

Разработка симулятора облачного ресурсного центра*

П.Н. Полежаев, А.Л. Коннов, А.Е. Шухман,
Оренбургский государственный университет

Данная статья посвящена проблеме создания образовательных ресурсных центров, как средства организации экономически выгодного удаленного доступа образовательных учреждений к платному программному обеспечению. Приводится имитационная схема симулятора образовательного ресурсного центра, который может быть использован для исследования алгоритмов планирования виртуальных классов и виртуальных машин. Разработан набор показателей эффективности образовательного ресурсного центра для их сравнения.

1. Введение

В настоящее время в развивающихся странах большинство образовательных учреждений функционирует в условиях недостаточного финансирования на закупку ПО, необходимого в учебном процессе. Это приводит к снижению качества образования, а также к росту уровня незаконного использования ПО.

В рамках данной работы предлагается решение этой проблемы путем создания образовательных ресурсных центров (ОРЦ) с возможностью удаленного доступа к ним образовательных учреждений. Каждый ОРЦ представляет собой центр обработки данных, поверх него развернута облачная система OpenStack [1] с виртуальными машинами, содержащими все необходимое платное ПО (см. рис. 1).

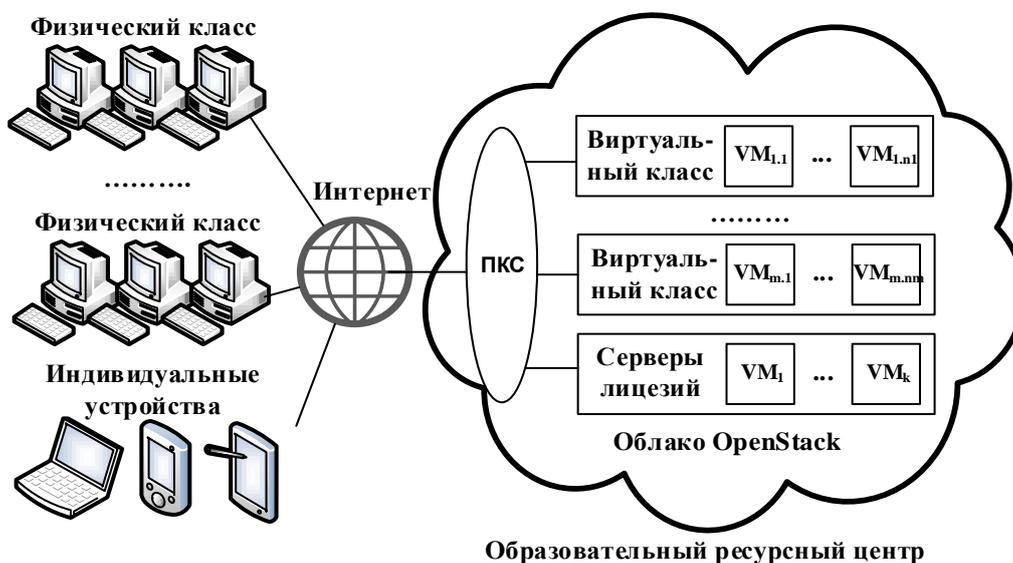


Рис. 1. Логическая структура образовательного ресурсного центра

Каждое образовательное учреждение, зарегистрированное в ОРЦ, получает доступ к одному или нескольким виртуальным компьютерным классам, представляющим собой группы экземпляров виртуальных машин со всем необходимым установленным платным и бесплатным программным обеспечением. Доступ предоставляется учащимся по сети Интернет, при этом сами учащиеся используют компьютеры физических классов образовательных учреждений или собственные индивидуальные устройства – ноутбуки, смартфоны, планшеты.

* Исследования выполнены при поддержке РФФИ (проекты №13-07-97046 и №14-07-97034)

Использование подобных виртуальных классов позволяет образовательным учреждениям не тратиться на закупку дорогостоящего ПО, необходимого в учебном процессе. ОРЦ предоставляет его в аренду в рамках виртуальных классов на время проведения занятий, при этом ПО может использоваться совместно в нескольких виртуальных классах с учетом лицензионных ограничений. Кроме того, образовательным учреждениям нет необходимости владеть мощными современными компьютерами. Они должны только поддерживать работу браузера для удаленного доступа к виртуальным классам и иметь быстрое подключение к сети Интернет, что приводит к значительной экономии денежных средств.

Помимо виртуальных классов в облаке OpenStack развернуто несколько виртуальных машин, содержащих сервера лицензий для платного ПО с лицензиями, ограничивающими количество одновременно работающих в сети экземпляров программ.

2. Проблемы создания образовательных ресурсных центров и их решение

Для реализации описанной концепции в ходе исследования должен быть решен ряд проблем.

Автоматическое создание и настройка виртуальных классов. Координатор образовательного учреждения должен иметь возможность создавать и настраивать виртуальные классы с помощью Web-узла ОРЦ. Он выбирает характеристики виртуальных машин, задает количество их экземпляров (размер класса), а также формирует набор необходимого программного обеспечения. ОРЦ должен создать виртуальный класс, установить и настроить все необходимое ПО автоматически, без участия координатора. Для решения данной задачи мы планируем использовать встроенные средства OpenStack для создания и управления виртуальными машинами, а также активно развивающуюся в настоящий момент SCM-систему (Software Configuration Management) Chef [2] для автоматической установки пакетов программ и их конфигурирования. При этом для программ, поддерживающих тихую установку, будут созданы соответствующие файлы ответов и/или использованы ключи для запуска инсталляторов.

Планирование виртуальных классов и виртуальных машин. Для ОРЦ должны быть разработаны алгоритмы планирования, которые учитывают: необходимость одновременного запуска к началу занятия всех виртуальных машин класса и их одновременную остановку по окончании, четкие временные интервалы начала и окончания занятий без возможности их динамического изменения, недельную периодичность составляемых расписаний, ограничения физических серверов, ограничения сетевых лицензий ПО, используемого в виртуальных машинах.

Стандартные алгоритмы планирования виртуальных машин OpenStack не учитывают описанные факторы. В рамках данного исследования будут предложены метаэвристические алгоритмы планирования, основанные на учете топологии и состоянии сети ЦОД при назначении виртуальных машин группы на физические узлы. Планируется рассмотреть случаи, когда:

- а) наибольшая часть трафика приходится на подключение удаленных пользователей к виртуальным машинам через граничные узлы;
- б) наибольшая часть трафика приходится на обмен данными внутри группы виртуальных машин;
- в) смешанный вариант, сочетающих оба варианта.

Для предварительной оценки эффективности предложенных алгоритмов в настоящее время разрабатывается симулятор ОРЦ. Для него в ходе теоретических исследований созданы структурная и имитационные модели ОРЦ. Применение симулятора позволит оценить работу алгоритмов для различных аппаратных конфигураций центра обработки данных, ограничений ПО, потоков заявок координаторов.

На следующем шаге лучшие варианты предложенных алгоритмов будут исследованы в рамках реального ОРЦ, который планируется построить на базе оборудования лаборатории облачных технологий нашего университета.

Эффективная прокладка маршрутов передачи данных между ОРЦ и граничными шлюзами, обеспечение QoS. Для обеспечения приемлемого уровня интерактивности в условиях передачи изображения удаленного рабочего стола необходимо задание параметров QoS для потоков

данных – минимальной гарантированной пропускной способности и максимальной гарантированной задержки. Также необходима адекватная прокладка маршрутов передачи данных, чтобы не нарушать требования к QoS других потоков и обеспечивать возможность динамического изменения маршрутов в случае миграции виртуальных машин.

Данная задача будет решена с использованием технологии SDN (Software Defined Networks) [3]. С помощью OpenFlow [3] для контроллера Floodlight будет реализован разрабатываемый авторами алгоритм маршрутизации и обеспечения QoS. Планируется исследование разработанного алгоритма маршрутизации обеспечения QoS в условиях реального ОРЦ.

3. Разработка симулятора образовательного ресурсного центра

Для экспериментального исследования эффективности разрабатываемых алгоритмов планирования виртуальных машин и виртуальных окружений, а также методов маршрутизации данных, необходимо создание симулятора образовательного ресурсного центра, основанного на ПКС. В его основе лежит имитационная модель, описываемая в данном разделе. Она включает:

- а) моделирование обработки заявок пользователей;
- б) моделирование сетевого трафика в ПКС, возникающего в процессе обработки заявок.

На рис. 2 приведена имитационная схема процесса обработки заявок пользователей.

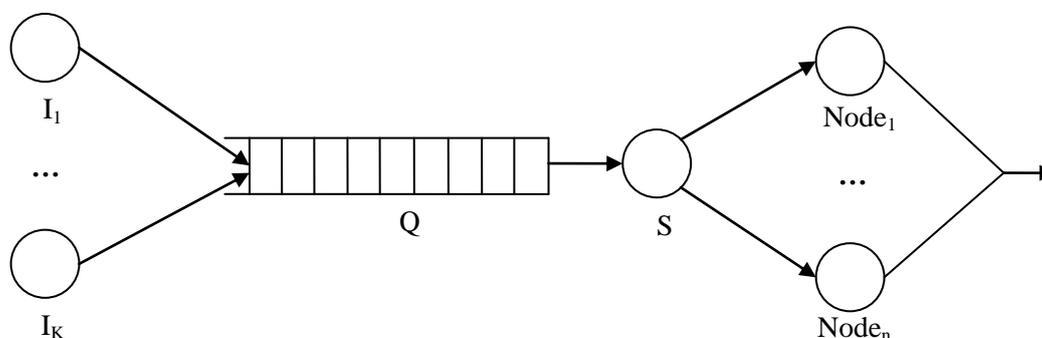


Рис. 2. Имитационная схема процесса обработки заявок пользователей

Каждый из источников заявок I_1, \dots, I_K представляет собой образовательное учреждение, подключенное к ресурсному центру. I_1, \dots, I_K совместно генерируют поток пользовательских заявок на обслуживание $J = \{J_j\}_{j=1, \dots, R}$. Каждая заявка J_j имеет следующий вид:

$$J_j = (a_j, t_j, Classroom_j),$$

где a_j – время поступления заявки, t_j – продолжительность времени, в течение которого виртуальный класс $Classroom_j$ должен быть доступен образовательному учреждению после развертывания. По истечению времени t_j виртуальный класс должен выгрузиться с физических узлов ЦОД образовательного ресурсного центра.

Все поступающие от пользователей заявки помещаются в очередь Q . Канал S представляет собой планировщик, который в соответствии с заложенным в него алгоритмом осуществляет извлечение заявок J_j из очереди Q , резервирование для них места в расписании (осуществляется выбор подмножества узлов $A_j \subseteq Nodes$ и момента времени T_j для их запуска). Набор всех резервирований формирует расписание образовательного ресурсного центра:

$$Schedule = \{(J_j, A_j, T_j)\}_{j=1, \dots, R}.$$

Обработка заявки J_j заключается в запуске соответствующего виртуального класса и при необходимости дополнительных виртуальных машин с серверами лицензирования на узлах множества A_j в текущий момент времени t и в его выгрузке в момент времени $t + T_j$.

Имитационная схема вычислительного узла p_i приведена на рис. 2, p_i соединен с другими узлами и коммутаторами вычислительной сети с помощью r_i дуплексных связей. Все входящие пакеты, а также пакеты сообщений, генерируемых назначенными узлу виртуальными машинами $\chi_{i,j}$, сначала поступают в очередь $Q_{inp,i}$, а затем маршрутизируются каналом обслуживания R_i . Если соответствующий пакет предназначен для локальному экземпляру виртуальной машины, то он передается ему непосредственно, иначе – помещается в одну из очередей $Q_{out,i,1}, Q_{out,i,2}, \dots, Q_{out,i,r_i}$, соответствующую выбранной алгоритмом маршрутизации исходящей связи.

При выполнении маршрутизации каналом R_i моделируется временная задержка. Заметим, если две виртуальные машины выполняются на одном узле, то обмен пакетами между ними также осуществляется через очередь $Q_{inp,i}$ и канал R_i .

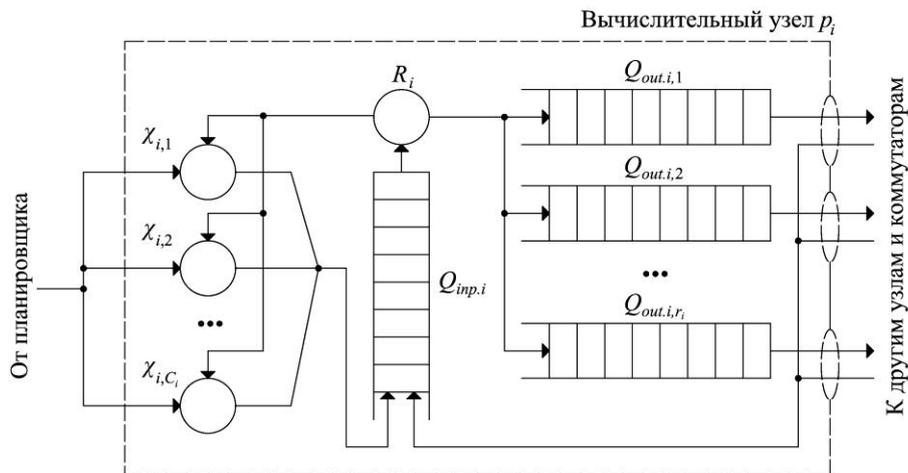


Рис. 2. Имитационная схема вычислительного узла

На рис. 3 приведена схема работы дуплексной связи $L_{ij} = \{E_{ij}, E_{ji}\}$, соединяющей сетевые устройства $d_i \in D$ и $d_j \in D$. Каналы обслуживания E_{ij} и E_{ji} добавляют к времени передачи пакета задержку величиной $b(E_{ij}) = b(E_{ji})$.

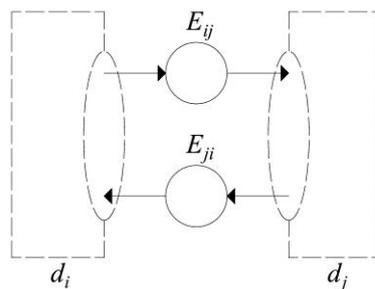


Рис. 3. Имитационная схема сетевой связи

Коммутатор OpenFlow f_i (см. рис. 4) отличается от обычного коммутатора тем, что его канал обслуживания R_i^{of} реализует поведение модуля OpenFlow – при получении нового пакета поиск и выполнение правил в таблице потоков, в случае отсутствия подходящего правила R_i^{of} отправляет пакет контроллеру OpenFlow, поместив его в очередь Q_{out,i,r_i+1}^{of} . Ответные команды контроллера в виде пакетов помещаются в очередь $Q_{inp,i}^{of}$, в которой для них задается максимальный приоритет обслуживания.

Контроллер OpenFlow установлен на выделенный сервер, его имитационная схема аналогична имитационной схеме вычислительного узла (см. рис. 5).

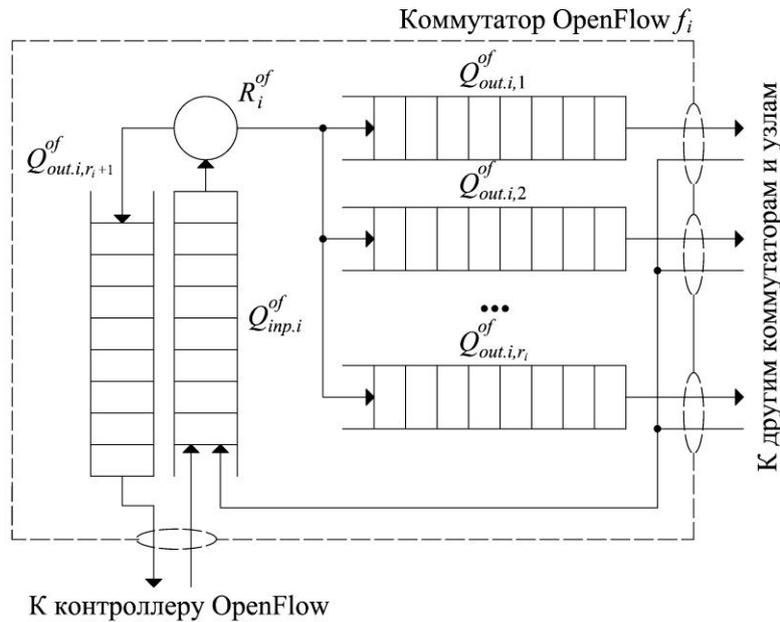


Рис. 4. Имитационная схема коммутатора OpenFlow

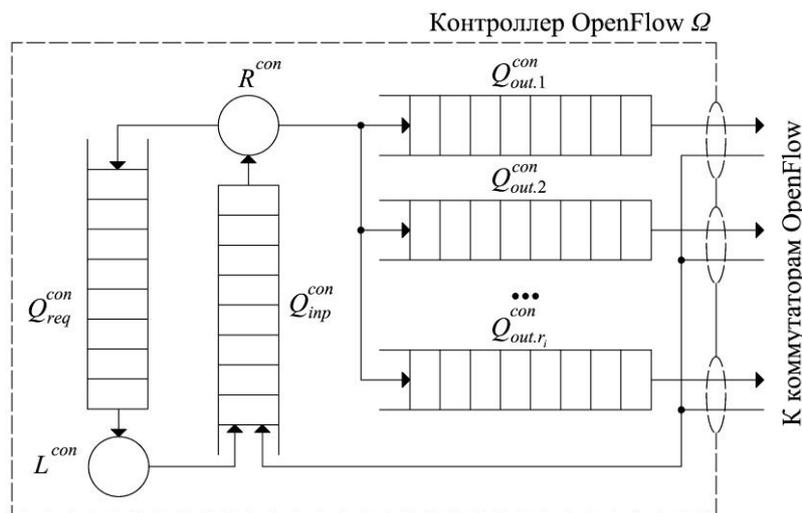


Рис. 5. Имитационная схема контроллера OpenFlow

Отличие в том, что канал R^{con} , помимо стандартной маршрутизации пакетов, помещает запросы к контроллеру OpenFlow в отдельную очередь Q_{req}^{con} . Канал обслуживания L^{con} реализует логику работы контроллера по принятию решений о дальнейших действиях с поступившим пакетом. Результатом его работы являются управляющие пакеты OpenFlow, передаваемые через Q_{inp}^{con} и R^{con} коммутаторам OpenFlow с целью отправки пакета на определенный порт или установки новых правил в их таблицы потоков. В Q_{inp}^{con} и очередях портов такие пакеты имеют максимальный приоритет.

Поток заявок на обслуживания определяется случайными величинами с законами распределения и параметрами, выбранными на основе существующих исследований [4]. Реализация данной модели в виде симулятора высокопроизводительной системы позволит исследовать эффективность предлагаемых в рамках проекта алгоритмических решений.

Для адекватной оценки эффективности работы образовательного ресурсного центра был разработан набор из следующих показателей:

а) Средняя загруженность вычислительных физических вычислительных ядер \bar{U} . Вычисляется, как среднее арифметическое средних загруженностей \bar{U}_{ij} каждого вычислительного ядра узлов ЦОД образовательного ресурсного центра:

$$\bar{U} = \frac{\sum_{i=1}^n \sum_{j=1}^{C_i} \bar{U}_{ij}}{\sum_{i=1}^n C_i}.$$

Единицами измерения \bar{U} и \bar{U}_{ij} являются проценты. Данный показатель позволяет оценить, насколько эффективно виртуальные машины виртуальных классов используют вычислительные ресурсы ЦОД образовательного ресурсного центра.

б) Индекс сбалансированности загрузки ресурсов физических вычислительных ядер σ . Может быть вычислен по формуле:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (\bar{U}_i - \bar{U})^2}{n-1}}.$$

Данный показатель позволяет определить равномерность использования вычислительных ресурсов. Будем считать, что имеет место сбалансированная загрузка, когда $\sigma \leq 10\%$.

в) Средняя пропускная способность образовательного ресурсного центра определяется как отношение количества обрабатываемых заявок R к интервалу времени ΔT_{sim} между окончанием выполнения последней заявки и приходом первой заявки:

$$v = \frac{R}{\Delta T_{sim}}.$$

Единица измерения – заявки в секунду. Данный показатель характеризует пропускную способность образовательного ресурсного центра.

Разработанных показателей достаточно, чтобы оценить основные параметры эффективности функционирования симулируемого образовательного ресурсного центра – производительности и сбалансированности.

4. Заключение

В рамках настоящего исследования предложена концепция ОРЦ, как средства организации экономически выгодного удаленного доступа образовательных учреждений к платному ПО. Определены основные проблемы создания образовательных ресурсных центров и их решение. Разработаны имитационная схема и показатели эффективности ОРЦ, которые станут основой создаваемого симулятора ОРЦ, который будет использован для исследования алгоритмов планирования виртуальных классов и виртуальных машин.

Литература

1. OpenStack Open Source Cloud Computing Software // Rackspace Cloud Computing, 2013. URL: <http://www.openstack.org/>
2. Chef // Opscode, 2013. URL: <http://www.opscode.com/chef/>
3. Intro to OpenFlow // Open Networking Foundation, 2011. URL: <https://www.opennetworking.org/standards/intro-to-openflow>
4. Feitelson, D. Workload Modeling for Computer Systems Performance Evaluation // Удаленный ресурс Еврейского университета Иерусалима. URL: <http://www.cs.huji.ac.il/~feit/wlmod/>