

ИССЛЕДОВАНИЕ ОРГАНИЗАЦИИ ВИРТУАЛЬНЫХ ЦОД НА БАЗЕ ГЕТЕРОГЕННОЙ ОБЛАЧНОЙ ПЛАТФОРМЫ ДЛЯ РАЗМЕЩЕНИЯ ИНТЕРАКТИВНЫХ ОБРАЗОВАТЕЛЬНЫХ СЕРВИСОВ

Болодурина И.П., Парфёнов Д.И.

Оренбургский государственный университет, г. Оренбург

В настоящее время проблемы эффективного использования имеющихся вычислительных ресурсов является весьма актуальной задачей. Прежде всего, это связано с тем, что в мировых корпоративных и частных центрах обработки данных (ЦОД) накоплен колоссальный объем вычислительных мощностей. При этом их востребованность остается весьма низкой по сравнению с объемом ресурсов. Для снижения затрат на содержание такой значительной инфраструктуры в ЦОД применяют методы виртуализации ресурсов и передают в совместное использование часть имеющихся ресурсов. При этом конечные пользователи получают распределенную инфраструктуру, включающую различные гетерогенные вычислительные ресурсы и сервисы для обработки «больших» данных с использованием сети передачи данных.

Современные облачные вычислительные ресурсы имеют высокую структурную сложность (в облачной платформе, как правило, используются различные программно-аппаратные ресурсы). Для описания организации взаимодействия между предметно-ориентированными программными сервисами и распределенными источниками данных в рамках сервис-ориентированной вычислительной инфраструктуры в работах А.В. Бухановского, С.В. Ковальчука[5,6] используется подход, основанный на описании рабочих потоков (Workflow). Этот подход воплощен при разработке облачной платформы CLAVIRE (CLoud Applications VIRtual Environment). Но как и все облачные платформы, CLAVIRE не позволяет использовать возможности самоорганизации при управлении ресурсами для обеспечения оптимальной производительности приложений, в том числе это относится и к хранилищам данных. Как правило, для управления ресурсами систем хранения данных облачной платформы используются простейшие классические алгоритмы планирования вычислительных заданий, представляющие собой либо сочетания алгоритма обратного заполнения Backfill [5] с методами их распределения на устройства First Fit или Best Fit [5], либо применение одного из двух типов алгоритмов планирования – «жадных» или «ленивые». Данные алгоритмы в состоянии обеспечить эффективное размещение данных при загрузенности системы (по объему данных) на уровне 50-60%, что, очевидно, является недостаточным. Основной особенностью существующих алгоритмов является отсутствие учета топологии системы, коммуникационных схем взаимодействия запускаемых проблемно-ориентированных приложений и их классификации по ресурсоемкости и направлению использования. Кроме этого алгоритмами не учитывается структура данных. Данный критерий является весьма критичным при размещении разнородных данных на единой платформе. Это обусловлено тем, что одни данные требуют оперативного доступа и для

них важна скорость передачи, для других критичным является объем занимаемого ими пространства.

В состав ресурсов, предоставляемых облачными платформами входят, различные объекты доступа, при этом взаимодействие между ними имеет приоритетное значение для оперативного удовлетворения потребностей конечных пользователей. Требования к организации предоставления ресурсов меняются постоянно в процессе работы сервисов, развернутых на облачной платформе. При этом важным аспектом является автоматизация управления структурой: создание виртуальных машин, реконфигурация хранилищ данных и размещение объектов. Ведущими учеными Lin W. и Qi D. предложено несколько различных подходов по управлению облачными платформами.

Основным недостатком существующих облачных платформ является слабая организация динамического управления ресурсами, связанная с хранилищем данных [8]. В проведенном нами исследовании разработан подход, основанный на динамическом управлении ресурсами с использованием самоорганизующихся методов. Преимуществом самоорганизации является использование децентрализованных ресурсов. Кроме того самоорганизация управления сокращает сложность системы управления облачной платформой и улучшает масштабируемость и гибкость системы, а так же позволяет более эффективно использовать вычислительные ресурсы.

Ряд зарубежных ученых предпринимали попытки применения принципов самоорганизации для облачных платформ. Так в работах ученых Sheng Di и Cho-Li Wang рассматривается облачная платформа, построенная на принципах самоорганизации. Каждый из элементов облачной платформы определен как автономная единица, выступающая как в роли потребителя, так и поставщика ресурсов, при этом сформированная система является полностью децентрализованной. Управление виртуальными машинами и выделением ресурсов осуществляется с использованием специализированного протокола Content Addressable Network (CAN). Отличительной особенностью протокола от существующих решений является компактность сообщений о запросе ресурсов. Разработанный протокол (CAN) производит только один запрос на каждую задачу, что существенно сокращает объем служебного трафика. Такой подход не только обеспечивает максимальную загрузку ресурсов, используя модель пропорционального распределения ресурсов (PSM), но также обеспечивает адаптивную оптимизацию для повышения эффективности их использования. Кроме того предлагаемый авторами алгоритм осуществляет эффективный поиск ресурсов для каждой задачи, динамически настраивая политики доступа, что снижает конкуренцию среди запрашивающих ресурс виртуальных машин и сервисов. Авторская оценка эффективности работы алгоритма с использованием протокола (CAT) производится на основе сравнения с моделью P2P и составляет увеличение эффективности работы системы на 15-60 процентов.

Основной особенностью применения методов самоорганизации для модели в Cloud Computing является реализация динамической оптимизации вычислительных ресурсов путем внедрения в системы планирования. Так в

Калифорнийском университете в Беркли применяют адаптивные методы составления расписания для распределенных систем [6]. В работах зарубежных ученых отмечается, что программное обеспечение облачных платформ подвержено следующим ограничениям, оказывающим непосредственное влияние на доступность предоставляемых услуг: ограниченная масштабируемость вычислительных ресурсов в рамках ЦОД и пиковая производительность системы хранения [7]. Исследователями рассматриваются подходы к самоорганизации приложений в облачных платформах и управление запуском в комбинированной системе, использующей слой маршрутизации P2P на основе распределенного хранилища данных. В качестве самоорганизации применяются протоколы и алгоритмы выбора узлов управления. При этом используется мультиагентный подход. Каждый из агентов системы поддерживает трехслойную самоорганизацию. Для обеспечения соглашения о качестве обслуживания (SLA) при распределении ресурсов облачной платформы применяется система GridBatch [1,2], направленная на решение масштабных по объему данных задач.

Анализ научно-информационных источников по проблеме исследования показал, что на данный момент отсутствуют эффективные алгоритмические решения по планированию ресурсов облачных платформ, включая виртуальные машины, системы хранения данных, облачные сервисы и проблемно-ориентированные приложения, а так же существующие решения, применяемые в облачных платформах, не используют принципы самоорганизации и недостаточно эффективны в плане масштабируемости и сбалансированности. Это демонстрирует новизну решений проведенного исследования.

Ранее нами рассматривались решения для облачных хранилищ данных, обеспечивающих миграцию данных, а также алгоритмы размещения данных на устройствах [3-4]. Учитывая новую парадигму программно-управляемых хранилищ, предлагаемые концепции требуют доработки в части определения типов размещаемых данных относительно их структуры. В ходе исследования нами проведена классификация данных, размещаемых на устройствах в сетевой среде. На основе матрицы соответствий нами построен алгоритм, позволяющий определять структурированность получаемых данных для определения конечного способа размещения (схд / sql / nosql), вида физического устройства (HDD или SSD), так же непосредственный выбор самого оптимального устройства. Таким образом, размещаемые в хранилище данные можно представить в виде структуры $Data = (TypeS, TypeD, RDisk)$, где $TypeS$ - способ размещения, $TypeD$ - вид физического устройства, $RDisk$ - физическое устройство хранения.

В ходе исследования установлено, что размещение данных на физических устройствах имеет ряд ограничений, влияющих как на производительность операции по чтению/записи данных, так и на процесс оптимизации расположения данных на устройствах. Для нейтрализации установленного ограничения предлагается применять программно-управляемые хранилища данных. Данный вид хранилища можно представить в виде следующей структуры: $SoftStg = (Vm, Lan, Stype, Dtype, RDisk(t), Vdisk)$, где Vm - виртуальная

машина или сетевой контейнер, Lan – скорость доступа сетевого интерфейса, $Stype$ – поддерживаемый способ размещения данных, $Dtype$ вид физического устройства, на которых будет размещаться виртуальная машина; $RDsik(t)$ конкретное физическое устройство, содержащее виртуальную машину в данный момент времени t ; $Vdisk$ – общий объем хранилища данных. Преимуществом такого размещения данных является возможность миграции хранилища между физическими устройствами. При этом связи и доступность данных со стороны облачной платформы остаются непрерывными, так как соединение осуществляется с виртуальным хранилищем, а не физическим устройством.

Концепция программно-управляемых хранилищ строится так же на принципах самоорганизации на базе абстракций. В настоящем исследовании использована модель вычислительных ресурсов облачной системы, разработанная ранее [10,11]. В дополнении к существующим объектам облачной инфраструктуры в модель введено понятие агента и узла управления. Агент представляет собой вычислительный узел облачной системы, который может выступать как в роли вычислительного узла *Snode*, системы хранения данных *Sstg*, сетевого хранилища данных *Snas*. При этом в любой момент времени агент может стать узлом управления. Это возможно благодаря кластеризации вычислительных узлов.

В основу самоорганизации программно-управляемого хранилища данных положена адаптивная модель динамической реконфигурации адаптации к изменениям ресурсов. Это помогает оптимизировать организационную структуру облачной платформы, а именно алгоритмы поиска оптимальных узлов управления, а так же выделение групп управления. Предлагаемая нами модель управления состоит из двух частей: узлов и ресурсов. При формировании программно-управляемых хранилищ на каждом виртуальном вычислительном узле запускается программное обеспечение, отвечающее за обмен техническими данными об устройствах. При этом обмен осуществляется в пределах группы узлов, осуществляющих хранение данных с одним способом хранения. При этом среди группы узлов выбирается наименее загруженный узел, выступающий в роли узла управления. Такой подход позволяет снизить риск деградации управляющего узла во время работы. Тем не менее, если происходит потеря связи с управляющим узлом, у оставшихся в группе виртуальных машин всегда присутствуют данные друг о друге, что позволяет произвести автоматически выбор нового узла управления и делегировать ему полномочия, что так же снижает риски отказов системы управления. Помимо задачи организации обмена и управления хранилищами своей группы, управляющий узел осуществляет взаимодействия с узлами управления других групп для поддержания актуальной информации о состоянии всей системы в целом. Таким образом, вся система программно-управляемых хранилищ построена по иерархическому сетевому принципу, включающему три базовых уровня: уровень оперативного доступа, уровень управления в группе и уровень обмена данными на уровне всей системы в целом.

На базе описанной концепции программно-управляемых хранилищ нами реализован алгоритм размещения данных в программно-управляемых хранилищах. Конкурентным преимуществом разработанного алгоритма управления программно-управляемым хранилищем данных по сравнению с имеющимися аналогами является эвристический анализ новых типов данных в процессе загрузки файлов в систему облачной платформы. При этом благодаря виртуализации хранилища, осуществляется прозрачное для клиента зеркальное дублирование данных, на несколько устройств хранения, что обеспечивает увеличение скорости размещения данных в плане обеспечения целостности и резервирования.

Формирование программно-управляемых самоорганизующихся хранилищ данных на базе виртуальных машин и контейнеров позволяет не только снизить риски, связанные с потерей или недоступностью данных, но и обеспечивает интеллектуальный анализ востребованности данных, на базе которого формируются карты размещения виртуальных машин и контейнеров. В основу алгоритма размещения данных в программно-управляемых хранилищах положена модель, позволяющая описать структуру и связи виртуальных устройств, машин и контейнеров с данными. Модель основана на мультиагентном подходе в организации устройств хранения. Агенты осуществляют сбор данных о состоянии системы. Полученная информация подвергается анализу с применением алгоритмов машинного обучения (Data Mining). На выходе проведенного анализа получается карта расположения устройств внутри самой облачной платформы с привязкой к физическим устройствам, а так же формируется карта востребованности самих данных. Путем анализа двух карт и эвристического алгоритма прогнозирования система управления облаком принимает решение о реконфигурации или перемещении устройств виртуального хранилища, а так же о ротации и перераспределении данных между различными узлами системы. При этом карты размещения являются динамическими объектами, формируемыми не только по мере возникновения событий загрузки или чтения данных, но с заданным интервалом времени.

Кроме обозначенной проблемы, разработанный алгоритм размещения данных в программно-управляемых хранилищах применяется для повышения производительности работы компонентов облачной системы. За счет эффективного перераспределения потоков данных между запущенными экземплярами виртуальных машин и контейнеров, обеспечивается не только должное качество обслуживания, но и компактное расположение устройств [11]. Для решения данной проблемы в качестве одного из элементов в основе разработанного в рамках исследования алгоритма размещения данных в программно-управляемых хранилищах использован агрессивный вариант алгоритма Backfill (алгоритма обратного заполнения), применяемого для оптимизации выполнения задач в Грид [9]. Задача повышения эффективности использования имеющихся устройств хранения на основе данных получаемых от агентов и узлов управления облачной системы путем динамического управления ресурсами в условиях ограничения потребления их вычислительных мощностей является актуальной для облачных систем. Это в

первую очередь связано с экономическими показателями использования облачных платформ.

Помимо использования элементов алгоритма Backfill для повышения эффективности работы облачной системы произведен ряд оптимизационных улучшений, касающихся непосредственно механизма доступа к данным. При работе с сервисами, размещенными в облачной системе, не исключена ситуация, при которой для обслуживания запроса пользователя могут быть задействованы сразу несколько хранилищ данных с различными характеристиками доступа. При работе с такими данными облачной системе необходимо осуществлять подготовку доступа для оптимизации времени чтения. Для этого разработанный алгоритм размещения данных в программно-управляемых хранилищах в ходе работы строит ряд внутренних правил, тем самым подстраивая под поток запросов пользователей каждый экземпляр хранилища. В результате, план выполнения запросов с одинаковой интенсивностью в разные моменты времени может быть распределен по-разному. Перестроение правил происходит в соответствии с востребованностью ресурсов, что позволяет эффективно управлять распределением и динамической балансировкой нагрузки.

Для оценки эффективности алгоритма размещения данных в программно-управляемых хранилищах, построенного с учетом представленных модели программно-управляемого хранилища данных, нами проведено исследование работы облачной системы построенной на базе Openstack с различными параметрами. При этом в качестве эталонных данных для сравнения в эксперименте использовались типовые алгоритмы, применяемые в облачных системах, а так же традиционные системы хранения. Для экспериментального исследования создан прототип облачной среды, включающей в себя основные узлы, а так же программные модули для разработанных алгоритмов, модифицирующие обработку запросов пользователей к данным в программно-управляемом хранилище. Схема экспериментального стенда на базе облачной системы Openstack представлена на рисунке 2.

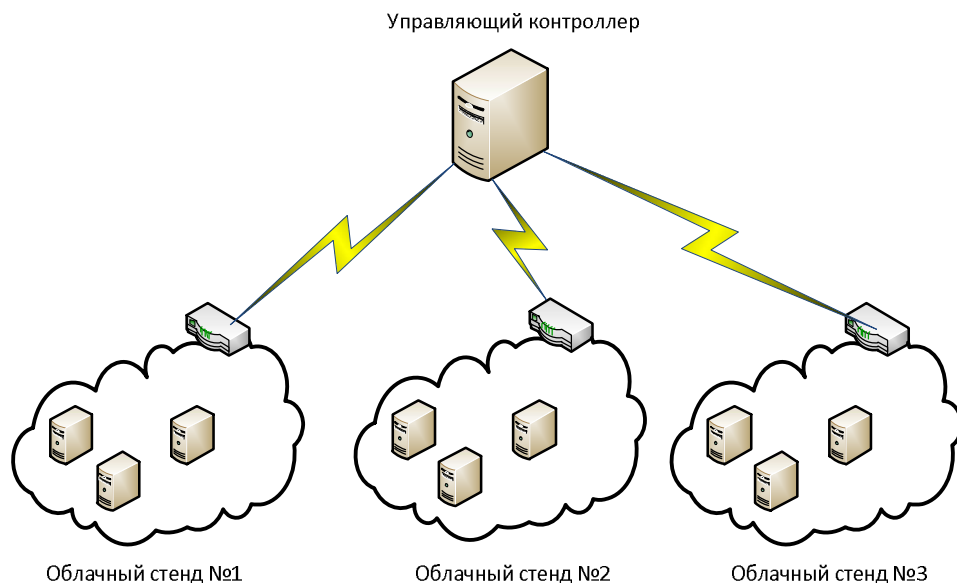


Рис.2. Схема экспериментального стенда на базе облачной системы Openstack

В облачной системе OpenStack реализован модуль, применяющий разработанный алгоритм размещения данных в программно-управляемых хранилищах для рационального использования вычислительных ресурсов облачной системы и эффективного распределения виртуальных машин по физическим узлам, а так же связанных с ними данных. В ходе эксперимента для анализа данных создан поток запросов, аналогичный реальному трафику облачной инфраструктуры, основанный на данных лог-записей доступа к определенным видам ресурсов с классификацией по типам данных и структуре запроса. Ретроспектива воспроизводимых запросов составила 3 года, при этом для нагрузочного эксперимента применялись усредненные данные. Полученные данные распределены на пул виртуальных машин по следующим критериям: тип клиента осуществившего обращение к данным, тип сервиса востребованного при подключении. При этом количество одновременных запросов, поступивших в систему, составило 100000.

Все сформированные запросы воспроизводились последовательно на трех экспериментальных площадках. Данное ограничение введено в связи необходимостью сопоставления результатов с физическими системами хранения данных, не способных к реконфигурации. Основным отличием экспериментальных площадок является использование твердотельных накопителей.

Помимо площадок для анализа эффективности сформировано 3 группы экспериментов, направленных на интенсивное выполнение операции по чтению (эксперимент 1), записи (эксперимент 2) и одновременных операциях чтения и записи данных (эксперимент 3). Время эксперимента составило один час, что соответствует наиболее длительному периоду времени пиковой нагрузки системы, зафиксированному в реальном трафике. По результатам проведенных экспериментов можно сделать вывод о снижении на 20-25% количества отказов в обслуживании при размещении данных в программно-управляемых хранилищах данных.

Кроме этого в рамках экспериментального исследования проведена оценка количества используемых устройств хранения для экспериментальных площадок 2 и 3, так как для физических СХД не поддерживают масштабирование ресурсов в реальном времени. За счет оптимального выделения ресурсов на каждом вычислительном узле гарантировано обеспечение совместной работы всех запущенных экземпляров приложений, что удовлетворяет требованиям потенциальных пользователей. При этом благодаря работе алгоритма размещения данных в программно-управляемых хранилищах возможно высвобождение от 20 до 30% выделенных ресурсов вычислительных узлов. Таким образом, предложенный алгоритм может использоваться для произвольных архитектур вычислительных систем, в том числе с неоднородной конфигурацией физических узлов и виртуальных машин.

Таким образом, оценивая общий результат работы алгоритма размещения данных в программно-управляемых хранилищах можно получить прирост производительности от 20 до 25% по сравнению с физическими системами хранения и виртуальными машинами, используемыми в качестве СХД, что

является весьма эффективным при большой интенсивности запросов. Кроме того, сокращение числа выделенных виртуальных ресурсов позволяет более эффективно масштабировать облачную систему, и обеспечить запас надежности при резком увеличении интенсивности использования выделенных приложений.

Исследование выполнено при финансовой поддержке РФФИ в рамках научных проектов № 16-47-560973, 16-47-560390, 16-37-60086.

Список литературы

1. *The Fourth Paradigm* //T. Hey, S. Tansley, K. Tolle (Eds.), *Data-Intensive Scientific Discovery*, Microsoft, 2009, p. 252.
2. Gil, Y. *Examining the challenges of scientific workflows* //IEEE Computer, 2007,40 (12) (2007) PP. 24–32.
3. Deelman, E. *Pegasus: a framework for mapping complex scientific workflows onto distributed systems* //Scientific Programming Journal, 2005, 13 (3), pp. 219–237.
4. 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.
5. 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.
6. Kovalchuk, S., Larchenko, A., Boukhanovsky, A. *Knowledge-Based Resource Management for Distributed Problem Solving* // *Knowledge Engineering and Management, Springer AISC* 123, pp. 121–128.
7. Марьин С.В., Ковальчук С.В., Бухановский А.В., Васильев В.Н. *HPC-NASIS: распределенная проблемно-ориентированная оболочка для вычислений в области квантовой химии* [Электронный ресурс]. – Режим доступа: <http://grid2010.jinr.ru/files/pdf/hpc-nasis.pdf>
8. Aida, K. *Job Scheduling Scheme for Pure Space Sharing among Rigid Jobs* / K. Aida, H. Kasahara, S. Narita // *Lecture Notes In Computer Science, Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing, Vol. 1459*. – Лондон: Springer-Verlag, 1998. PP. 98-121
9. Гергель, В.П. *Исследование алгоритмов планирования параллельных задач для кластерных вычислительных систем с помощью симулятора* / Гергель В.П., Полежаев П.Н. // Журнал «Вестник ННГУ», № 5, 2010. С. 201–208. Ерёмин И.И. *Фейеровские методы для задач выпуклой и линейной оптимизации*. Челябинск: Изд-во ЮУрГУ, 2009. 200 с.
10. Болодурина И.П., Парфёнов Д. И. *Управление потоками запросов при доступе к широкополосным мультимедийным образовательным ресурсам системы дистанционного обучения* // *Проблемы управления*. – 2013. – № 6. - С. 24-30.
11. Болодурина И.П., Парфёнов Д. И. *Алгоритмы комплексной оптимизации потребления вычислительных ресурсов в облачной системе* // *Вестник*

Оренбургского государственного университета. – 2013. № 9 (158). – С. 177-184.