АЛГОРИТМЫ ОПТИМИЗАЦИИ ПОТРЕБЛЕНИЯ РЕСУРСОВ В ОБЛАЧНОМ ХРАНИЛИЩЕ ДАННЫХ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

И.П. Болодурина, Д.И. Парфёнов

Аннотация. В рамках представленного исследования построена модель хранения и организации распределенного доступа к данным с использованием облачной платформы мультимедийных образовательных ресурсов, развернутых в системе дистанционного обучения. При этом основной задачей исследования является разработка алгоритмов и методов управления производительностью и оптимизация использования программных и аппаратных ресурсов.

Ключевые слова: мультимедийные образовательные ресурсы, распределение нагрузки, облачные вычисления, системы хранения 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 мультимедейным 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.

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, такие как VMware ESX, Xen и другие, но и хорошо документированные комплексы с открытым исходным кодом, такие как OpenStack.

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;
- 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.

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 нашего 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. Это определяет новизну настоящей работы.

2 Модель обслуживания запросов 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 элементам, которая может быть получена с использованием внутрисистемных алгоритмов идентификации пользователей, что в свою очередь позволяет оценить востребованность и спрогнозировать нагрузку на устройства системы хранения. 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 доступа к данным построим общую модель доступа к данным системы хранения.

Пусть $R = (U,M,Q)$,

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

$M = \{m_1,m_2,\ldots\}$ – 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 распределение элементов данных, для этого введем множество M_c.

$M_c = \{ m_1^{j_1}, m_1^{j_2}, m_1^{j_3}, \ldots, m_2^{j_1}, m_2^{j_2}, m_2^{j_3}, \ldots \}$,

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.

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

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

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

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

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

$G \in \boldsymbol{N}$ – 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 $S_{farm}=\{S_1,S_2,\ldots\}$.

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 $S(t)$.

$S(t)=(M_c(t),D(t),P(t),L(t),C,R(t),G)$,

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

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

При этом для оптимизации затрат на аппаратные ресурсы и сокращения одновременно используемых устройств введем кортеж отношений $S_{cloud}(t)$.

$S_{cloud}(t)=\{S(t),D(t),D_{use}(t)\}$,

где $\forall t$, $D_{use}(t) \subseteq D(t)$ множество устройств хранения используемых в масштабируемом хранилище S в момент времени t.

Кроме того, 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 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:

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

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

,

где N – общее количество заявок поступивших в систему на интервале времени ΔT.

3 Алгоритм балансировки нагрузки в облачном хранилище

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. 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 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 осуществлять размещение информации и предоставлять доступ к востребованным данным по сети [7].

Нами разработан алгоритм, который 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 [8,9].

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 причинной является неравномерное размещение больших и малых по объему данных, что в свою очередь увеличивает время занятости устройств.

При многократном обращении к одним и тем же данным в хранилище, устройства 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 [8] дает дополнительный прирост производительности облака и решаемых в нем задач на 5-9% по сравнению со стандартными средствами управления хранилищем данных в OpenStack (Рисунок 1).

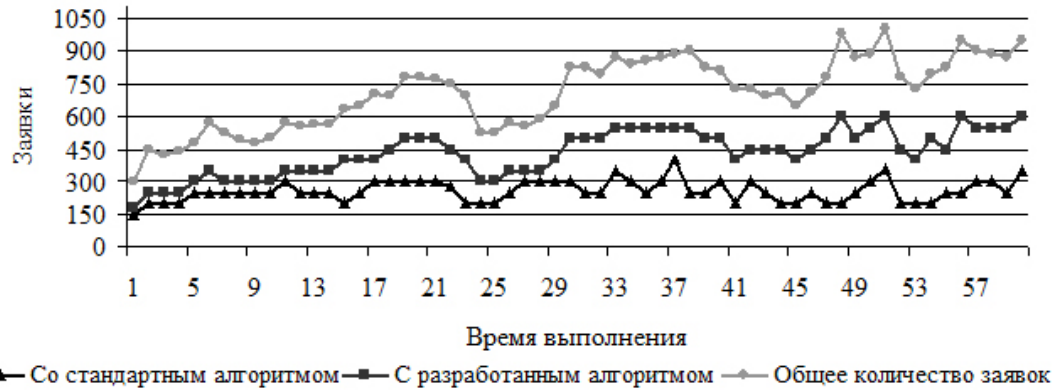

Рис. 1 – Производительность обслуживания заявок в хранилище данных с применением алгоритма интеллектуального кэширования

4 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 влияния 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 данных, например при проведении видео трансляции. 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 нами, используя возможности облачной системы OpenStack, разработан алгоритм, формирующий план миграции данных, а также модуль осуществляющий распределенную обработку созданных вычислительных задач.

При этом все операции задаваемые подсистемой планирования можно описать как граф требований G.

G (V,E,P),

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).

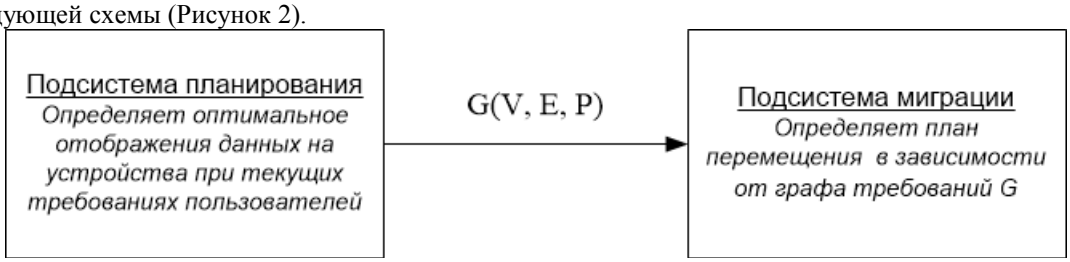

Рисунок 2 - Схема взаимодействия подсистемы планирования и подсистемы миграции

При формировании плана миграции одним 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 выделяется множества независимых операций DM_j. Выбор и объединение операций в каждом множестве определяется, во-первых, связностью устройств участвующих в текущей операции, во-вторых 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 DMc и DMnc. В множество DMc отнесем наиболее критичные операции в плане времени выполнения, в DMnc все остальные. Разработанный нами планировщик вычислительный задач направлен на параллельную обработку двух подмножеств. При этом, на каждом этапе выполнения вычислительных задач производится анализ связей операций каждого из множеств, а так же составляется обновленный ранжированный список приоритетов миграции, с учетом показателей приведенных ранее. Таким образом, нами разработана система реального времени, отслеживающая состояние устройств, размещенных на них данных, а также запросы пользователей.

Проведя опытную эксплуатацию системы, с применением алгоритмов кэширования и интеллектуальность миграции данных, нами получен суммарный прирост производительности облака OpenStack на 15-19%(Рисунок 3).

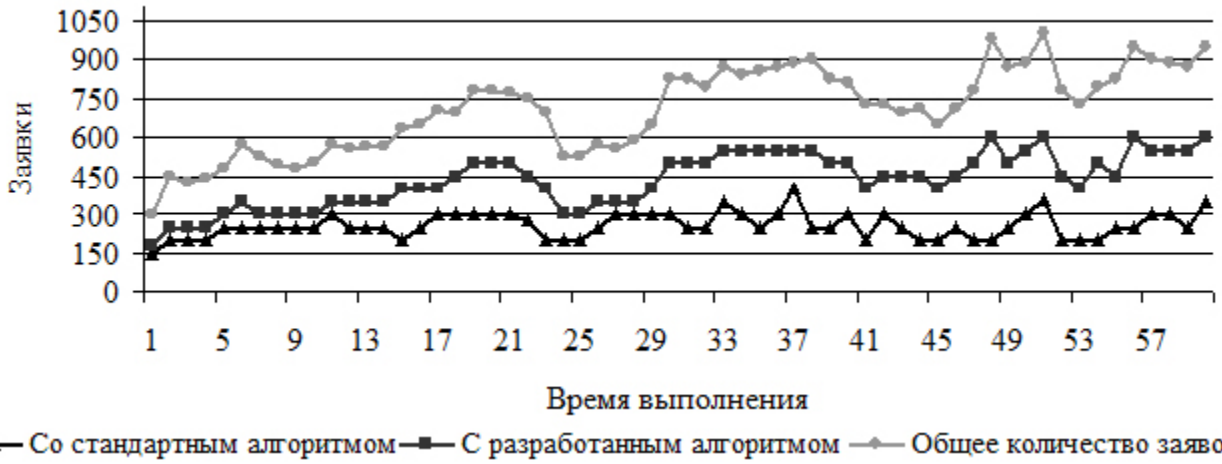

Рис. 3 Производительность обслуживания заявок в хранилище данных с применением алгоритма приоритетной миграции данных

Заключение

Комплексное моделирование работы облачной системы проводилось учетом особенностей компонентов мультимедийных ресурсов системы дистанционного обучения, при этом используемые интеллектуальные алгоритмы позволили масштабировать облако, не снижая при этом объемы задействованных в работе ресурсов. Разработанные модули облачной системы OpenStack показали свою эффективность в качестве балансировщика нагрузки, что позволило предоставить эффективный доступ к пользователям к различным типам данных.

Кроме того, оценка производительности показала уменьшение времени обработки пользовательских запросов за счет увеличения пропускной способности системы при использовании разработанной технологии.

Дальнейшие исследования будут направлены на детальную разработку методики оценки необходимого числа вычислительных узлов, требуемых для обслуживания запросов пользователей.

Представленная работа поддержана грантом Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» № 14.B37.21.1881 и № 14.132.21.1801 и РФФИ № 13-07-00198.

ЛИТЕРАТУРА:

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. Болодурина И.П., Парфёнов Д.И. Исследование методов размещения и организации распределенного доступа к данным облачного хранилища системы дистанционного обучения // Параллельные вычислительные технологии (ПаВТ'2013): труды международной научной конференции (1–5 апреля 2013 г., г. Челябинск). Челябинск: Издательский центр ЮУрГУ, 2013. – С. 301-308 с.
9. Парфёнов Д.И. Сравнение эффективности алгоритмов динамического распределения данных в облачных хранилищах системы дистанционного обучения // Системы управления и информационные технологии, № 4.1(50), 2012. – С. 163-168

10. Парфёнов Д.И. Сравнение эффективности алгоритмов динамического распределения данных в гибридных облачных системах дистанционного обучения // Информационные технологии моделирования и управления, № 6(78), 2012. – С. 491-498
11. OpenStack Open Source Cloud Computing Software. [Электронный ресурс]. - Режим доступа: http://www.openstack.org/