

## КОМПЛЕКСНОЕ РЕШЕНИЕ ЭФФЕКТИВНОГО ДОСТУПА К ДАННЫМ В ГИБРИДНЫХ ОБЛАЧНЫХ СИСТЕМАХ С УЧЕТОМ ОСОБЕННОСТЕЙ МУЛЬТИМЕДИЙНЫХ СЕРВИСОВ

Современное состояние исследований по теме облачных вычислений раскрывает ряд существенных недостатков, связанных с производительностью и эффективностью использования ресурсов высоконагруженными приложениями. В частности для мультимедийных сервисов, активно использующих системы хранения данных для размещения значительных объемов информации, востребованы алгоритмы способные продуктивно обрабатывать запросы на доступ к ресурсам. Однако, используемые методы и алгоритмы в существующих системах виртуализации имеют ряд ограничений масштабируемости, физическим ресурсам, а также времени принятия управляющего решения в критических ситуациях. В связи с этим требуется разработка новых эффективных технологий, методов и алгоритмов для моделирования и оптимизации затрат на виртуализацию ресурсов.

В рамках исследования построена модель организации доступа к данным мультимедийных сервисов с использованием облачной платформы. При этом так же решена задача анализа эффективности работы алгоритмов, методов управления производительностью и оптимизации использования программных и аппаратных ресурсов. Кроме этого решены задачи модельного исследования облачной системы, оценки эффективности использования вычислительных ресурсов с учетом особенностей работы мультимедийных сервисов, а так же разработано комплексное алгоритмическое решение, направленное на повышение скорости обработки поступающих запросов. С этой целью разработана модель потоков данных, циркулирующих в облачных системах, а так же проведены экспериментальные исследования алгоритмов планирования и методов управления потоками данных.

Результаты исследования, основанные на экспериментальных данных, полученных в ходе апробации комплексного алгоритмического решения, позволяют повысить эффективность доступа к данным в гибридных облачных системах с учетом особенностей мультимедийных сервисов. Построенных модели доступа к данным мультимедийных сервисов позволяют расширить представление о внутренних особенностях работы облачных систем, а также описать механизмы взаимодействия основных компонентов, влияющих на работоспособность системы в периоды пиковой нагрузки. Полученный эффект от внедрения разработанных алгоритмов, позволит снизить затраты на аренду облачных ресурсов, за счет оптимизации их использования.

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

На сегодняшний день проводится достаточно много исследований по анализу эффективности и оптимизации сервисов на основе гибридных облачных систем, а так же связанных с ними сервисами по предоставлению мультимедийного контента. Одним из актуальных направлений является исследование механизмов хранения и передачи потоковых данных (видео, аудио контент) [1], [2].

Традиционно оптимизация использования вычислительных ресурсов осуществляется при помощи процедуры балансировки нагрузки. Для высоконагруженных приложений, работающих с мультимедиа контентом так же применяют масштабируемые распределенные системы хранения данных. Проведенный анализ публикаций по теме исследования показал, что на сегодняшний день нет достаточно эффективных комплексных методов по обеспечению доступа к данным [3], [4].

Рассмотрим концепции балансировки нагрузки на системы хранения данных (СХД). Каждое из устройств СХД не способно хранить все данные, которые могут быть востребованы, так как при высоких нагрузках происходит снижение производительности и увеличивается время отклика. В промышленных СХД применяют избыточную запись, но они не поддерживают динамического планирования нагрузки. В облачных системах используется динамический принцип масштабирования дискового пространства [5], [6]. Это позволяет распределять нагрузку, но не снимает проблемы управления потоками данных [7]. Как отмечает Петров Д.Л. в своем исследовании масштабирование и переконфигурация неразрывно связаны с алгоритмами миграции данных [12], [13]. При этом выполнение миграции данных не должно приводить к снижению качества обслуживания.

Существующие алгоритмы не учитывают особенности реконфигурации хранилища и размещения данных [8], [11]. Кроме того, сам алгоритм миграции не учитывает распределенное дублирование данных по устройствам, что значительно снижает отказоустойчивость системы [9], [10].

Кроме перечисленных проблем облачные системы не учитывают особенности приложений, запускаемых на их платформах [14],[15]. Это приводит к увеличению потребляемых ресурсов. В рамках исследования определены ключевые параметров, влияющие на работу каждого из ресурсов, задействованных при построении распределенных гибридных облачных системы и оптимизация их потребления. Это определяет новизну настоящей работы. Рассмотрим ключевые модели комплексной оптимизации облачных систем.

**Модель потоков данных в облачных системах**

Потоки передаваемых данных в облачных системах обрабатываются в несколько этапов. При этом на каждой фазе может применяться несколько элементов, обеспечивающих балансировку и распределение нагрузки между элементами облачной системы. Представим облачную систему в виде кортежа

$$IS = \{S_1^0, \dots, S_l^0, S_1^1, \dots, S_n^1, S_1^2, \dots, S_m^2, S_1^3, \dots, S_k^3, S_1^4, \dots, S_p^4\}, (1)$$

где  $S_i$  - элемент, выполняющий обработку данных, на основе поступающего потока запросов пользователей,  $i=1..r$  ( $r$  - общее число элементов в информационной системе).

$m \in N, n \in N, k \in N$  - число элементов, входящих в состав системы на соответствующих фазах  $f$ ;

$l \in N, p \in N$  - число элементов на входе и выходе информационной облачной системы.

Заявки формируются в потоки данных, обладающие индивидуальными характеристиками. Обозначим все входящие потоки элемента  $S_i^j$  как  $X_i^j$ , а выходящие  $Y_i^j$ , где  $i$  - номер элемента на  $j$ -й фазе обслуживания. При этом траектория заявки между фазами изменяется динамически. Для эффективного обслуживания запросов, образующих потоки данных облачной системе, должно существовать и выполняться

однозначное отображение вида  $R^j : X^j \rightarrow Y^j$ . Тогда сами потоки данных в облачной системе можно представить в виде:

$$Y_i^j = R^j(X_i^j) = \begin{cases} R(X_i^j), & X_i^j \in X^j \\ P_y^j(X^{j*}), & X^{j*} \in \bigcup_{S_i^j} X_i^{j,0} \cup \left( \bigcup_{S_i^{j-1}} X_i^{j,j-1} \right) \\ Q_x^j(Y^{j*}), & Y^{j*} \in \bigcup_{S_i^j} Y_i^{j,0} \cup \left( \bigcup_{S_i^{j+1}} Y_i^{j,j+1} \right) \end{cases} (2)$$

где  $Q_x^j(Y^{j*})$  характеризует входящие, а  $Q_y^j(Y^{j*})$  выходящие пересекающиеся потоки направленные фазу  $j+1$ ;

$P_x^j(X^{j*})$  характеризует входящие, а  $P_y^j(X^{j*})$  выходящие пересекающиеся потоки, поступающие из фазы  $j-1$ .

Для повышения надежности и качества предоставляемых мультимедийных сервисов в проведенном исследовании использованы методы распределения нагрузки, на основе прогнозирования запросов пользователей с применением целевой функции вида [2]:

$$\sum_{i=1}^M \sum_{j \in (T_j)} \mu_i x_{ij} P_i \rightarrow \max, (3)$$

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 обслуживание в облачной системе.

Это позволило увеличить количество запросов, обрабатываемых в единицу времени и повысить эффективность обработки очередей задач в облачной системе.

**Модель доступа к данным хранилища облачной системы**

Установлено, что единой точкой агрегации трафика выступает система хранения данных (СХД). Для оптимизации ее работы разработана модель доступа к мультимедийным данным. Ключевым отличием мультимедийных данных является неоднородность размещаемой информации

и, как следствие, разные подходы к организации доступа к ней [10]. При этом важным аспектом управления ресурсами системы является грамотная организация процесса размещения и распределение элементов данных по устройствам [2], [5]. Облачные хранилища поддерживают реконфигурацию структуры в зависимости от потребляемых ресурсов [11]. Это необходимо учитывать при размещении данных относительно друг друга и клиентов, подключаемых к ним.

В работе облачных СХД так же есть ряд побочных процессов, требующих отдельного внимания. К основным из них относят миграцию и дубликацию данных. Для решения указанных проблем нами разработана модель доступа к данным системы хранения в виде потока вида:

$$R = (U, M, Q), \quad (4)$$

где  $U = (u_1, u_2, \dots)$  – множество пользователей;

$M = (m_1, m_2, \dots)$  – множество уникальных элементов данных, размещаемых на устройствах хранения.

Тогда функция распределения элементов данных по устройствам хранения принимает вид:

$$P(t) : M_c \rightarrow D. \quad (5)$$

где  $M_c$  – множество копий данных;

$D$  – множество устройств хранения данных.

Исходя из изложенного выше, запишем требование пользователя к элементам данных:

$$Q(t) : U \rightarrow X \subseteq M_c. \quad (6)$$

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

Тогда хранилище данных можно записать в виде кортежа

$$S(t) = (M_c(t), D(t), P(t), L(t), C, R(t), G), \quad (7)$$

где  $D(t) = (d_1, d_2, \dots)$  – множество устройств хранения;

$L(t) = (l_1, l_2, \dots)$  – 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;

$G \in N$  – натуральный коэффициент, характеризующий географический приоритет использования хранилища.

Введем целевую функцию вида:

$$\begin{aligned} \sum_{i=1}^N P_i(t) &\rightarrow \min \\ \sum_{i=1}^N L_i P_i(t) R_i(t) &\rightarrow \max \end{aligned} \quad (8)$$

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

Это позволит минимизировать количество одновременно используемых устройств хранения в рамках одного масштабируемого хранилища и максимизировать обработку запросов пользователей в единицу времени.

#### Алгоритмы комплексной оптимизации доступа к данным облачных систем

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. 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 [5], [6].

Эффективность работы алгоритма доказана экспериментальными исследованиями. Дополнительный прирост производительности облака и решаемых в нем задач на 5–9% по сравнению со стандартными средствами управления хранилищем данных в OpenStack.

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. Для оптимизации данного процесса нами используя возможности облачной системы OpenStack, разработан алгоритм формирующий план миграции данных, а также модуль, осуществляющий распределенную обработку созданных в плане вычислительных задач.

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

1. Текущая загрузка 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  $DM_j$ , где  $j=1..N$  ( $N$  – количество параллельно выполняемых операций в хранилище). Выбор и объединение операций в каждом множестве определяется, как, связностью устройств, участвующих в текущей опе-

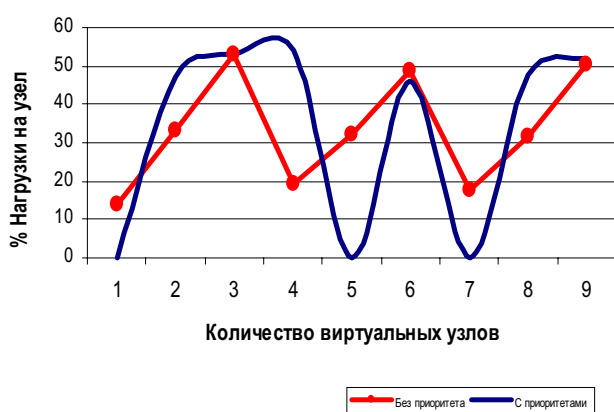


Рисунок 1. График эффективности использования вычислительных ресурсов облачной системы

рации, 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 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, применяемых в облачных системах нами получен суммарный прирост производительности облака на 15–19% относительно стандартных средств применяемых в OpenStack

## Результаты

В рамках экспериментального исследования в облачной системе OpenStack проведено сравнение реального потребления выделенных виртуальных ресурсов проводилось по следующим критериям оценки эффективности: время отклика сервера и количество выделенных виртуальных серверов для каждой из подсистем. На рисунке 1 представлен график, показывающий эффективность работы системы управления вычислительными ресурсами облачной системы.

Представленные на графике результаты экспериментального исследования загрузки вычислительных узлов показывают что, применение разработанных в рамках исследования алгоритмов позволяют на 30% повысить эффективность использования ресурсов облачной системы.

Комплексное моделирование работы облачной системы проводилось учетом особен-

ностей компонентов мультимедийных ресурсов, 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, это позволило предоставить эф-

фективный доступ к пользователям к различным типам данных.

Кроме того, оценка производительности показала уменьшение времени обработки пользовательских запросов, а так же задержек за счет увеличения пропускной способности системы при использовании разработанной технологий.

26.11.2015

**Работа выполнена при поддержке Российского фонда фундаментальных исследований  
проект № 13-07-00198**

**Список литературы:**

1. Бойченко И.В., Коротников 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.
2. 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.
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 научно-технической конференции «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. // Материалы 6-й научно-технической конференции «Новые информационной технологии в ЦБП и энергетике». / Петрозаводск, 2004. - С. 87-90.
5. 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
6. 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
7. Петров Д.Л. Динамическая модель масштабируемого облачного хранилища данных // Известия ЛЭТИ. 2010. - №4 - С. 17-21.
8. Петров Д.Л. Оптимальный алгоритм миграции данных в масштабируемых облачных хранилищах // Управление большими системами. 2010. - №. 30 - С.180-197.
9. 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.
10. 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.
11. Deelman, E. Pegasus: a framework for mapping complex scientific workflows onto distributed systems // Scientific Programming Journal, 2005, 13 (3), pp. 219–237.
12. Gil, Y. Examining the challenges of scientific workflows // IEEE Computer, 2007, 40 (12) (2007) PP. 24–32.
13. Knyazkov, K.V., Kovalchuk, S.V., Tchurov, T.N., Maryin, S.V., Boukhanovsky, A.V. CLAVIRE: e-Science infrastructure for data-driven computing // Journal of Computational Science 2012, 3, pp. 504–510.
14. Korkhov, V, et al., WS-VLAM: towards a scalable workflow system on the grid // Proceedings of the 2nd Workshop on Workflows in Support of Large-scale Science (WORKS'07), ACM, New York, NY, USA, 2007, pp. 63–68.
15. Kovalchuk, S., Larchenko, A., Boukhanovsky, A. Knowledge-Based Resource Management for Distributed Problem Solving // Knowledge Engineering and Management, Springer AISC 123, pp. 121–128.

Сведения об авторах:

**Болодурина Ирина Павловна**, заведующий кафедрой прикладной математики  
Оренбургского государственного университета, доктор технических наук, профессор  
460018, г. Оренбург, пр-т Победы, 13, ауд. 20608, тел. (3532) 372536, e-mail: prmat@mail.osu.ru

**Парфёнов Денис Игоревич**, начальник отдела программно-технической поддержки  
дистанционного обучения Оренбургского государственного университета, кандидат технических наук  
460018, г. Оренбург, пр-т Победы, 13, ауд. 3319, тел. (3532) 375932, e-mail: fdot\_it@mail.osu.ru