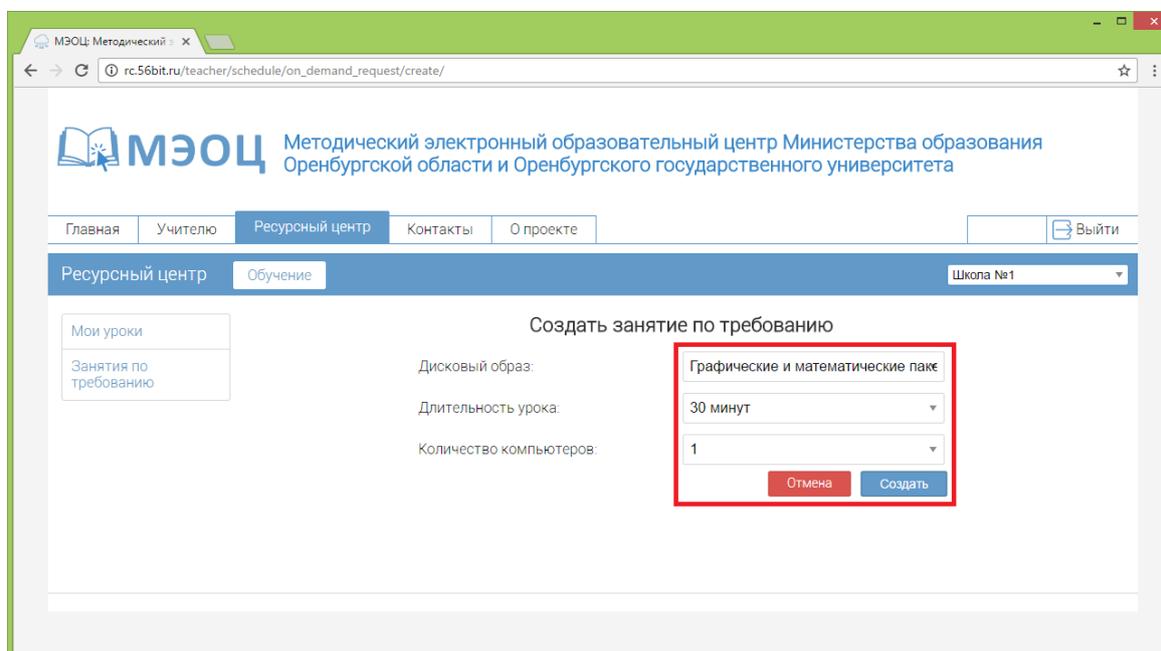


Рисунок 12.44 – Создание занятия по требованию

Виртуальные машины создаются в течение нескольких минут. Статус можно отслеживать в соответствующей колонке. Как только статус изменится на

«Запущено», можно приступать к работе. Для предоставления доступа к удаленному рабочему столу ученикам следует сообщить код для подключения. Код и более подробную информацию о занятии можно посмотреть, нажав ссылку «Открыть» в колонке «Подробности». Занятие можно отменить раньше, чем оно должно закончиться, нажав кнопку «Отменить» для запущенного занятия (рисунок 12.45).

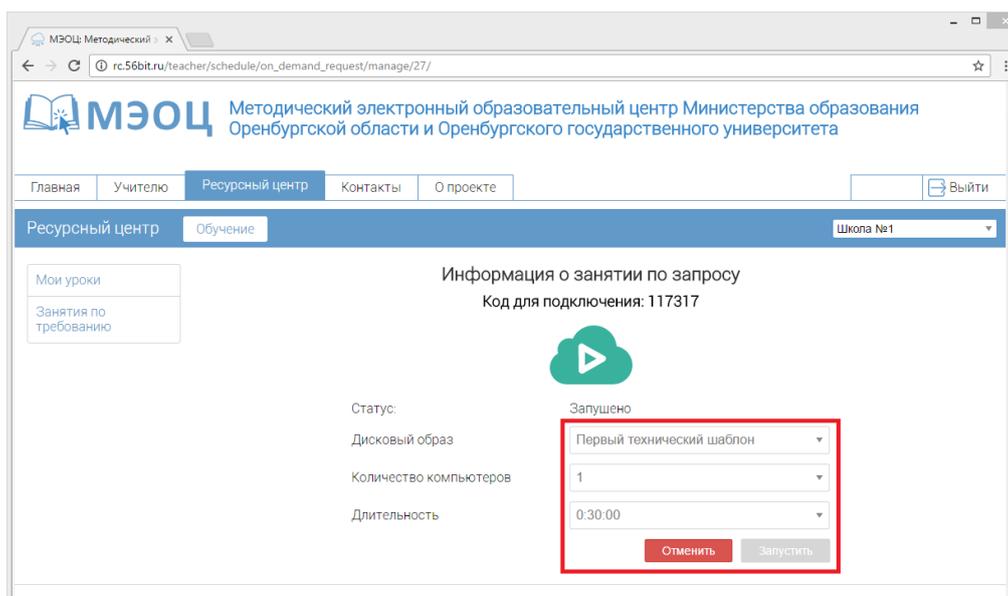


Рисунок 12.45 – Информация о занятии по требованию

Учитель может продемонстрировать процесс входа, используя данный код и выбрав вкладку «Я ученик» (рисунок 12.46).

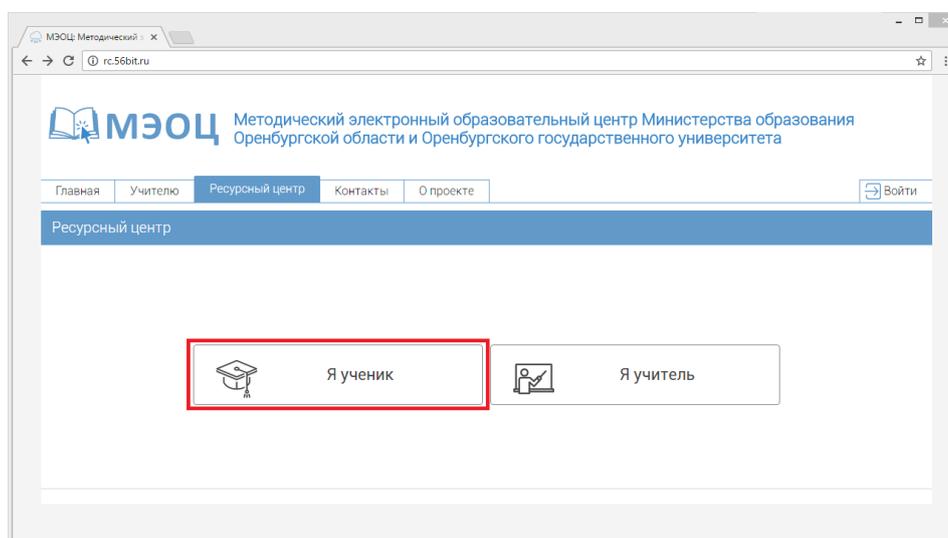


Рисунок 12.46 – Демонстрация входа для учеников

## 12.4 Ученик ресурсного центра коллективного доступа

Ученик по просьбе учителя заходит на портал МЭОЦ и переходит в раздел «Ресурсный центр». Далее для перехода к занятию следует выбрать вкладку «Я ученик».

Для доступа к уроку необходимо ввести код в специальное поле (рисунок 12.47). Код предоставляет учитель. Для работы с виртуальной машиной в РЦКД ученику необходимо нажать на кнопку «Войти».

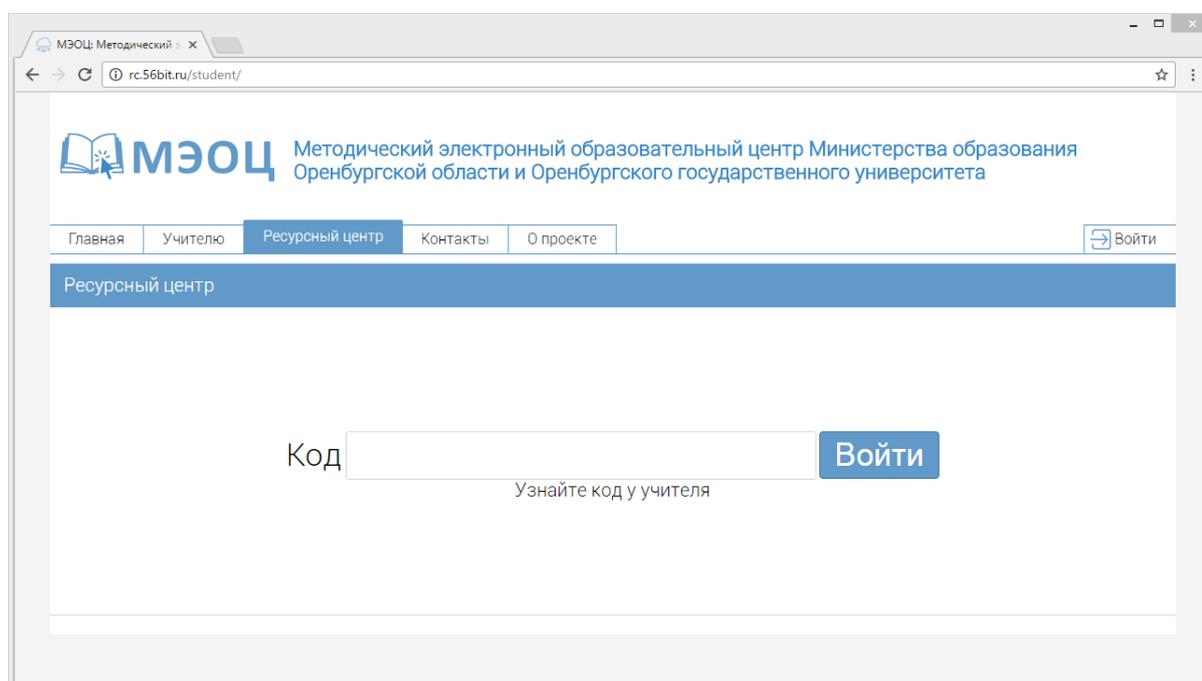


Рисунок 12.47 – Ввод кода доступа

В новом окне откроется панель доступа к удаленному рабочему столу. Для того чтобы приступить к работе, необходимо нажать на кнопку «Войти» (рисунок 12.48).

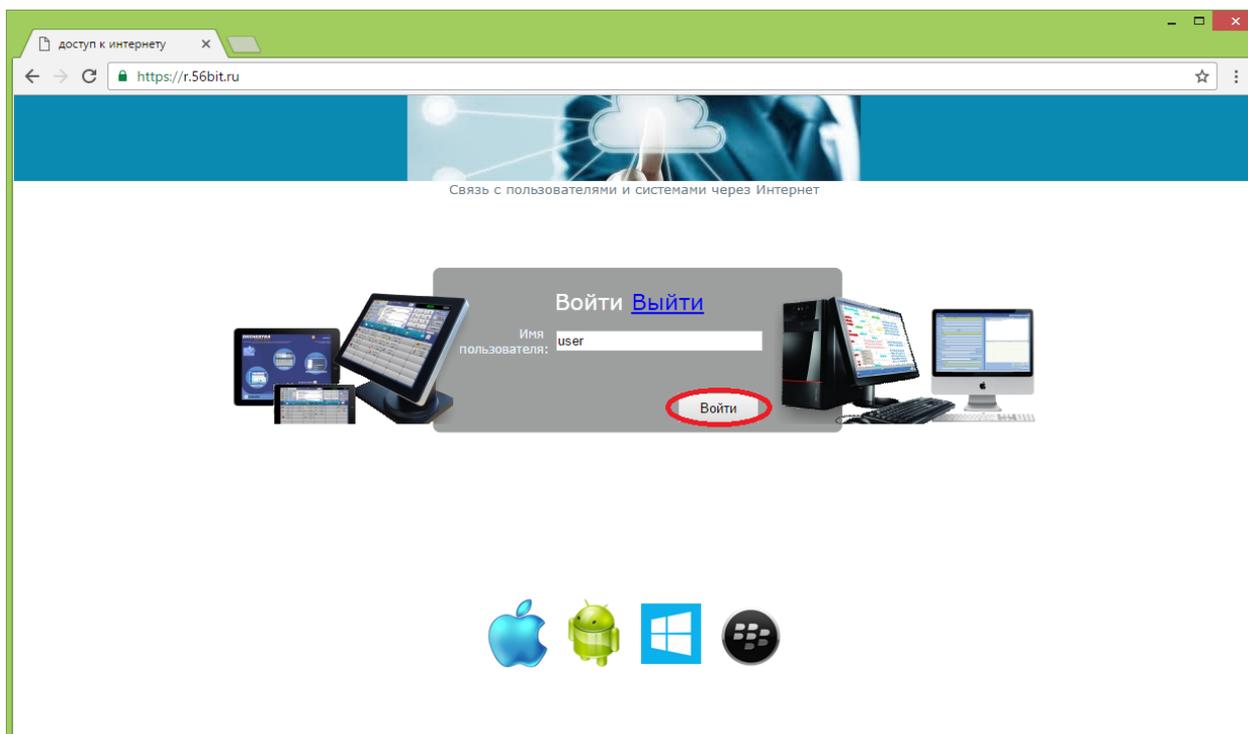


Рисунок 12.48 – Панель доступа к удаленному рабочему столу

Ученику предоставлен доступ к удаленному рабочему столу (рисунок 12.49).

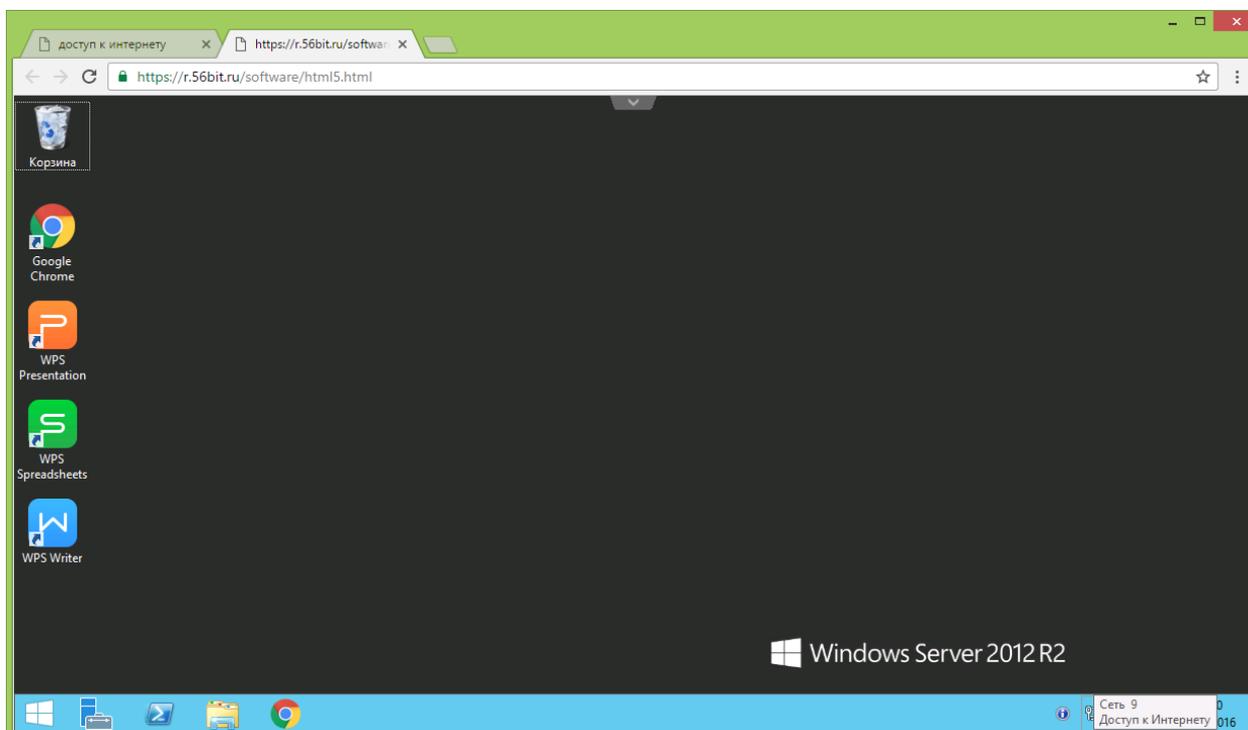


Рисунок 12.49 – Удаленный рабочий стол РЦКД

## Заключение

На первом этапе выполнения исследования проведен мониторинг потребностей образовательных учреждений в образовательных программных продуктах и возможностях использования сервисов отечественных провайдеров облачных услуг для их удовлетворения. В анкетировании приняло участие 722 общеобразовательных организаций из всех районов и городов Оренбургской области. Выявлена значимая разница между сельскими и городскими школами в скорости доступа в сеть Интернет. В 243 сельских школах (47 %) скорость менее 1 Мбит/с. В то же время в Оренбурге 52 % школ имеют возможность высокоскоростного доступа.

ОС Linux используется редко, зато неподдерживаемые официально операционные системы Windows 2000, XP, Vista, 7 в том или ином виде применяются в 89 % школ, а новейшая Windows 10 – только в 15 %. Основная причина использования устаревших версий Windows – устаревшие конфигурации компьютеров. 383 школы используют компьютеры с объемом ОЗУ 512 Мб и менее. Только в 41 школе минимальная конфигурация компьютеров имеет 3 Гб ОЗУ и более.

В большинстве школ используются устаревшие версии Microsoft Office, что связано с использованием устаревших версий Windows. Однако также параллельно в 32 % школ применяются бесплатные офисные пакеты OpenOffice и LibreOffice. В основном в школах используются бесплатные графические пакеты. Однако количество установок Corel Draw и Adobe Photoshop или Illustrator позволяет оценить суммарные расходы на обновление используемых графических пакетов в Оренбургской области до 15 млн. рублей.

Анализ публикаций показал, что в настоящее время отсутствуют разработки централизованных систем управления облачными ресурсами, которые учитывали бы специфику работы образовательных организаций.

Разработана система критериев оценки и проведен сравнительный анализ различных технологий реализации виртуальных классов Linux и Windows на базе виртуальных машин, серверов терминалов и контейнеров. Для реализации выбраны

следующие технологии: реализация виртуализации рабочего места через Microsoft Server User Roaming Login, реализация сервера терминалов через Tsplus.

Проведена формализация задачи оптимального планирования уроков виртуальных классов. Для решения задачи разработаны интеллектуальные алгоритмы на основе имитации отжига и генетических алгоритмов. Проведен экспериментальный анализ алгоритмов на имитационных моделях. Для реализации в системе выбран алгоритм имитации отжига, продемонстрировавший высокую эффективность. Также разработан алгоритм отслеживания лицензий на платные программные продукты, экспериментально подтверждена его эффективность.

Разработана методика оптимизации ресурсного обеспечения РЦКД, определены оптимальные конфигурации виртуальных машин для поддержки виртуальных классов с количеством компьютеров от 4 до 20, обоснована оптимальность реализации виртуального класса в рамках одного виртуального сервера с терминальным доступом к рабочим столам.

Выполнено проектирование системы управления региональным центром коллективного доступа к образовательным программным продуктам на базе облачных услуг отечественных провайдеров: разработана архитектура системы, иллюстрирующая ее основные компоненты, также построена диаграмма вариантов использования UML, отражающая основных пользователей системы и их функциональные возможности. Сравнительный анализ каркасов Web-приложений определил в качестве основной платформы для реализации модулей Python с каркасом Django.

Разработана технология защиты информации в РЦКД, включающая защиту инфраструктуры от внешних воздействий и защиту пользователей от нежелательного контента. Технология подразумевает установку и настройку межсетевого экрана и систем мониторинга на все узлы, установку антивируса на Windows системы, установку прокси-сервера с фильтрами для доступа в Интернет с узлов, настройку DNS на безопасный фильтрованный DNS Yandex, применение политик ограниченного использования программ, периодическое восстановление всех Windows узлов из образов, установление защищенного соединения между

пользователем и компонентами РЦКД, использование разовых кодов для предоставления доступа к виртуальным окружениям.

Реализованы основные инфраструктурные компоненты системы: модуль динамической настройки nginx, модуль методического электронного образовательного центра, модуль подсистемы доступа к виртуальным серверам, модуль запуска и отслеживания виртуальных серверов, модуль обертки над API публичного облачного провайдера, база данных МЭОЦ и база данных РЦКД.

Создана подсистема совместного доступа к виртуальным урокам, включающим учебные текстовые и видеоматериалы, которая получила название «Методический электронный образовательный центр». Для каждого школьного предмета методические материалы структурированы по годам обучения, разделам и темам. В системе поддерживается размещение текстовых материалов, презентаций, видеороликов и архивов с прочими файлами.

Разработаны типовые конфигурации виртуальных классов для изучения базового и углубленного курсов информатики в 10-11 классе на основе УМК И.Г. Семакина и К.П. Полякова. Определены характеристики и набор программного обеспечения для офисной конфигурации и мультимедиа конфигурации виртуальных серверов, которые успешно использовались на этапе тестовой эксплуатации РЦКД.

Для тестовой эксплуатации МЭОЦ проведено размещение методических материалов для 9-11 классов по 6 школьным предметам: информатике, обществознанию, истории, геометрии, алгебре, физике. На портале размещено 1103 урока, которые включают 1301 текстовый документ, 513 презентаций, 100 заданий, 24 видеоролика, 49 других материалов.

Для тестовой эксплуатации РЦКД в систему были добавлены координаторы 9 школ г. Оренбурга и Оренбургской области. Всего в процессе тестирования РЦКД было создано 29 занятий по требованию, 62 урока в постоянном расписании.

В целом, тестовая эксплуатация системы показала ее соответствие выдвинутым функциональным и нефункциональным требованиям, что позволило считать разработку завершенной и перейти к эксплуатации системы с 1.12.2016 г.

Проведены экспериментальные исследования эффективности, сбалансированности и производительности сервисов РЦКД, которые позволяют сделать выводы о работоспособности системы, а также высокой эффективности системы кеширования и балансировки нагрузки.

Разработаны подробные рекомендации по использованию ресурсов РЦКД для образовательных организаций.

Результаты исследования представлены на следующих научных конференциях:

а) 8th IEEE International Conference on Intelligent Systems IS'16 (Болгария, г. София);

б) 16th International Conference on Next Generation Wired/Wireless Advanced Networks and Systems NEW2AN 2016 (г. Санкт-Петербург);

в) 10th IEEE International Conference on Application of Information and Communication Technologies (Азербайджан, г. Баку);

г) международная конференция «Информационные технологии в промышленности и производстве» (г. Томск);

д) XII Международный симпозиум «Интеллектуальные системы» INTELS'2016 (г. Москва);

е) VIII Международная научно-практическая конференция «Информационные технологии в образовании» (г. Саратов);

ж) XVII Международная научно-техническая конференция «Проблемы техники и технологий телекоммуникаций» (г. Самара);

з) международная научно-практическая конференция «Современные информационные технологии и ИТ-образование» (г. Москва);

и) XIII Международная научно-техническая конференция «Новые информационные технологии и системы» НИТиС – 2016 (г. Пенза).

Публикации по теме исследования, индексируемые в Scopus и Web of Science:

1. Konnov A., Legashev L., Polezhaev P., Shukhman A. Concept of Cloud Educational Resource Datacenters for Remote Access to Software // Proceedings of 11th International Conference on Remote Engineering and Virtual Instrumentation (REV), Polytechnic of Porto (ISEP) in Porto, Portugal from 26-28 February 2014 – P. 246-247.

2. Bolodurina I., Legashev L., Polezhaev P., Shukhman A., Ushakov Yu. Request generation and intelligent scheduling for cloud educational resource datacenter // Proceedings of 8th Int. Conf. on Intelligent Systems – 2016. – P. 747-752.

3. Shukhman A., Bolodurina I., Polezhaev P., Ushakov Yu., Legashev L. Creation of regional center for shared access to educational software based on cloud technology // Selected Papers of the XI International Scientific-Practical Conference Modern Information Technologies and IT-Education SITITO-2016 – 2016. – P. 180-188.

4. Shukhman A., Bolodurina I., Polezhaev P., Legashev L. Cloud educational resource datacenter simulator // Procedia Computer Science – 2017. – V. 103 – P. 543-548.

Публикации по теме исследования, индексируемые в РИНЦ:

5. Полежаев П.Н., Адрова Л.С. Концепция регионального центра коллективного доступа к образовательным программным продуктам // Информация и образование: границы коммуникаций INFO'16: сборник научных трудов № 8 (16); под ред. А. А. Темербековой, Л. А. Альковой. – Горно-Алтайск : РИО ГАГУ, 2016. – С. 73-74.

6. Шухман А.Е., Болодурина И.П., Полежаев П.Н., Ушаков Ю.А., Легашев Л.В. Создание регионального центра коллективного доступа к образовательным программным продуктам на базе облачных технологий // Современные информационные технологии и ИТ-образование. Т. 12 (№ 3), часть 2, 2016. – С. 83-90.

7. Болодурина И.П., Шухман А.Е., Полежаев П.Н., Легашев Л.В. Симулятор регионального центра коллективного доступа к образовательным программным продуктам // «Проблемы техники и технологий телекоммуникаций» ПТиТТ-2016. Материалы XVII Международной научно-технической конференции (г. Самара, 22-24 ноября 2016 г.). – Казань: ООО «16ПРИНТ», 2016. – С. 84-85.

8. Полежаев П.Н., Легашев Л.В., Шухман А.Е., Ушаков Ю.А. Алгоритмы планирования вычислений и поддержки принятия решений для облачных ресурсных центров // «Проблемы техники и технологий телекоммуникаций» ПТиТТ-2016. Материалы XVII Международной научно-технической конференции (г. Самара, 22-24 ноября 2016 г.). – Казань: ООО «16ПРИНТ», 2016. – С. 115-116.

9. Болодурина И.П., Легашев Л.В., Анциферова Л.М. Интеллектуальные алгоритмы управления облачными образовательными ресурсами // Сборник статей XXXII Междунар. науч.-техн. конф. (г. Пенза, 6-8 июня 2017 г.) - Пенза: Изд-во ПГУ – 2017. – Т. 1. – С. 97-100.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Костенецкий, П.С. Создание образовательной платформы "Персональный виртуальный компьютер" на базе облачных вычислений / П.С. Костенецкий, А.И. Семёнов, Л.Б. Соколинский // Научный сервис в сети Интернет: экзафлопсное будущее: Труды международной научной конференции (19-24 сентября 2011 г., г. Новороссийск). М.: Изд-во МГУ – 2011. – С. 374-377.

2 Bo Wang, HongYu Xing The Application of Cloud Computing in Education Informatization // IEEE International Conference on Cloud Computing Technology and Science – 2011. – P. 2673-2676.

3 Shinichiro Kibe, Teruaki Koyama, Minoru Uehara The Evaluations of Desktop as a Service in an Educational Cloud // 15th International Conference on Network-Based Information Systems – 2012. – P. 621-626.

4 Justin Cappos, Ivan Beschastnikh, Arvind Krishnamurthy, Tom Anderson Seattle: A Platform for Educational Cloud Computing // SIGCSE'09, March 3-7. – 2009. – P. 111-115.

5 Kibe S., Uehara M., Yamagiwa, M. Evaluation of Bottlenecks in an Educational Cloud Environment // 2011 Third International Conference on Intelligent Networking and Collaborative Systems. – 2011. – P. 520-525.

6 Ercan, T. Effective use of cloud computing in educational institutions // Procedia-Social and Behavioral Sciences. – 2010. – Т. 2. – №. 2. – P. 938-942.

7 Mathew, S. Implementation of Cloud Computing in Education-A Revolution // International Journal of Computer Theory and Engineering. – 2012. – Т. 1. – №. 5. – P. 6.

8 Фаулер, М. Архитектура корпоративных программных приложений / М. Фаулер, Пер с англ. – Москва: Вильяме – 2006. – 544 с.

9 Магдануров, Г.И. ASP. NET MVC Framework / Г.И. Магдануров // БХВ-Петербург, 2010.

10 Vaswani, V. Zend Framework, A Beginner's Guide / V. Vaswani // McGraw Hill Professional, 2010.

11 Bowler, T. Symfony 1.3 web application development / Bowler T., Bancer W // Packt Publishing Ltd, 2009.

12 Hartl, M. Ruby on rails tutorial: learn Web development with rails / M. Hartl // Addison-Wesley Professional, 2015.

13 Saxena, A. Struts based Approach for the Development of Java Applications / A. Saxena //Journal of Software Engineering Tools & Technology Trends. – 2014. – Т. 1. – №. 2. – С. 20-27.

14 Дронов, В.А. Django: практика создания Web-сайтов на Python / В.А. Дронов // БХВ-Петербург, 2016.– 528 с.

15 Iptables – Викиучебник: Wikimedia Project // Электрон. дан. – 2016. Режим доступа: <https://ru.wikibooks.org/wiki/Iptables> – 27.10.2016.

16 Настройка Fail2ban для защиты SSH // PuTTY: Telnet/SSH Клиент. – Электрон. дан. – 2016. – Режим доступа: <http://putty.org.ru/articles/fail2ban-ssh.html> – 27.10.2016.

17 Возможности и преимущества Microsoft Security Essentials // WindowsFAQ. – Электрон. дан. – 2016. – Режим доступа: <http://www.securrity.ru/articles/1238-vozmozhnosti-i-preimuschestva-microsoft-security-essentials.html> – 27.10.2016.

18 Nagios - The Industry Standard In IT Infrastructure Monitoring // Nagios Enterprises, LLC. – Электрон. дан. – 2016. – Режим доступа: <https://www.nagios.org/> – 27.10.2016.

19 squid : Optimising Web Delivery // Squid Software. – Электрон. дан. – 2016. – Режим доступа: <http://www.squid-cache.org/> – 27.10.2016.

20 Яндекс.DNS // ООО «Яндекс». – Электрон. дан. – 2016. – Режим доступа: <https://dns.yandex.ru/> – 27.10.2016.

21 Ограничение списка запускаемых программ при помощи групповых политик / Владислав Семёнов. – Электрон. дан. – 2016. – Режим доступа: <http://www.windowsfaq.ru/content/view/694/46/> – 27.10.2016.

22 Веденюхин, А. Ключи, шифры, сообщения: как работает TLS / А. Веденюхин – Электрон. дан. – 2016. – Режим доступа: <https://tls.dxdn.ru/tls.html> – 27.10.2016.

23 Зачем нужен Docker и практика работы с ним // Блог о программировании, операционных системах, СУБД, девайсах, сетях, алгоритмах и пр. – Электрон. дан. – 2016. – Режим доступа: <http://eax.me/docker/> – 27.10.2016.

24 Гражданский кодекс Российской Федерации часть 4 (ГК РФ ч.4) // База документов Консультант Плюс – Электрон. дан. – 2016. – Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_64629/](http://www.consultant.ru/document/cons_doc_LAW_64629/) – 7.11.2016.

25 Семакин, И.Г. УМК по информатике / И.Г. Семакин, Е.К. Хеннер, Т.Ю. Шеина // Лаборатория базовых знаний – Электрон. дан. – 2016. – Режим доступа: <http://metodist.lbz.ru/authors/informatika/2/> – 7.11.2016.

26 Еремин, Е.А. УМК по информатике / Е.А. Еремин, К.Ю. Поляков // Лаборатория базовых знаний – Электрон. дан. – 2016. – Режим доступа: <http://metodist.lbz.ru/authors/informatika/7/> – 7.11.2016.

## МОНОГРАФИЯ

Ирина Павловна Болодурина  
Андрей Леонидович Коннов  
Петр Николаевич Полежаев  
Юрий Александрович Ушаков  
Александр Евгеньевич Шухман  
Леонид Вячеславович Легашев  
Денис Игоревич Парфенов

# ИСПОЛЬЗОВАНИЕ ОБЛАЧНЫХ ТЕХНОЛОГИЙ ПРИ СОЗДАНИИ РЕГИОНАЛЬНОГО ЦЕНТРА КОЛЛЕКТИВНОГО ДОСТУПА К ОБРАЗОВАТЕЛЬНЫМ ПРОДУКТАМ

Подписано в печать 16.01.2018 г. Формат 60x84<sup>1</sup>/<sub>16</sub>,  
бумага офсетная, гарнитура «Таймс».  
Усл. печ. листов 10,0. Тираж 100 экз. Заказ 37.

---

Издательско-полиграфический комплекс ОГУ  
460018, г. Оренбург, пр-т Победы 13,  
Оренбургский государственный университет

---

ISBN 978-5-7410-1904-7

