

## **АЛГОРИТМЫ КОМПЛЕКСНОЙ ОПТИМИЗАЦИИ ПОТРЕБЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ В ОБЛАЧНОЙ СИСТЕМЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ**

**В статье рассмотрены основные особенности организации распределенного доступа к данным с использованием облачной платформы развернутой в системе дистанционного обучения. Построены модели клиент-серверного взаимодействия компонентов облачных систем, использующих мультимедийные ресурсы. На базе построенных моделей разработаны алгоритмы комплексной оптимизации использования основных компонентов программных и аппаратных ресурсов и управления производительностью вычислительных узлов.**

**Ключевые слова:** мультимедийные образовательные ресурсы, распределение нагрузки, облачные вычисления, системы хранения данных, базы данных, миграция данных.

### **Введение**

На сегодняшний день одной из основных проблем при организации мультимедийных ресурсов является потребность в качественном предоставлении услуг конечным пользователям. Эта проблема особенно актуальна в сфере образования. Одним из наиболее развитых направлений, обеспечивающих широкополосный доступ к мультимедийным услугам, является обучение с применением дистанционных образовательных технологий. Ключевой особенностью применяемых технологий в процессе обучения в отличие от традиционных мультимедийных сервисов является возможность предоставлять различные сервисы, используя единый комплекс, обеспечивающий интерактивную связь с пользователем посредством информационных каналов связи. Это позволяет применять унифицированные решения в плане построения архитектуры таких сервисов. Кроме того, отличительной особенностью услуг, входящих в систему дистанционного обучения, является возможность прогнозировать поведение пользователей. Это обусловлено спецификой образовательного процесса, а также регламентом работы таких систем. Эти и другие факторы позволяют осуществить выбор оптимального решения, способного обеспечить высокое качество услуг, а также определить основные механизмы управления данными сервисами, учитывая специфику предметной области [6].

Наиболее перспективным направлением на сегодняшний день является использование технологий облачных вычислений для построения масштабируемых и высоконагруженных систем, таких как системы дистанционного обучения. На

рынке облачных вычислений присутствуют не только проприетарные решения, но и хорошо документированные комплексы с открытым исходным кодом, такие как XEN, OpenStack, CloudStack и другие.

В рамках исследования нами установлены следующие особенности потребления программно-аппаратных ресурсов, используемых для обеспечения работы системы дистанционного обучения (СДО) Оренбургского государственного университета (ОГУ):

- нагрузка на ключевые ресурсы носит периодический и неравномерный характер;
- пропускная способность внешних каналов связи ограничена и не позволяет предоставлять доступ к мультимедийному контенту с должным качеством обслуживания;
- одновременно происходят обращения к нескольким типам ресурсов;
- интенсивность обращения к каждому ресурсу может изменяться в зависимости от внешних условий;
- ввиду отсутствия распределения нагрузки между ресурсами при пиковой нагрузке оборудование не всегда позволяет обслужить все запросы;
- до 90% нагрузки предопределены, поскольку для доступа к ресурсам используется предварительная регистрация.

Кроме того, стоит отметить, что 80% ресурсов востребованы лишь в 20% времени работы сервисов [1].

В настоящее время существующие решения, построенные на базе облачных сервисов, используют универсальный подход для организации доступа к размещаемым в них ресурсам.

При этом не учитываются особенности каждого сервиса, что в свою очередь приводит к увеличению потребляемых ресурсов и неэффективному их использованию.

Целью исследования является определение ключевых параметров, влияющих на работу каждого из ресурсов, задействованных при построении системы дистанционного обучения и оптимизация их потребления с учетом решаемой ими вычислительной задачи.

Для достижения поставленной цели необходима разработка моделей, описывающих процессы взаимодействия между участниками сетевого взаимодействия с клиент-серверной облачной системой дистанционного обучения, а также алгоритмов, оптимизирующих выявленные недостатки.

В рамках проведенного исследования авторами решены следующие задачи:

- 1) разработка моделей клиент-серверного взаимодействия с основными компонентами облачной системы;
- 2) разработка алгоритмов комплексной оптимизации потребления вычислительных ресурсов облачной системы.

Для решения поставленных задач исследования используется системный подход, позволяющий определить основные компоненты, используемые в облачной системе, и основные характеристики каждого из участников процесса сетевого взаимодействия.

В результате проведенного исследования авторами разработаны:

1. Модель обслуживания запросов пользователей в системе дистанционного обучения.
2. Модель требований пользователей в облачном хранилище данных.
3. Алгоритм балансировки нагрузки в облачном хранилище данных.
4. Алгоритм интеллектуальной миграции данных в облачном хранилище.

Рассмотрим более подробно построенные модели и результаты, полученные в ходе работы алгоритмов оптимизации вычислительных ресурсов.

#### **Модель обслуживания запросов пользователей в системе дистанционного обучения**

Комплекс, обеспечивающий работу мультисервисного набора услуг для распределенных пользователей системы дистанционного обучения, предъявляет различные требования к при-

кладному программному обеспечению оборудования и качеству обслуживания (QOS) каждого из компонент.

Для повышения надежности и улучшения качества предоставляемых сетевых мультимедийных услуг требуется внедрение эффективных методов обеспечения распределения нагрузки аппаратно-программных ресурсов. Проанализировав интенсивность использования каждого из компонентов в СДО, нами получен рейтинг востребованности ключевых ресурсов:

- 1) канал связи;
- 2) система хранения данных;
- 3) система управления базами данных.

Для представленных в рейтинге ресурсов могут быть применены методы, позволяющие оптимизировать и повысить эффективность обслуживания запросов, поступающих от пользователей. При этом следует учитывать индивидуальные характеристики выбранного ресурса и алгоритмы его работы.

На сегодняшний день эффективным считается прогнозирование поведения клиентов. Используя механизмы предварительной регистрации (подписки на сервисы), а также статистику потребления ресурсов по каждой из подсистем, мы можем предопределить объем необходимых вычислительных мощностей, требуемых для обслуживания поступающего потока заявок [1].

Однако, в виду того, что требуется одновременное обслуживание нескольких типов заявок, поступающих в разные каналы обслуживания, необходимо эффективное управление потоками запросов, поступающих на широкополосные мультимедийные ресурсы системы дистанционного обучения. Так как представленные подсистемы являются веб-сервисами, их можно описать как систему массового обслуживания с ограниченным временем пребывания в очереди и пуассоновским потоком заявок?, причем длительность процедуры обслуживания каждым из каналов является случайной величиной, подчиненной экспоненциальному закону распределения [4,5]. Отличительной особенностью имитационной модели, построенной для исследования процесса обслуживания заявок в данной предметной области, является неоднородный поток событий, поступающий на вход системы. Это связано со следую-

щим набором признаков, характеризующих каждую из поступающих заявок:

– ресурсоемкость – оценивается с использованием рейтинга востребованности основных ресурсов системы;

– предполагаемое время выполнения – оценивается с использованием статистики обслуживания однотипных заявок в зависимости;

– рейтинг конечного исполнителя заявки – используется в качестве весового коэффициента для рационального распределения ресурсов с использованием уровневой модели подсистем [1].

Каждая из заявок во входном потоке данных получает динамический приоритет, в зависимости от представленного набора признаков и текущего состояния всей СМО в целом. Все каналы (К) обслуживания, в рамках выбранного класса решаемой задачи, идентичны, и любая заявка может быть обслужена свободным каналом. При этом, в каждом из каналов, для эффективного обслуживания заявок применяют относительные приоритеты.

Несмотря на то, что в модели можно четко классифицировать поступающие заявки, использование группового режима обработки в качестве дисциплины обслуживания не эффективно, так как не позволяет обеспечить равноценное качество для всех представленных мультимедийных сервисов.

Учитывая особенности каждого сервиса, формализуем характеристики построенной модели. Количество источников  $I$ , и их интенсивность  $\mu_n$ ,  $n=1, \dots, I$  напрямую зависит от количества пользователей, обращающихся в данный момент к облаку СДО, причем в случае одновременного обращения одного клиента к разным уровням подсистем будем считать заявки, как поступившие от двух независимых друг от друга источников. Учитывая это, интенсивность  $\mu$  поступления заявок в облачную систему в целом будет неравномерной независимо от выбранного интервала времени моделирования. Поэтому моделирование будем проводить в переходном режиме функционирования СМО. Кроме того, в СМО облака можно выделить несколько фаз (F) обслуживания заявок. Это обусловлено архитектурой технического решения, позволяющего масштабировать вычислительные мощности в зависимости от поставленных задач. Облачный контроллер, управляю-

щий размещением вычислительных задач на запущенных экземплярах приложений, а так же запуском/остановкой вычислительных узлов, способен определять классы задач, что дает возможность использовать гибкое управление потоками запросов.

В рамках исследования нами выделены три основные фазы обслуживания заявок:

– накопление заявок в контроллере облачной системы (F1);

– приоритетное обслуживание заявок на выбранном вычислительном узле (F2);

– генерация пакетов данных, запрашиваемых пользователями (F3).

Для увеличения максимального количества запросов пользователей в единицу времени и эффективной обработки очередей в представленной модели обслуживания запросов пользователей в системе дистанционного обучения введем целевую функцию вида:

$$\sum_{i=1}^M \sum_{j \in I_j(T_j)} \mu_i x_{ij} P_i \rightarrow \max,$$

где  $\mu_i$  – интенсивность поступающего потока заявок от источника (пользователя);

$x_{ij}$  – Статус обработки  $j$ -ой заявки поступившей на  $i$ -ый уровень подсистемы;

$P_i$  – динамический приоритет заявки в очереди на обслуживания в облачной системе.

#### **Модель требований пользователей в облачном хранилище данных**

В ходе исследования установлено, что единой точкой агрегации трафика выступает система хранения данных (СХД), обеспечивающая обработку потока запросов, поступивших от потребителей мультимедийных образовательных услуг. Следовательно, эффективность работы всей системы дистанционного обучения, а так же качество предоставляемых услуг напрямую зависит от производительности хранилища данных. Поэтому для эффективного управления потоком запросов нами разработана модель доступа к мультимедийным данным хранилища облачной системы.

Ключевым отличием хранилищ мультимедийных данных является неоднородность размещаемой информации (текстовые, аудио или видео данные) и, как следствие, разные подходы к организации доступа к ней. Помимо методов доступа к данным существенным является интенсивность обращения к тем или иным эле-

ментам, которая может быть получена с использованием внутрисистемных алгоритмов идентификации пользователей, что в свою очередь позволяет оценить востребованность и спрогнозировать нагрузку на устройства системы хранения [8]. В связи с этим важным аспектом управления ресурсами системы, при значительном увеличении количества одновременных запросов, является грамотная организация процесса размещения и распределение элементов данных по устройствам [2,3].

Отличительной характеристикой облачных хранилищ является реконфигурируемость их структуры в зависимости от потребляемых ресурсов. Это в свою очередь позволяет внедрять алгоритмы оптимизации в плане размещения данных внутри дискового пространства, а также управлять изменением количества используемых системой устройств. При этом процесс оптимизации размещения не должен приводить к снижению качества обслуживания клиентов СХД, для чего в алгоритмах необходимо учитывать пропускную способность сети и максимальный объем данных, который можно передавать в один момент времени [4]. Кроме того необходимо учитывать текущую загрузку самих устройств, а также их расположение относительно друг друга и клиентов, подключаемых к ним.

Для оптимизации механизмов обращения к данным построим общую модель доступа к системе хранения.

Пусть  $R=(U,M,Q)$ ,

где  $U = \{u_1, u_2, \dots\}$  – множество пользователей;

$M = \{m_1, m_2, \dots\}$  – множество уникальных элементов данных, размещаемых на устройствах хранения. При этом минимальной единицей данных  $m_i$  будем считать файл, имеющий обязательное свойство  $h$  – размер.

Для обеспечения безопасного хранения данных и балансировки нагрузки между устройствами хранения определим функцию распределения элементов данных, для этого введем множество  $M_c = \{m_1^{j_1}, m_1^{j_2}, m_1^{j_3}, \dots, m_2^{j_1}, m_2^{j_2}, m_2^{j_3}, \dots\}$ , где  $m_i^{j_k}$  –  $k$ -я копия элемента размещаемых данных ( $m_i$ ) на  $j_k$ -м устройстве хранения, при условии  $k \geq 3$  (не менее трех копий минимальной единицы хранения на различных устройствах).

Тогда функция распределения элементов данных по устройствам хранения принимает вид  $P: M_c \rightarrow D$ .

Исходя из изложенного выше, запишем требование пользователя к элементам данных

$$Q: U \rightarrow X \subseteq M_c,$$

где  $X$  – множество данных запрошенных множеством пользователей  $U$ .

Тогда хранилище данных можно записать в виде кортежа  $S=(M_c, D, P, L, C, R, G)$ , где

$D=\{d_1, d_2, \dots\}$  – множество устройств хранения;

$L=\{l_1, l_2, \dots\}$  – множество значений характеризующее загрузку каждого устройства хранения (количество одновременных обращений пользователей к конкретному устройству);

$C=\{c_1, c_2, \dots\}$  – множество значений, характеризующее объем каждого из устройств в хранилище;

$GON$  – натуральный коэффициент, характеризующий географический (топологический) приоритет использования хранилища.

Как правило, для крупных облачных структур используются консолидированные хранилища, состоящие из ферм, объединяющих несколько хранилищ в единый массив. Представим его как  $Sfarm=\{S_1, S_2, \dots\}$ .

Так как характеристики требований пользователей меняются во времени, преобразуем кортеж требований  $R(t)=(U, M_c, Q(t))$ . Тогда  $Q(t): U \rightarrow X \subseteq M_c$  – требования пользователя к элементам данных, меняющиеся во времени. Так как кроме активности пользователя изменяются свойства хранилища, запишем кортеж хранилища в зависимости от времени  $S(t)=(M_c(t), D(t), P(t), L(t), C, R(t), G)$ , где

$D(t)=\{d_1, d_2, \dots\}$  – множество устройств хранения, меняющихся во времени, таких что  $\forall t, D(t) > 0$ ;

$P(t): M_c \rightarrow D$  – функция распределения элементов данных по устройствам хранения, меняющаяся во времени.

При этом, для оптимизации затрат на аппаратные ресурсы и сокращения количества одновременно используемых устройств введем кортеж отношений  $Scloud(t)=\{S(t), D(t), Duse(t)\}$ , где  $\forall t, Duse(t) \subseteq D(t)$  множество устройств хранения в масштабируемом хранилище  $S$  в момент времени  $t$ . Кроме того, при масштабировании хранилища и миграции данных должно выполняться условие  $\forall t, i, j \neq j \Rightarrow D_i(t) \cap D_j(t) = 0$ , т. е. при миграции данных хранилища не должны использовать одни и те же устройства. Это позволит как гарантировать

скорость обработки информации, так и обеспечить приемлемое время реконфигурации.

Таким образом, для минимизации количества одновременно используемых устройств хранения в рамках одного масштабируемого хранилища и максимизации количества обработанных запросов пользователей в единицу времени, введем целевую функцию вида:

$$\sum_{i=1}^N P_i(t) \rightarrow \min$$

$$\sum_{i=1}^N L_i P_i(t) R_i(t) \rightarrow \max$$

где  $i=1, \dots, N$  – количество заявок, поступивших в систему на интервале времени  $\Delta T$ .

**Алгоритм балансировки нагрузки в облачном хранилище данных**

На основе модели требований пользователей в хранилище данных нами разработан алгоритм балансировки нагрузки между устройствами, реализованный в виде программного модуля с использованием API-средств для компонента Swift облачной системы OpenStack [11]. Выбор данной облачной системы обусловлен открытостью ее архитектуры и возможностью ее модификации под поставленные задачи. Основными недостатками OpenStack является неэффективный алгоритм распределения вычислительных задач между узлами хранения данных. Стандартный алгоритм, предложенный в системе, не учитывает маршрутизацию виртуальной и топологию локальной сети, а также удаленность виртуальных машин, выполняющих обработку запро-

сов пользователей и сетевых хранилищ, обеспечивающих передачу данных [7,8]. Все это негативно влияет на время отклика, как самой облачной системы, так и запущенных в ней экземпляров приложений. Кроме того, сами алгоритмы распределения данных, применяемые в хранилище облачной системы, не позволяют эффективно осуществлять размещение информации и предоставлять доступ к востребованным данным по сети [10].

Разработанный нами алгоритм позволяет снизить время отклика, используя информацию о топологии и маршрутизации основных потоков данных, а гибкое управление их размещением позволяет сократить накладные расходы вычислительных мощностей при миграции данных и виртуальных машин [9,10].

Для оценки эффективности разработанного алгоритма проведено моделирование работы системы хранения с различными параметрами. При этом получены следующие закономерности при работе стандартных алгоритмов облачной системы.

При увеличении количества копий данных происходит значительное снижение нагрузки на основных устройствах хранения. Однако, при этом возрастает количество задействованных устройств, что не соответствует поставленной задаче.

При одновременном доступе к нескольким устройствам, содержащим разный объем данных, возникает дисбаланс производительности хранилища, что приводит к отказам в обслуживании запросов пользователя. Основной причиной является неравномерное размещение больших и малых по объему данных, что в свою



Рисунок 1. Производительность обслуживания заявок в хранилище данных с применением алгоритма интеллектуального кэширования

очередь увеличивает время занятости устройств.

При многократном обращении к одним и тем же данным устройства, содержащие востребованные элементы, не в состоянии обслужить запросы, так как отсутствует распределение нагрузки между узлами. При этом, применяемые в СХД алгоритмы кеширования, не могут эффективно предоставить доступ к таким данным.

Разработанный нами алгоритм позволяет учесть перечисленные недостатки работы системы управления хранением данных, что в свою очередь, с учетом алгоритма приоритетного обслуживания [9], дает дополнительный прирост производительности облака и решаемых в нем задач на 5-9%, по сравнению со стандартными средствами управления хранилищем данных в OpenStack.

На рисунке 1 представлен график обслуживания заявок, поступающих на вычислительные узлы облачной системы дистанционного обучения в заданный интервал времени. На нем наглядно продемонстрировано пропорциональное увеличение количества обслуженных запросов от клиентов, что говорит об эффективности предлагаемого решения.

#### Алгоритм интеллектуальной миграции данных в облачном хранилище

Помимо алгоритма распределения нагрузки немаловажным фактором, влияющим на производительность системы хранения данных, является процесс миграции данных между устройствами хранения. Данная операция оказывает существенное влияние на время отклика системы, так как размещаемые в хранилище данные, как отмечалось ранее, являются неоднородными, а некоторые из них являются еще и зависимыми друг от друга. Это особенно актуально при обращении к потоковым данным, например, при проведении видеотрансляции. Кроме того, процесс тиражирования (процесс распределения данных между устройствами), в том числе для кеширования наиболее востребованных данных

так же напрямую зависит от эффективности алгоритмов, применяемых при миграции данных. Для оптимизации данного процесса, используя возможности облачной системы OpenStack, нами разработан алгоритм формирующий план миграции данных, а также модуль, осуществляющий распределенную обработку, созданных в плане вычислительных задач.

При этом все операции, задаваемые подсистемой планирования, можно описать как граф требований  $G$ , где  $V$  – направление перемещения (оконечное устройство),  $E$  элемент данных (файл), востребованный на устройстве,  $P$  – приоритет выполнения операции в плане миграции. В общем виде схема взаимодействия ресурсов в процессе миграции данных может быть представлена в виде следующей схемы (рисунок 2).

При формировании плана миграции одной из основных особенностей является использование приоритетного подхода при выборе операции. Помимо этого учитываются следующие показатели:

1. Текущая загруженность узлов;
2. Результаты прогнозирования нагрузок, опирающиеся на историю обращений пользователей к тем или иным элементам данных, а также на интеллектуальные алгоритмы внутрисистемной авторизации пользователей.
3. Размер и тип востребованных элементов данных;
4. Пропускная способность каналов связи как внешних, так и внутренних (в зависимости от направления миграции данных).
5. Востребованность активных данных, используемых в текущий момент (количество пользователей обращающихся к одному и тому же ресурсу в независимости от его расположения в распределенной системе хранения).

Для составления вычислительных задач по миграции, планировщиком выделяются множества независимых операций  $DM_j$ , где  $j=1..N$  ( $N$  – количество параллельно выполняемых опера-

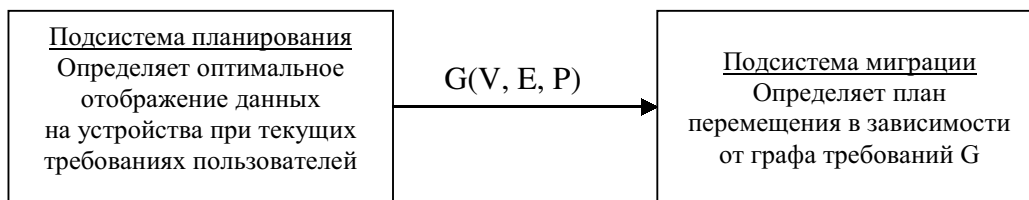


Рисунок 2. Схема взаимодействия подсистемы планирования и подсистемы миграции

ций в хранилище). Выбор и объединение операций в каждом множестве определяется, как связностью устройств, участвующих в текущей операции, так и связностью направления миграции с другими задачами. Каждому множеству  $DM_j$  назначается приоритет, равный максимальному приоритету операции, входящей в данное множество. Множества упорядочиваются в соответствии с расставленными приоритетами. В ранжированном списке вычислительных задач выделим два ключевых множества и обозначим их как  $DM_c$  и  $DM_{nc}$ . В множество  $DM_c$  отнесем наиболее критичные операции в плане времени выполнения, в  $DM_{nc}$  все остальные. Разработанный нами планировщик вычислительных задач направлен на параллельную обработку двух подмножеств. При этом, на каждом этапе выполнения вычислительных задач производится анализ связей операций каждого из множеств, а также составляется обновленный ранжированный список приоритетов миграции, с учетом показателей приведенных ранее. Таким образом, разработана система реального времени, отслеживающая состояние устройств, размещенных на них данных, а также запросы пользователей.

Проведя опытную эксплуатацию с использованием основных мультимедийных сервисов СДО Оренбургского государственного университета, нами получен суммарный прирост производительности облака на 15-19% относительно стандартных средств, применяемых в OpenStack. График, представленный на рисунке 3, показывает суммарную производительность обслуживания заявок на вычислительных узлах облачной системы с тремя запущенными подсистемами СДО (контроля знаний, электронной библиотеки и видеопортала) на интер-

вале времени работы стандартной web-сессии. Наглядно видно преимущество использования авторского алгоритма по сравнению со стандартными средствами облачной системы.

### Обсуждение полученных результатов

Построенные в рамках проведенного исследования модели и разработанные на их базе алгоритмы позволяют:

1. Описать клиент-серверное взаимодействие между основными компонентами облачной системы.
2. Учитывать особенности мультимедийных ресурсов, размещенных в облаке.
3. Оптимизировать использование вычислительных узлов за счет балансировки нагрузки между основными ресурсами системы дистанционного обучения.
4. Повысить производительность обслуживания запросов клиентов за счет эффективного размещения данных на устройствах хранения.
5. Учитывать виртуальную и физическую топологию облачной вычислительной системы для оптимизации миграции данных между устройствами хранения.

### Вывод

Представлены модели клиент-серверного взаимодействия основных компонентов облачной системы дистанционного обучения, состоящие из модели требований пользователей в облачном хранилище данных и модели обслуживания запросов пользователей в системе дистанционного обучения. Модели являются теоретической основой разработанных алгоритмов комплексной оптимизации вычислительных ресурсов, состоящих из алгоритма балансировки нагрузки и алгоритма интеллектуаль-

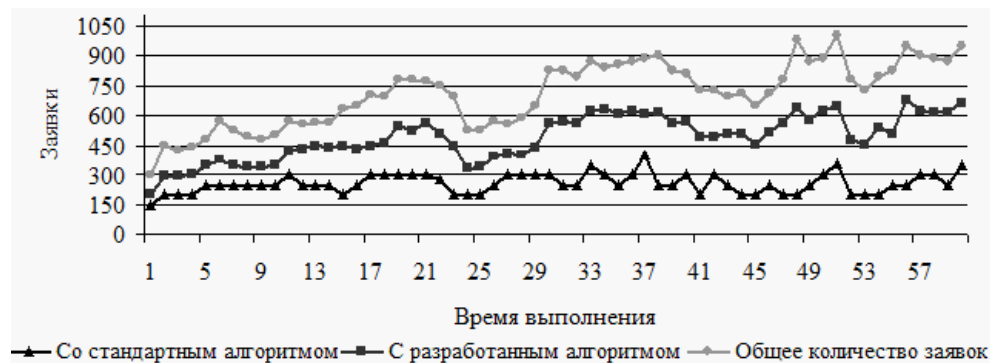


Рисунок 3. Производительность обслуживания заявок в хранилище данных с применением алгоритма приоритетной миграции данных

ной миграции данных. Оба алгоритма совместно обеспечивают эффективное разделение вычислительных задач между доступными вычислительными узлами. При этом учитывается их производительность, текущая загруженность, топологическая (географическая) удаленность относительно клиента и других узлов, а также тип сервиса запущенного на дан-

ном экземпляре виртуальной машины. Это позволяет масштабировать облако, не выделяя при этом дополнительные объемы аппаратных ресурсов, а также уменьшение времени обработки пользовательских запросов за счет сокращения задержек обработки очередей, не требуя увеличения пропускной способности каналов связи.

28.06.2013

**Работа выполнена при поддержке Минобрнауки РФ (ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы, №14.В37.21.1881 и №14.132.21.1801) и Российского фонда фундаментальных исследований (проекты №13-07-00198 и №13-01-97050)**

**Список литературы:**

1. Болодурина И.П., Решетников В.Н., Парфёнов Д.И. Распределение ресурсов в информационной системе дистанционной поддержки образовательного процесса // Программные продукты и системы – 2012. – №3. – С. 151-155.
2. Петров Д.Л. Оптимальный алгоритм миграции данных в масштабируемых облачных хранилищах // Управление большими системами. 2010. – №30. – С.180-197.
3. Петров Д.Л. Динамическая модель масштабируемого облачного хранилища данных // Известия ЛЭТИ 2010. – №4. – С. 17-21.
4. Гусев О.В., Жуков А.В., Поляков В.В., Поляков С.В. Проблема адекватной оценки производительности веб-серверов в корпоративных сетях на предприятиях ЦБП // Материалы 6-й научно-технической конференции «Новые информационной технологии в ЦБП и энергетике» / Петрозаводск, 2004. – С. 84-87.
5. Жуков А.В. Некоторые модели оптимального управления входным потоком заявок в интранет-системах. // Материалы 6-й научно-технической конференции «Новые информационной технологии в ЦБП и энергетике» / Петрозаводск, 2004. – С. 87-90.
6. Бойченко И.В., Корытников С.В. Управление ресурсами в сервис-ориентированных системах типа «приложение как сервис» // Доклады Томского государственного университета систем управления и радиоэлектроники, Вып. 1-2, 2010. – С. 156-160.
7. Тарасов В.Н., Полежаев П.Н., Шухман А.Е., Ушаков Ю.А., Коннов А.Л. Математические модели облачного вычислительного центра обработки данных с использованием OpenFlow // Вестник Оренбургского государственного университета. – 2012. – №9. – С. 150-155.
8. Полежаев П.Н. Математическая модель распределенного вычислительного центра обработки данных с программно-конфигурируемыми сетями его сегментов // Вестник Оренбургского государственного университета. – 2013. – №5. – С. 174-180.
9. Парфёнов Д.И. Сравнение эффективности алгоритмов динамического распределения данных в облачных хранилищах системы дистанционного обучения // Системы управления и информационные технологии. – 2012. – №4.1. – С. 163-168.
10. Парфёнов Д.И. Сравнение эффективности алгоритмов динамического распределения данных в гибридных облачных системах дистанционного обучения // Информационные технологии моделирования и управления, – 2012. – №6. – С. 491-498.
11. OpenStack Object Storage API v1 Reference. [Электронный ресурс]. – Режим доступа: <http://docs.openstack.org/api/openstack-object-storage/1.0/content/>

Сведения об авторах:

**Болодурина Ирина Павловна**, заведующая кафедрой прикладной математики Оренбургского государственного университета, доктор технических наук, профессор  
**Парфёнов Денис Игоревич**, начальник отдела программно-технической поддержки дистанционного обучения Оренбургского государственного университета  
460018, г. Оренбург, пр-т Победы, 13, ауд. 3208, тел. (3532) 372536; 375932,  
e-mail: prmat@mail.osu.ru; fdot\_it@mail.osu.ru

**UDC 519.687**

**Bolodurina I.P., Parfenov D.I.**

Orenburg state university, e-mail: prmat@mail.osu.ru; fdot\_it@mail.osu.ru

**ALGORITHM OF COMPLEX OPTIMIZATION OF THE USE COMPUTING RESOURCES IN THE CLOUD SYSTEM DISTANCE LEARNING**

The article describes the main features of the organization of distributed data access using the cloud platform deployed in distance learning system. A model of client-server interaction components of cloud systems using multimedia resources. On the basis of the constructed models of complex algorithms for optimizing the use of the main components of software and hardware resources and performance management of computing nodes.

Key words: multimedia educational resources, load balancing, cloud computing, data storage systems, databases, data migration.