УДК 37.018.43:004.75:519.872

УПРАВЛЕНИЕ ПОТОКАМИ ЗАПРОСОВ ПРИ ДОСТУПЕ К ШИРОКОПОЛОСНЫМ МУЛЬТИМЕДИЙНЫМ ОБРАЗОВАТЕЛЬНЫМ РЕСУРСАМ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ[1]

И.П. Болодурина, Д.И. Парфенов

Представлены модель обслуживания заявок пользователей широкополосных мультимедийных образовательных ресурсов и модель доступа к данным хранилища гибридной облачной системы. Описано их применение для повышения эффективности использования вычислительных ресурсов в системе дистанционного обучения путем распределения потоков запросов и балансировки нагрузки между узлами облака с помощью разработанного дополнительного управляющего модуля для контроллера системы «OpenStack».

Ключевые слова: облачные вычисления, мультимедийные образовательные ресурсы, распределение нагрузки, гибридная облачная система «OpenStack».

ВВЕДЕНИЕ

Развитие информационных технологий и возрастающие потоки передаваемой информации требуют построения масштабируемых решений, способных обслуживать большое количество одновременных запросов пользователей. Одна из возникающих при этом задач состоит в обеспечении гибкого управления и эффективного использования выделенных для этих целей вычислительных ресурсов. Наиболее перспективное направление ее решения заключается в применении технологий облачных вычислений. Они позволяют унифицировать доступ не только к конечным данным, но и к ресурсу в целом, что очень важно для построения приложений, требующих поддержания высокого качества обслуживания и круглосуточной доступности сервиса. На рынке облачных вычислений присутствуют не только проприетарные решения, такие как «VMware ESX», «Xen» и др., но и хорошо документированные комплексы с открытым исходным кодом, такие как «OpenStack» [1]. Одним из направлений развития таких сервисов — создание мультимедийных ресурсов, осуществляющих трансляцию видео как в режиме реального времени, так и по запросу пользователя. Наиболее широкое распространение эти ресурсы получили при организации обучения с помощью дистанционных образовательных технологий. Как правило, образовательные ресурсы представляют собой многокомпонентную систему, решающую одновременно множество вычислительных задач для обеспечения работы различных информационных сервисов. Поэтому возникает необходимость разработки особых подходов как в организации управления инфраструктурой в целом, так и каждым компонентом в отдельности.

В результате исследования нами установлен ряд особенностей потребления программно-аппаратных ресурсов, используемых для обеспечения работы системы дистанционного обучения (СДО) Оренбургского государственного университета:

— 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;

[1] Работа выполнена при финансовой поддержке Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» (гранты № 14.B37.21.1881 и 14.132.21.1801), а также РФФИ (гранты № 13-07-00198 № 13-07-00198 и 13-01-97050).

— до 90 % нагрузки предопределены, поскольку для доступа к ресурсам используется предварительная регистрация.

Отметим, что 80 % ресурсов востребованы лишь 20 % времени работы сервисов.

В существующих решениях на базе облачных сервисов применяется универсальный подход к организации доступа к размещаемым в них ресурсам. Особенности каждого сервиса не учитываются, что приводит к увеличению потребляемых ресурсов и неэффективному их использованию. Цель нашего исследования заключается в определении ключевых параметров, влияющих на работу каждого из ресурсов, задействованных при построении системы дистанционного обучения, и оптимизации потребления ресурсов с учетом решаемой ими вычислительной задачи.

1. СХЕМА ПРЕДОСТАВЛЕНИЯ ДОСТУПА К МУЛЬТИМЕДИЙНЫМ ОБРАЗОВАТЕЛЬНЫМ РЕСУРСАМ ПОЛЬЗОВАТЕЛЯМ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

Ограничение пропускной способности выходного канала связи создает трудности применения широкополосных мультимедийных образовательных услуг в дистанционном обучении, особенно для пользователей, обращающихся к веб-приложениям из сети Интернет. Для анализа эффективности имеющейся архитектуры нами разработана трехуровневая модель подсистем СДО: уровень — подсистема контроля знаний, второй — подсистема предоставления учебно-методических комплексов (электронная библиотека) и третий — подсистема трансляции и публикации видео- и аудиоматериалов (видеопортал дистанционного обучения).

Комплекс подсистем, обеспечивающий работу мультисервисного набора услуг для физически *распределенных пользователей*, предъявляет различные требования к прикладному программному обеспечению оборудования и качеству обслуживания на каждом из уровней модели.

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 для обеспечения необходимого качества предоставляемого сервиса.

Эффективным считается прогнозирование поведения клиентов. Используя механизмы предварительной регистрации (подписки на сервисы), а также статистику потребления ресурсов по каждой из подсистем, можно предопределить объем необходимых вычислительных мощностей, требуемых для обслуживания поступающего потока заявок [2].

Поскольку 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 (СМО) с ограниченным временем пребывания в очереди и пуассоновским потоком заявок λ, причем длительность процедуры обслуживания каждым из каналов является случайной величиной, подчиненной экспоненциальному закону распределения [3, 4]. Отличительная особенность имитационной модели, построенной для исследования процесса обслуживания заявок в данной предметной области, состоит в неоднородном потоке событий, поступающем на вход системы. Это связано с рядом признаков, характеризующих каждую из поступающих заявок:

— ресурсоемкость — оценивается по рейтингу востребованности основных ресурсов системы;

— предполагаемое время выполнения — оценивается с помощью статистики обслуживания однотипных заявок в зависимости;

— рейтинг конечного исполнителя заявки — учитывается в качестве весового коэффициента для рационального распределения ресурсов в соответствии с приведенной ранее уровневой модели подсистем.

Каждая из заявок во входном потоке данных получает динамический приоритет, в зависимости от представленных признаков и текущего состояния всей СМО в целом. Все каналы K обслуживания, в рамках выбранного класса решаемой задачи, идентичны и любая заявка может быть обслужена любым свободным каналом. В каждом из каналов для эффективного обслуживания заявок применяются относительные приоритеты.

Несмотря на то, что в модели можно четко классифицировать поступающие заявки, групповой режим обработки в качестве дисциплины обслуживания не эффективен, так как он не позволяет обеспечить равноценное качество для всех представленных мультимедийных сервисов.

Учитывая особенности каждого сервиса, формализуем характеристики построенной модели. Число

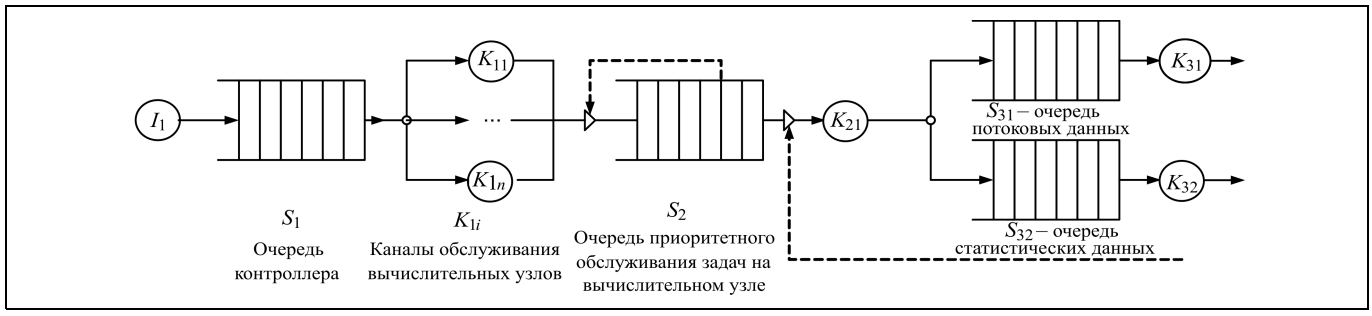

Рис. 1. Схема управления потоками запросов в системе дистанционного обучения

источников I и их интенсивности μ_n, $n = 1, ..., I$, напрямую зависит от числа пользователей, обращающихся в данный момент к облаку СДО. 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 μ 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, управляющий размещением вычислительных задач на запущенных экземплярах приложений, а также запуском/остановкой вычислительных узлов, способен определять классы задач, что дает возможность гибкого управления потоками запросов.

Определим схему управления потоками запросов (рис. 1) и выделим три фазы обслуживания заявок: накопление заявок в контроллере облачной системы первая фаза, приоритетное обслуживание заявок на выбранном вычислительном узле вторая фаза, генерация пакетов данных, запрашиваемых пользователями третья фаза.

Рассмотрим каждую из фаз обслуживания потока заявок более подробно. При поступлении заявки в систему дистанционного обучения она попадает в очередь S_1 облачного контроллера для последующего распределения по каналам K_{1i}, $i = 1, ..., n$, обслуживания вычислительных узлов (n — число узлов). Длина очереди S_1 в данном случае не ограничена, так как время пребывания заявки в очереди на обслуживание фиксировано, что обусловлено принципом работы веб-приложений. Число узлов, выступающих каналами обслуживания K_{1i}, напрямую зависит от текущей загруженности облака, объема решаемых задач, а также числа экземпляров каждой из подсистем, запущенных в данный момент. После того, как контролер выбрал доступный вычислительный узел, используя алгоритм минимизации потребляемых ресурсов и максимизации обслуживания клиентов, заявка поступает в очередь приоритетного обслуживания S_2. В соответствии с алгоритмом приоритетного обслуживания заявки поступают в канал K_{21} для выполнения запрошенных вычислительных операций. Для генерации и передачи обработанного пакета данных запросы пользователей направляются в одну из двух очередей: для потоковых данных или для статических данных. Откуда они в соответствии алгоритмом предоставления доступа к данным в системе хранения облака предаются в каналы обслуживания K_{31} или K_{32} соответственно.

Выбор и расстановка приоритетов для каждой заявки, поступающей в облако, базируется на модели обслуживания запросов пользователей мультимедийных образовательных ресурсов, построенной в рамках настоящего исследования.

2. МОДЕЛЬ ОБСЛУЖИВАНИЯ ЗАЯВОК ПОЛЬЗОВАТЕЛЕЙ ШИРОКОПОЛОСНЫХ МУЛЬТИМЕДИЙНЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ

Отличительная особенность облачных вычислений заключается в возможности управлять порядком выполнения вычислительных задач, используя различные алгоритмы обработки очередей. Это позволяет эффективно распределять запросы в высоконагруженных приложениях с критическим временем отклика.

Нами предложен алгоритм выбора и расстановки приоритетов для обработки поступающих заявок на второй фазе обслуживания запроса. В основу алгоритма положен расчет востребованности базовых ресурсов каждого из выделенных ранее уровней подсистем, входящих в СДО.

Численные показатели использования базовых ресурсов можно определить по формуле $R_{i\text{исп}} = R_i 100/(R_1 + ... + R_n)$, где $R_1, ..., R_n$ — численные показатели использования ресурса по каж-

дому из классификационных признаков, полученные в результате измерений на интервале времени ΔT [5].

Индикаторы приоритета обслуживания уровней модели определим на основе рейтинга востребованности ресурсов системы в целом. Общую ресурсоемкость системы дистанционного обучения определим как суммарную площадь $U_{\text{СДО}}$, занимаемую всеми уровнями модели (U_i). Максимально возможные ресурсы сервера обозначим как площадь, полученную при использовании 100 % всех ключевых сервисов (рис. 2).

Так как система работает непрерывно, поступление заявок к ее ресурсам можно описать в дискретном времени: $I_j(T_j) = \{j{:}t \in (0,\ T_i)\}$ — множество номеров заявок, пришедших в интервал времени $(0,\ T_i)$ на подсистему i (i — уровень подсистемы, $i = 1, ..., M$).

Статус обработки j-й заявки поступившей на i-й уровень обозначим x_{ij}. Будем считать, что отказу соответствует $x_{ij} = 0$, а успеху $x_{ij} = 1$.

Интенсивность поступления и обработки заявок на каждый из уровней модели обозначим λ_i, она напрямую зависит от ресурсоемкости подсистемы. Введем показатель приоритета P_i для каждого из уровней, распределение которого зависит от числа одновременно используемых ресурсов. Тогда на нагрузку, создаваемую каждым из уровней, можно наложить ограничение

$$\sum_{I_j(T_j)} U_i x_{ij} \leqslant H_i, \quad I = 1, ..., M.$$

При задании целевой функции введены ограничения, связанные с предметной областью исследования:

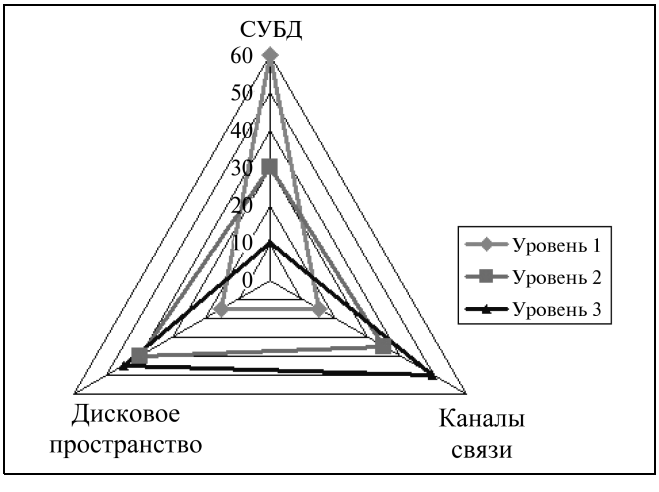

Рис. 2. Диаграмма приоритетов востребованности ресурсов системы дистанционного обучения

— время обработки T любого запроса ограничено;

— мощность сервера H фиксирована.

Ввиду неравномерности использования основных ресурсов каждым из уровней системы дистанционного обучения следует определить условия максимальной загрузки каждого из вычислительных узлов облака, при которой возможна безотказная работа всех запущенных экземпляров приложений:

$$\sum_{i=1}^{M} \sum_{j \in I_j(T_j)} U_i x_{ij} \leqslant H, \quad x_{ij} = \{0,\ 1\}.$$

Таким образом, для обработки максимального числа запросов пользователей в единицу времени получим целевую функцию вида

$$\sum_{i=1}^{M} \sum_{I_j(T_j)} \lambda_i x_{ij} P_i \to \max.$$

При выборе приоритетов оцениваются характеристики заявки:

— время нахождения заявки в очереди;

— текущая длина очереди заявок;

— интенсивность обращения к каждому из компонентов ресурса, необходимых для выполнения заявки.

Выбор приоритетов и оценка текущей ресурсоемкости задачи производятся на основе компонентов ресурса, имеющих индивидуальные пороговые значения, связанные с физическими ограничениями оборудования. В ходе исследования выполнено моделирование обслуживания заявок в облачной системе в рамках описанной модели. Была принята приоритетная обработка запросов на основе оценки востребованности ресурсов системы. Это позволило повысить эффективность работы компонентов системы благодаря сокращению времени пребывания заявки в очереди, что привело к сокращению ее длины. Эффективность предложенного решения оценивалась по значению отношения числа обслуженных заявок на выходе третий фазы к общему числу поступивших заявок на вход первой фазы, на интервале времени моделирования $\Delta T = 60$ с. В результате получен прирост на 12—15 % по сравнению со стандартными средствами обработки очередей (рис. 3).

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 также качество предоставляемых услуг напрямую зависит от производительности хранилища данных. Поэтому для эффективного управления потоком запросов необходимо разработать модель доступа

Рис. 3. Диаграмма обслуживания заявок в облачной системе

к мультимедийным данным хранилища гибридной облачной системы.

3. МОДЕЛЬ ДОСТУПА К ДАННЫМ ХРАНИЛИЩА ГИБРИДНОЙ ОБЛАЧНОЙ СИСТЕМЫ

Ключевое отличие хранилищ мультимедийных данных состоит в неоднородности размещаемой информации (текстовые, аудио или видеоданные), и, как следствие, существуют разные подходы к организации доступа к ней. Помимо методов доступа к данным, существенна интенсивность обращения к тем или иным элементам, значение которой может быть получено с помощью внутрисистемных алгоритмов идентификации пользователей, что, в свою очередь, позволяет оценить востребованность и спрогнозировать нагрузку на устройства системы хранения. В связи с этим важный аспект управления ресурсами системы, при значительном увеличении числа одновременных запросов, состоит в грамотной организации процесса размещения и распределения элементов данных по устройствам [6, 7].

Отличительной характеристикой облачных хранилищ является реконфигурируемость их структуры в зависимости от потребляемых ресурсов. Это позволяет внедрять алгоритмы оптимизации размещения данных внутри дискового пространства, а также управлять изменением числа используемых системой устройств. 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 [3]. Кроме того, необходимо учитывать текущую загрузку самих устройств, а также их расположение относительно друг друга и клиентов, подключаемых к ним.

Для оптимизации механизмов доступа к данным необходимо построить общую модель доступа к данным системы хранения. Пусть $R = (U, M, Q)$, где $U = \{u_1, u_2, ...\}$ — множество пользователей; $M = \{m_1, m_2, ...\}$ — множество уникальных элементов данных, размещаемых на устройствах хранения. 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 $M_c = \{m_1^{j_1}, m_1^{j_2}, m_1^{j_3}, ..., m_2^{j_1}, m_2^{j_2}, m_2^{j_3}\}$, где $m_i^{j_k}$ — k-я копия элемента размещаемых данных (m_i) на j_k-м устройстве хранения, при условии $k \geqslant 3$ (не менее трех копий минимальной единицы хранения на различных устройствах). Тогда функция распределения элементов данных по устройствам хранения принимает вид $P: M_c \to D$.

Исходя из изложенного, запишем требование пользователя к элементам данных. $Q: U \to X \subseteq M_c$, где X — множество данных запрошенных множеством пользователей U. Тогда хранилище данных можно описать кортежем $S = (M_c, D, P, L, C, R, G)$, где $D = \{d_1, d_2, ...\}$ — множество устройств хранения; $L = \{l_1, l_2, ...\}$ — множество значений, характеризующее загрузку каждого устройства хранения (число одновременных обращений пользователей к конкретному устройству); $C = \{c_1, c_2, ...\}$ — множество значений, характеризующее объем каждого из устройств в хранилище; $G \in N$ — 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 $S_{farm} = \{S_1, S_2, ...\}$.

Так как характеристики требований пользователей меняются во времени, преобразуем кортеж требований $R(t) = (U, M_c, Q(t))$. Тогда $Q(t): U \to X \subseteq M_c$ — 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 $D(t) = \{d_1, d_2, ...\}$ — множество устройств хранения, меняющихся во времени, таких что $\forall t\ D(t) > 0$; $P(t): M_c \to D$ — функция распределения элементов данных по устройствам хранения, меняющаяся во времени.

Для оптимизации затрат на аппаратные ресурсы и сокращения одновременно используемых устройств введем кортеж отношений $S_{cloud}(t) = \{S(t), D(t), D_{use}(t)\}$, где $\forall t\ D_{use}(t) \subseteq D(t)$ — множество уст-

ройств хранения, используемых в масштабируемом хранилище S в момент времени t. При масштабировании хранилища и миграции данных должно выполняться условие $\forall t, i, j, i \neq j \Rightarrow D_i(t) \cap D_j(t) = 0$, т. е. при миграции данных хранилища не должны использоваться одни и те же устройства. Это позволит как гарантировать скорость обработки информации, так и обеспечить приемлемое время реконфигурации.

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

$$\sum_{i=1}^{N} P_i(t) \to \min; \quad \sum_{i=1}^{N} L_i P_i(t) R_i(t) \to \max.$$

На основе модели доступа к данным хранилища нами разработан алгоритм балансировки нагрузки между устройствами, реализованный в виде программного модуля для компонента Swift облачной системы «OpenStack». Выбор данной облачной системы обусловлен открытостью ее архитектуры и возможностью ее модификации под поставленные задачи. Основной недостаток системы «OpenStack» заключается в неэффективном алгоритме 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 запросов пользователей, и хранилищ данных, обеспечивающих передачу данных. 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 размещать информацию и предоставлять доступ к востребованным данным по сети [5].

При исследовании алгоритма, применяемого в хранилище данных, нами получен ряд закономерностей, оказывающих существенное влияние на производительность СХД.

- При увеличении числа копий данных значительно снижается нагрузка на основных устройствах хранения. Однако при этом возрастает число 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. Основная причина — неравномерное размещение больших и малых по объему данных, что 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. Применяемые в СХД алгоритмы кеширования не могут эффективно предоставить доступ к таким данным.

Как правило, для распределения нагрузки и повышения эффективности работы масштабируемых хранилищ, помимо дублирования и перемещения данных между устройствами, также применяют систему кеш-областей (массивы устройств, обеспечивающих возможность быстрой обработки операций чтения/записи), построенных с использованием твердотельных SSD-накопителей или больших объемов оперативной памяти [5]. Однако, алгоритмы и методы использования таких ресурсов недостаточно эффективны. Чаще всего устройства кеш-области заполняются наиболее востребованными данными, при этом не учитывается модель поведения пользователя. Как правило, при обращении к мультимедийному сервису клиент отправляет последовательно несколько запросов для получения данных. В рамках мультимедийного образовательного сервиса можно предсказать набор запрашиваемых данных и порядок их получения, что позволяет построить прогноз и зарезервировать вычислительные ресурсы для решения поставленной задачи.

Учитывая изложенное, представим алгоритм, позволяющий оптимизировать доступ пользователя к мультимедийным данным.

Шаг 1. Получение входных параметров. При регистрации нового запроса, выделяются узлы (устройства хранения D), содержащие необходимые данные и анализируется их загрузка (L) и географический приоритет относительно клиента (G). Определяется тип (статические, динамические) и рейтинг востребованности запрошенных данных, составленный на основе статистики обращений.

Шаг 2. Обработка запроса. Для статических данных, используя полученные на шаге 1 показатели (G, L), определяется оптимальный узел. Для динамических данных осуществляется поиск необходимого элемента данных в кеш-области. Если он не найден, то производится процедура кеширования данных с оптимального узла, полученного с использованием показателей (G, L).

Далее, применяя алгоритм поиска связей, учитывающий рейтинг востребованности ресурсов, формируется перечень элементов, которые могут быть запрошены клиентом в ближайшее время. Для эффективной работы алгоритма осуществляется поиск наименее нагруженных узлов системы, содержащих необходимые данные, что в свою оче-

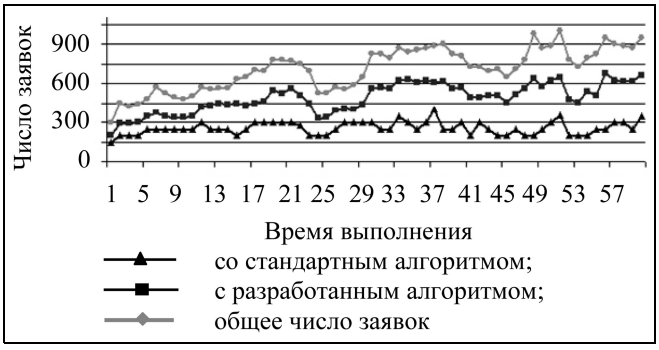

Рис. 4. Диаграмма обслуживания заявок в облачной системе при использовании алгоритма интеллектуального размещения данных

редь позволяет частично изолировать процесс кеширования от основных операций, производимых системой. Используя полученные данные, производится процедура кеширования. Число элементов зависит от востребованности начальных данных и общей нагрузки на систему.

Шаг 3. Передача данных. Запрошенные в текущий момент времени данные направляются пользователю из выбранного источника.

Шаг 4. Постобработка результатов. По окончанию работы алгоритма в базе данных хранимых ресурсов обновляется рейтинг востребованности использованных в обработке элементов.

Разработанный нами алгоритм позволяет снизить время отклика, используя информацию о топологии и маршрутизации основных потоков данных, а гибкое управление их размещением и кэшированием данных позволяет сократить накладные расходы вычислительных мощностей при миграции данных и виртуальных машин [8, 9].

Предложенный алгоритм учитывает перечисленные ранее недостатки работы стандартных средств системы управления хранением данных, что с учетом динамически формируемых приоритетов в каналах обслуживания дает прирост производительности облака и решаемых в нем задач. Относительно средств, используемых по умолчанию в «OpenStack», на 5—9 % увеличено число обслуженных заявок (рис. 4).

ЗАКЛЮЧЕНИЕ

Для оценки эффективности результатов разработанного модуля облачной системы «OpenStack» в качестве балансировщика нагрузки, проведено комплексное моделирование работы гибридной облачной системы с учетом особенностей работы компонентов мультимедийных ресурсов системы дистанционного обучения. Прогнозирование нагрузки и приоритезация очереди запросов позволяет управлять масштабированием облака, снижая при этом объемы задействованных в работе ресурсов, а применение алгоритмов оптимизации в системе хранения позволяет предоставлять эффективный доступ к пользователям независимо от запрошенного объема и типа данных. В результате комплексного моделирования процесса обслуживания потока заявок получен прирост производительности от 14 до 21 % по сравнению со стандартными средствами, что весьма эффективно при большой интенсивности запросов.

Разработанные оптимизационные модели управления могут применяться для дальнейшего исследования эффективности использования аппаратных и программных ресурсов в целях повышения качества предоставления услуг не только в распределенных информационных системах дистанционного обучения, но и для разработки мультимедийных ресурсов в целом.

ЛИТЕРАТУРА

1. *OpenStack* Open Source Cloud Computing Software. [Электронный ресурс]. — Режим доступа: http://www.openstack.org/ (дата обращения 29.09.2013).
2. *Болодурина И.П., Решетников В.Н., Парфенов Д.И.* Распределение ресурсов в информационной системе дистанционной поддержки образовательного процесса // Программные продукты и системы. — 2012. — № 3. — С. 151—155.
3. *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 науч.-техн. конф. «Новые информационной технологии в ЦБП и энергетике». — Петрозаводск, 2004. — С. 84—87.
4. *Жуков А.В.* Некоторые модели оптимального управления входным потоком заявок в интранет-системах // Там же. — С. 87—90.
5. *Математические* модели облачного вычислительного центра обработки данных с использованием OpenFlow / В.Н. Тарасов, П.Н. Полежаев, А.Е. Шухман и др. // Вестник Оренбургского гос. ун-та. — 2012. — № 9. — С. 150—155.
6. *Петров Д.Л.* Оптимальный алгоритм миграции данных в масштабируемых облачных хранилищах // Управление большими системами. — 2010. — № 30. — С. 180—197.
7. *Петров Д.Л.* Динамическая модель масштабируемого облачного хранилища данных // Известия ЛЭТИ. — 2010. — № 4. — С. 17—21.
8. *Парфёнов Д.И.* Сравнение эффективности алгоритмов динамического распределения данных в облачных хранилищах системы дистанционного обучения // Системы управления и информационные технологии. — 2012. — № 4.1 (50). — С. 163—168.
9. *Парфёнов Д.И.* Сравнение эффективности алгоритмов динамического распределения данных в гибридных облачных системах дистанционного обучения // Информационные технологии моделирования и управления. — 2012. — № 6 (78). — С. 491—498.

Статья представлена к публикации членом редколлегии А.С. Манделем.

Ирина Павловна Болодурина — д-р техн. наук, зав. кафедрой, ☎ (3532) 37-25-36, ✉ prmat@mail.osu.ru,

Денис Игоревич Парфенов — аспирант, ☎ (3532) 37-59-32, ✉ parfenovdi@mail.ru,

Оренбургский государственный университет.