УДК 519.687

DOI: 10.15827/0236-235X.116.015-022

Дата подачи статьи: 16.06.16

2016. Т. 29. № 4. С. 15–22

МОДЕЛИРОВАНИЕ РАЗМЕЩЕНИЯ СЕРВИС-ОРИЕНТИРОВАННЫХ ПРИЛОЖЕНИЙ В ПРОГРАММНО-УПРАВЛЯЕМОЙ ИНФРАСТРУКТУРЕ ВИРТУАЛЬНОГО ЦЕНТРА ОБРАБОТКИ ДАННЫХ

*И.П. Болодурина, д.т.н., профессор, prmat@mail.osu.ru;**Д.И. Парфенов, к.т.н., fdot_it@mail.osu.ru**(Оренбургский государственный университет, просп. Победы, 13, г. Оренбург, 460018, Россия);**В.Н. Решетников, д.ф.-м.н., профессор, rvn_@mail.ru**(Центр визуализации и спутниковых информационных технологий НИИСИ РАН,
Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

Развитие информационных технологий в последнее время обуславливает использование распределенных сервис-ориентированных бизнес-приложений. При этом актуальны исследование механизмов управления размещением приложений в облачной среде и эффективное выделение соответствующих ресурсов для их работы в рамках существующих ограничений.

Целью данного исследования являются моделирование структуры и механизмов взаимодействия типовых облачных сервис-ориентированных приложений, а также определение подходов к оптимизации их размещения в системе хранения. Авторами разработана обобщенная модель сервис-ориентированного облачного приложения, включающая в себя три базовых слоя: модель приложения, модель сервиса и модель ресурса. В качестве ресурса в модели выступает некий объект, описывающий поведение и характеристики отдельных элементов инфраструктуры в зависимости от текущего состояния и параметров программно-управляемой инфраструктуры виртуального центра обработки данных. Отличительная особенность модели в том, что ресурс является универсальным, а это позволяет рассматривать его как с точки зрения пользователя, для которого он представляет собой закрытую систему, так и с точки зрения самой программно-управляемой инфраструктуры виртуального центра обработки данных, для которого он является открытой системой. Новизна модели заключается в одновременном описании размещения данных, связанных с ними приложений, а также состояния виртуальной среды, учитывающей сетевую топологию. Для детализации модели ресурса программно-управляемой инфраструктуры виртуального центра обработки данных в качестве отдельного элемента разработана модель виртуального самоорганизующегося хранилища данных, позволяющая определить их параметры и связи приложений с виртуальными и физическими узлами хранилища данных.

Ключевые слова: *облачные приложения, облачные сервисы, вычислительные ресурсы, виртуальные центры обработки данных, программно-управляемые устройства хранения, облачные вычисления, машинное обучение, OpenStack.*

В последние годы технология облачных вычислений стала популярной для предоставления доступа к приложениям и сервисам с целью обеспечения работы бизнес-услуг [1]. Существуют три основных подхода к предоставлению услуг в сфере облачных вычислений: инфраструктура как услуга (IaaS), платформа как услуга (PaaS) и ПО как услуга (SaaS). Применение этих подходов при развертывании облачных вычислительных платформ имеет множество преимуществ, таких как надежность и качество обслуживания [2]. В то же время существует ряд ограничений со стороны как пользователей, так и поставщиков облачных услуг. Для потребителя облачные ресурсы представляются бесконечными с точки зрения масштабируемости, однако, если рассмотреть экономический аспект их потребления, видим, что возможности их существенно сужаются. Со стороны провайдеров облачных услуг, предоставляющих доступ к ресурсам, набор сервисов и вычислительных мощностей ограничен. Для максимизации экономической составляющей за счет увеличения числа пользователей поставщики вынуждены применять политики, позволяющие эластично использовать выделяемые ресурсы, минимизируя при этом операционные затраты.

Актуальной проблемой является управление ресурсами и облачными приложениями в современных виртуальных и физических *центрах обработки данных* (ЦОД), поскольку оно напрямую влияет на эксплуатационные расходы. Крупные корпорации отрасли информационных технологий, такие как Amazon, Google, Salesforce, IBM, Microsoft, Oracle и др., в течение последних нескольких лет разрабатывают обновленные подходы к управлению ресурсами и объектами в ЦОД, используемых для размещения решений для облачных приложений [3, 4]. Основной тенденцией в данной области развития современных информационных технологий является оптимизация использования ресурсов ЦОД.

В проведенных ранее исследованиях авторами статьи были разработаны подходы к оптимизации хранения данных облачных приложений, а также к повышению эффективности доступа к ресурсам облачной среды [5]. Однако это не решает проблему размещения экземпляров самих приложений в облачной среде. Кроме этого, проведенный обзор исследований в данной области показал, что проблема оптимизации выбора ресурсов для конкретных типов облачных приложений недостаточно изучена [6–9].

Для решения обозначенных проблем авторы предлагают подход, основанный на управлении размещением облачных приложений и выделением соответствующих ресурсов для их работы.

Для понимания принципов работы облачных приложений необходимо определить их место в инфраструктуре ЦОД. На сегодняшний день строительство и эксплуатация выделенных физических ЦОД экономически нецелесообразны. Поэтому все чаще имеется в виду виртуальный ЦОД. Такой подход позволяет организовывать собственную инфраструктуру на базе арендуемых вычислительных мощностей. Виртуальный ЦОД с точки зрения облачного приложения является изменяющимся объектом. При этом его состояние в любой произвольный момент времени t можно описать в виде ориентированного графа: $VirtDC=(Node(t), Connect(t))$, где вершины $Node(t)=\{Node_1, Node_2, \dots, Node_n\}$ представляют собой активные элементы облачной инфраструктуры виртуального ЦОД (вычислительные узлы, системы хранения данных и др.), используемые активными экземплярами приложений $CloudAppl(t) = \{CloudAppl_1, CloudAppl_2, \dots, CloudAppl_c\}$, запущенные на виртуальных ресурсах; ветви – $Connect(t) = \{Connect_1, Connect_2, \dots, Connect_v\}$ – активные подключения пользователей к облачным приложениям.

Особенность работы облачных приложений в том, что слой виртуализации скрывает от конечных пользователей количество и расположение запущенных экземпляров сервисов или приложений, с которыми они работают. Чаще всего им доступны только адрес агрегирующего узла и имя приложения. Облачная система самостоятельно выбирает оптимальную виртуальную машину для выполнения запроса. Таким образом, с точки зрения пользователя, облачное приложение является черным ящиком, который осуществляет обработку поступающих запросов.

Прежде чем говорить о способах размещения облачных приложений в виртуальном ЦОД, необходимо определить их структуру, основные параметры и ключевые характеристики работы, влияющие на эффективность использования этих приложений. Для этого была разработана обобщенная модель сервис-ориентированного облачного приложения.

Обобщенная модель сервис-ориентированного облачного приложения представляет собой многослойную структуру, описываемую в виде графов, характеризующих связи отдельных элементов. Модель может быть представлена в виде трех базовых срезов, детализирующих связи отдельных облачных объектов инфраструктуры виртуального ЦОД: приложений, связанных с ними сервисов, а также предоставляемых им для работы ресурсов.

Сервис-ориентированное облачное приложение – это взвешенный ориентированный ациклический

граф зависимостей по данным, в котором вершинами являются задачи, направленные на получение или обработку информации из источников в соответствии с запросом пользователя, а дугами – зависимости приложений от источников данных между соответствующими вершинами: $CloudAppl=(G, V)$, где G – вершины графа задачи, направленные на получение или обработку информации по запросу пользователя; V – дуги зависимости приложений от источников данных.

Каждая вершина (G) характеризуется предъявляемыми ресурсными требованиями, количеством запускаемых экземпляров, оценкой времени выполнения запросов пользователей, коммуникационными шаблонами передачи данных между источниками и вычислительными узлами: $G=\{Res, NAppl, Utime, SchemeTask\}$, где Res – требования приложения к ресурсам; $NAppl$ – требования к количеству запускаемых экземпляров приложений; $Utime$ – оценка времени выполнения запросов пользователей; $SchemeTask$ – коммуникационные шаблоны передачи данных приложений.

Каждая дуга (V), соединяющая приложение с требуемым источником данных, характеризуется типом передаваемых данных, методом получения доступа к источнику информации, физическим типом объекта доступа, объемом трафика, требованиями к качеству обслуживания при передаче данных (QoS): $V=\{Tdata, Mdata, Fdata, Vdata, Qdata\}$, где $Tdata$ – тип передаваемых данных; $Mdata$ – метод получения доступа к источнику информации (REST, JSON и др.); $Fdata$ – тип объекта доступа (доступ к файлу в СХД, к локальному файлу, к распределенной БД, к сервису данных и пр.); $Vdata$ – объем трафика, оцениваемый на основе информации о данных (Мб); $Qdata$ – требования к QoS.

Оригинальность модели заключается в том, что для каждого приложения производится консолидированная оценка эффективности работы с источниками данных, что позволяет прогнозировать работу всей системы в целом.

Как отмечалось ранее, одним из ключевых срезов обобщенной модели сервис-ориентированного облачного приложения является облачный сервис. Он служит автономным источником данных для приложения, выполняющего роль консолидированного обработчика информации. Как правило, облачный сервис является узкоспециализированным и предназначен для выполнения ограниченного набора функций. Преимущество подключения облачного приложения к сервису, в отличие от прямого доступа к данным, в изолированной обработке информации, что позволяет сократить время исполнения запроса пользователя. Облачный сервис можно описать как закрытую систему в виде ориентированного графа зависимостей функционала от поступающего набора данных. Для описания доступа требуются адрес агрегирующего вы-

числительного узла и имя сервиса, а также требуемый на выходе формат данных. Агрегатор сервиса сам выбирает оптимальную для запроса виртуальную машину, на которой он и исполняется. Также все его приложения распределены между виртуальными машинами или физическими серверами предустановленного набора, их новые экземпляры масштабируются динамически в зависимости от количества поступающих запросов на выполнение функции конкретного сервиса со стороны облачного приложения, конечного пользователя или других облачных сервисов.

$CloudServ = \{AgrIP, NameServ, Format\}$, где $AgrIP$ – адрес агрегирующего вычислительного узла сервиса; $NameServ$ – имя сервиса; $Format$ – требуемый на выходе формат данных.

Для описания размещения облачных сервисов и приложений в инфраструктуре ЦОД была также реализована модель облачного ресурса. Под облачным ресурсом будем понимать объект ЦОД, описывающий поведение и характеристики отдельных элементов инфраструктуры в зависимости от его текущего состояния и параметров. В качестве объектов могут выступать дисковые массивы, в том числе отдельные устройства хранения, виртуальные машины, программно-конфигурируемые хранилища данных, а также СУБД различных видов (SQL/NoSQL) и др. Кроме этого, каждый облачный сервис или приложение предъявляет требования к количеству ядер, размерам оперативной и дисковой памяти, наличию специальных библиотек на физических или виртуальных узлах, используемых для запуска их среды.

$CloudRes = \{TRes, Param, State, Core, Rmem, Hmem, Lib\}$, где $TRes$ – тип ресурса; $Param$ – параметры доступа; $State$ – состояние ресурса; $Core$ – количество вычислительных ядер; $Rmem$ – размер оперативной памяти; $Hmem$ – размер дисковой памяти; Lib – требования к наличию специальных библиотек.

Отличительная особенность модели в универсальности ресурса, что позволяет рассматривать его как с точки зрения пользователя, для которого он представляет собой закрытую систему, так и с точки зрения самой программно-управляемой инфраструктуры виртуального ЦОД, для которого он является открытой системой.

Новизна модели заключается в одновременном описании размещения данных, связанных с ними приложений, а также состояния виртуальной среды, учитывающей сетевую топологию.

Для детализации модели ресурса программно-управляемой инфраструктуры виртуального ЦОД в качестве отдельного элемента разработана модель виртуального самоорганизующегося хранилища данных (SDS), представляемая также в виде ориентированного мультиграфа, включающего в себя в качестве вершин элементы виртуального ЦОД, отвечающие за размещение данных приложений

(виртуальные дисковые массивы, СУБД и т.д.). При этом в качестве дуг выступают связи данных с облачными приложениями и с сервисами. В качестве весов использована консолидированная оценка, состоящая из характеристики востребованности данных, а также загруженности конкретных устройств как на физическом, так и на виртуальном уровне.

$Stg = (StgNode, Data)$, где $StgNode$ – элементы виртуального ЦОД, отвечающие за размещение данных приложений; $Data$ – связи данных с облачными приложениями и сервисами.

Каждый элемент виртуального самоорганизующегося хранилища данных обладает индивидуальными характеристиками:

$$Stg_{ki} = (MaxV_{ki}, P_{ki}^{stg}, Vol_{ki}(t), \bar{R}_{ki}(t), \bar{W}_{ki}(t), s_{ki}^{stg}(t)),$$

где $MaxV_{ki} \in N$ – максимальная емкость для хранения (Mb); $P_{ki}^{stg} = \{p_{kij}^{stg}\}_j$ – множество сетевых портов; $Vol_{ki}(t) \in N \cup \{0\}$ – доступный объем для хранения в Mb; $\bar{R}_{ki}(t)$ и $\bar{W}_{ki}(t)$ – скорость чтения и записи; $s_{ki}^{stg}(t) \in \{"online", "offline"\}$ – состояние.

Система хранилищ данных для облачных приложений напоминает слоеный пирог и работает на принципах самоорганизации ресурсов. Адаптивная модель динамической реконфигурации ресурсов является основой самоорганизации управления хранилищами данных. Кроме того, принципы самоорганизации ресурсов применяются при оптимизации структуры управления самой облачной платформой на базе адаптивного выбора управляющих узлов. В разработанной авторами модели управления ресурсами облачной системы условно можно выделить два ключевых компонента: виртуальные узлы и вычислительные ресурсы. Управление ресурсами в виртуальных самоорганизующихся хранилищах осуществляется посредством специализированного ПО (агенты управления), осуществляющего синхронизацию и сбор технической информации о вычислительных узлах, выделенных для размещения данных. Для снижения объемов потока трафика, формируемого агентами управления, обмен данными происходит между узлами с одинаковым способом хранения информации. При формировании больших групп для организации виртуальных самоорганизующихся хранилищ данных возможно выделение подгрупп и назначение дополнительных управляющих узлов. При назначении управляющего агента предпочтение отдается наименее загруженным узлам. Децентрализованный подход, основанный на принципах самоорганизации управления, позволяет исключить фактор потери связи с управляющим узлом во время работы. Каждый управляющий агент на виртуальном вычислительном узле системы хранения в любой момент времени имеет данные о других узлах, входящих в его группу или подгруппу, а также

информацию об управляющих узлах других групп, что позволяет провести выборы и делегировать полномочия новому управляющему узлу. Это обеспечивает надежность разработанной системы управления виртуальными хранилищами данных. В описанной концепции управления виртуальными самоорганизующимися хранилищами данных можно выделить три слоя управления, подчиняющихся друг другу по иерархическому принципу: первый (локальный) – управление на уровне оперативного доступа; второй (групповой) – управление на уровне группы ресурсов; третий (глобальный) – управление на уровне обмена данными во всей облачной системе. На базе описанной модели управления виртуальными самоорганизующимися хранилищами данных реализован алгоритм размещения данных.

Новизна обобщенной модели сервис-ориентированного облачного приложения заключается в одновременном описании подзадач и источников данных разного типа с указанием направления и способа передачи данных, а также требуемых для этого ресурсов.

Обобщенная модель облачного сервис-ориентированного приложения положена в основу при разработке алгоритмов планирования облачных сервисов и приложений поверх виртуальных машин и физических серверов инфраструктуры виртуального ЦОД, а также в алгоритме планирования размещения и организации доступа к данным облачных сервисов и приложений, что является оригинальным подходом.

Реализационная часть

По сравнению с имеющимися аналогами ключевой особенностью спроектированного алгоритма управления размещением облачных приложений в самоорганизующемся хранилище ЦОД являются эвристический анализ потоков запросов, а также классификация трафика по типу передаваемой информации [10–12]. Гибкость предлагаемого решения обусловлена виртуализацией хранилища данных. Это позволяет организовывать динамическое размещение данных и приложений на физических устройствах для обеспечения постоянного доступа к услугам. Разработанное решение дает возможность масштабировать сервисы и приложения, размещаемые в облачной системе на несколько виртуальных устройств хранения. Это обеспечивает сокращение времени отклика приложения и повышает отказоустойчивость облачной платформы в целом. Формирование самоорганизующейся системы хранения с использованием средств виртуализации ресурсов дает возможность не только повысить надежность инфраструктуры, но и собирать информацию о востребованности различных облачных ресурсов, используемых приложениями и

сервисами в процессе работы. Алгоритм управления размещением приложений и сервисов в виртуальном самоорганизующемся хранилище данных использует модель облачного ресурса, характеризующую параметры и структуру связи объектов ЦОД.

Формализованная в рамках настоящего исследования модель использует мультиагентный подход для сбора информации о состоянии носителей информации и вычислительных узлах, задействованных для работы облачной системы. Для обработки полученной информации используются алгоритмы интеллектуального анализа данных (Data Mining), основанные на нейросетевом подходе. Результатом применения данного подхода являются две карты, одна из которых характеризует размещение виртуальных хранилищ данных внутри самой облачной платформы с привязкой к физическим устройствам, другая описывает непосредственно востребованность информации в системе хранения. На основе алгоритма эвристического анализа построенных карт и интеллектуального прогнозирования управляющая система перераспределяет данные внутри виртуального хранилища и сами узлы хранения между физическими устройствами. При этом перестроение карт происходит постоянно, а интервалы времени изменения зависят от интенсивности обращения приложений и сервисов к системе хранения.

В рамках исследования определен наиболее оптимальный интервал времени для анализа карт, при котором план перестроения самоорганизующихся виртуальных хранилищ данных облачной платформы будет наименее ресурсоемким, но в то же время максимально эффективным.

Предложенный алгоритм управления размещением приложений в самоорганизующемся виртуальном хранилище данных позволяет также повысить производительность работы всей облачной платформы в целом. При этом благодаря адаптивной маршрутизации и балансировке потоков данных, передаваемых между запущенными экземплярами облачных приложений, обеспечивается не только должное качество обслуживания, но и компактное расположение устройств [13]. Кроме того, алгоритм управления размещением облачных приложений решает задачу оптимизации количества задействованных физических устройств хранения, что, в свою очередь, позволяет повысить экономическую эффективность использования облачных платформ. Для получения результата используется информация от агентов и управляющих узлов облачной платформы [14].

Для эффективной обработки поступающего потока запросов пользователей облачные приложения и сервисы, как правило, задействуют сразу несколько типов ресурсов инфраструктуры виртуального ЦОД. При этом характеристики времени

полнения каждого запроса зависят от параметров доступа к конкретному облачному ресурсу. Для повышения эффективности обработки запросов в облачной системе необходимо проводить упреждающую подготовку ресурсов, одновременно оптимизируя характеристики доступа, относительно времени отклика приложений. Разработанное алгоритмическое решение осуществляет проактивное планирование размещения облачных ресурсов, тем самым подстраивая каждый экземпляр самоорганизующегося виртуального хранилища данных под текущую востребованность данных и облачных приложений. Для оптимизации характеристик доступа к облачным ресурсам алгоритм позволяет осуществлять адаптивную балансировку потоков запросов пользователей в зависимости от текущей нагрузки на выделенный виртуальный узел самоорганизующейся системы хранения данных. На рисунке 1 представлена блок-схема работы алгоритма размещения облачных приложений в самоорганизующемся виртуальном хранилище данных ЦОД.

Рис. 1. Обобщенная блок-схема алгоритма управления размещением облачных приложений в программно-конфигурируемых хранилищах данных виртуального ЦОД

Fig. 1. A generalized flow chart of a control algorithm for setting cloud applications in software defined data storage of virtual DPC

Всего в работе алгоритма управления размещением облачных приложений в самоорганизующемся виртуальном хранилище ЦОД можно выделить три этапа планирования и оптимизации выполнения потока запросов, формируемых пользователями.

На первом этапе работы алгоритма проводится анализ параметров запроса, поступающего от пользователя к приложению, а именно, определяются тип приложения и вид передаваемых данных. Запрос данных является многокомпонентным, то есть для организации канала задействуются сразу несколько элементов системы. Поэтому на втором этапе система управления определяет наиболее подходящие ресурсы, способные гарантировать выполнение запроса. Если среди ресурсов существует вариативность, на основе информации о прогнозе востребованности каждого из доступных вариантов ресурса строится ранжированный список с привязкой к виртуальному хранилищу данных. При этом учитываются только наименее загруженные хранилища данных. На третьем этапе работы алгоритма анализируются текущие параметры ресурсов и прогнозируется время, необходимое на выполнение текущего запроса. Полученные результаты сохраняются для дальнейшего использования при самообучении алгоритма. В случаях высокой ресурсоемкости запроса для предварительной оценки в алгоритме применяется поэлементный анализ объектов доступа, входящих в запрос.

Экспериментальная часть

Для оценки эффективности разработанного алгоритма управления размещением облачных приложений в самоорганизующемся хранилище данных виртуального ЦОД авторами исследована работа облачной системы, построенной на базе Openstack с различными параметрами. В качестве эталона для проведения эксперимента применялись типовые алгоритмы, используемые облачными системами для распределения экземпляров облачных приложений и востребованных ими данных по вычислительным узлам системы хранения. Для анализа эффективности и производительности работы алгоритмов на различных системах хранения были определены типовые условия эксперимента, включающие традиционные устройства на магнитных дисках (HDD), твердотельные накопители (SSD) и виртуальные хранилища (SDS). Для экспериментального исследования создан прототип облачной среды, включающий в себя основные узлы, а также программные модули для разработанных алгоритмов, выполняющие перераспределение запросов пользователей к данным в самоорганизующемся виртуальном хранилище.

В облачной системе OpenStack реализован модуль, применяющий разработанные алгоритмы управления размещением облачных приложений в самоорганизующихся хранилищах данных виртуального ЦОД для рационального использования вычислительных ресурсов облачной системы и эффективного размещения виртуальных машин по физическим узлам, а также связанных с ними данных. В ходе эксперимента для анализа данных создан поток запросов, аналогичный реальным запросам пользователей к облачным приложениям и сервисам, основанный на архивных данных за последние три года, содержащих параметры и интенсивность обращения к различным видам ресурсов облачной системы, классифицированных по типам данных и структуре запроса. Для воспроизведения нагрузочного эксперимента использовались средние значения полученных данных. Поток запросов пользователей в ходе эксперимента распределялся между вычислительными узлами облачной системы в зависимости от следующих параметров доступа: тип клиента, тип сервиса. Общее количество запросов, сгенерированных в ходе проведения испытаний и поступивших на вход системы, составило 100 000.

Для проведения экспериментального исследования построены три технические площадки, включающие в себя физические системы хранения на базе магнитных жестких дисков (площадка 1) и твердотельных накопителей (площадка 2), а также виртуальное хранилище данных (площадка 3). На каждой из технических площадок последовательно проводился ряд вычислительных экспериментов, направленных на интенсивное выполнение операции чтения (эксперимент 1) (рис. 2), записи (эксперимент 2) (рис. 3) и одновременно операций чтения и записи данных (эксперимент 3) (рис. 4).

Для проведения экспериментальных исследований был определен интервал времени, соответствующий максимальному значению пиковой активности пользователей в облачной системе. Время каждого эксперимента составило один час. Результаты экспериментальных исследований доказывают, что размещение облачных приложений с использованием предложенного алгоритма более эффективно независимо от типа физических устройств, используемых в хранилище данных.

Результаты экспериментальных исследований показывают возможность снизить на 20–25 % количество отказов в обслуживании при размещении информации, востребованной приложениями и пользователями, в виртуальных самоорганизующихся хранилищах данных, что подтверждает целесообразность использования алгоритма для предоставления эффективного доступа к приложениям и сервисам, развернутым в виртуальном ЦОД на базе облачной платформы.

В дополнение к данным экспериментам авторами проведено исследование, направленное на

Рис. 2. Анализ времени отклика приложений при выполнении запросов на чтение данных (эксперимент 1)

Fig. 2. An analysis of application response time when running read request (experiment 1)

Рис. 3. Анализ времени отклика приложений при выполнении запросов на запись данных (эксперимент 2)

Fig. 3. An analysis of application response time when running write request (experiment 2)

Рис. 4. Анализ времени отклика приложений при одновременном выполнении запросов на чтение и запись данных (эксперимент 3)

Fig. 4. An analysis of application response time when running read and write request simultaneously (experiment 3)

оценку количества задействованных в работе устройств хранения. На рисунке 5 представлен график, показывающий эффективность использования ресурсов системы хранения данных экспериментального стенда виртуального ЦОД.

Рис. 5. График эффективности использования вычислительных ресурсов системы хранения данных в программно-управляемой инфраструктуре виртуального ЦОД

Fig. 5. A diagram of computing resource usage efficiency in a cloud system

За счет оптимального выделения ресурсов на каждом узле системы хранения гарантировано обеспечение совместной работы всех запущенных экземпляров облачных приложений, что актуально при значительной нагрузке на систему хранения. Разработанный алгоритм управления размещением облачных приложений и сервисов в виртуальной самоорганизующейся системе хранения данных позволяет высвободить от 20 до 30 % устройств хранения. Таким образом, предложенный алгоритм показал свою эффективность как для традиционных хранилищ, так и для систем с неоднородной конфигурацией физических узлов и виртуальных машин.

В результате исследования разработана модель сервис-ориентированного облачного приложения, описывающая структуру, связи, а также концепцию применения виртуальных самоорганизующихся хранилищ данных для размещения приложений и данных в облачной системе с учетом технических ограничений виртуального ЦОД. Разработанная модель позволила реализовать алгоритм управления размещением облачных приложений в виртуальной самоорганизующейся системе хранения. В ходе экспериментальных исследований установлено, что применение разработанного алгоритма позволяет сократить время отклика облачных приложений и сервисов и, как следствие, повысить производительность обработки запросов пользователей. При этом использование виртуальной самоорганизующейся системы хранения позволяет сократить количество физических устройств, что говорит об эффективности разработанного решения.

Работа выполнена при поддержке РФФИ, научные проекты №№ 16-37-60086 мол_а_дк, 16-07-01004.

Литература

1. Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Тимченко В.А., Шалфеева Е.А. Базовая технология разработки интеллектуальных сервисов на облачной платформе

IACPaaS. Ч. 2. Разработка агентов и шаблонов сообщений // Программная инженерия. 2016. № 1. С. 14–20.

2. Майер-Шенбергер В., Кукер К. Большие данные. Революция, которая изменит то, как мы живем, работаем, мыслим. М.: Манн, Иванов и Фербер, 2014. 240 с.

3. Bolodurina I., Parfenov D., Shukhman A. Approach to the effective controlling cloud computing resources in data centers for providing multimedia services Proc. 2015 Intern. Siberian Conf. on Control and Communications, SIBCON, 2015, pp. 714–717.

4. Болодурина И.П., Парфенов Д.И. Комплексное решение эффективного доступа к данным в гибридных облачных системах с учетом особенностей мультимедийных сервисов // Вестн. Оренбургского гос. ун-та. 2015. № 13 (188). С. 114–118.

5. Болодурина И.П., Парфенов Д.И. Управление потоками данных в высоконагруженных информационных системах, построенных на базе облачных вычислений // Системы управления и информационные технологии. 2015. № 1.1 (59). С. 111–118.

6. Тарасов В.Н., Полежаев П.Н., Шухман А.Е., Ушаков Ю.А., Коннов А.И. Математические модели облачного вычислительного центра обработки данных с использованием OpenFlow // Вестн. Оренбургского гос. ун-та. 2012. № 9. С. 150–155.

7. Устименко О.В. Перспективы развития распределенного хранения сверхбольших объемов данных // Матер. XVIII междунар. науч. конф. ОМУС ОИЯИ, Дубна, 2014. С. 198–207.

8. Wuhib F., Stadler R., Lindgren H. Dynamic resource allocation with management objectives – Implementation for an OpenStack cloud. Network and service management, 2012 Proc. 8th Intern. Conf. and 2012 Workshop on Systems Virtualization Management, 2012, pp. 309–315.

9. Bocchi E., Drago I., Mellia M. Personal Cloud Storage Benchmarks and Comparison. IEEE Transactions on Cloud Computing. 2015, vol. 99. DOI: 10.1109/TCC.2015.2427191.

10. Darabseh A., Al-Ayyoub M., Jararweh Y., Benkhelifa E., Vouk M., Rindos A. SDStorage: A Software Defined Storage Experimental Framework. Proc. of Cloud Engineering (IC2E), IEEE Press, 2015, pp. 341–346.

11. Plakunov A., Kostenko V. Data center resource mapping algorithm based on the ant colony optimization. Proc. of Science and Technology Conf. (MoNeTeC), IEEE Press, 2014, pp. 1–6.

12. Wu X., Kumar V., Ross J. Top 10 algorithms in data mining. Jour. Knowledge and Information Systems.

13. Singh A., Korupolu M., Mohapatra D. Server-storage virtualization: integration and load balancing in Data Centers. Proc. of the 2008 ACM/IEEE Conf. on Supercomputing. IEEE Press, 2008, pp. 1–12.

14. 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.

MODELING OF PLACING SERVICE-ORIENTED APPLICATIONS IN A SOFTWARE-DEFINED INFRASTRUCTURE OF THE VIRTUAL DATA CENTER

I.P. Bolodurina¹, Dr.Sc. (Engineering), Professor, prmat@mail.osu.ru

D.I. Parfenov¹, Ph.D. (Engineering), fdot_it@mail.osu.ru

V.N. Reshetnikov², Dr.Sc. (Physics and Mathematics), Professor, rvn_@mail.ru

¹ Orenburg State University, Pobedy Ave. 13, Orenburg, 460018, Russian Federation

² Center of Visualization and Satellite Information Technologies SRISA, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation

Abstract. The information technology development in recent years leads to using distributed service-oriented business applications. Thus, research on mechanisms of managing hosted applications in a cloud environment is currently important, as well as efficient allocation of appropriate resources for their work within the existing constraints.

The aim of the study is to simulate the structure and interaction mechanisms of typical cloud service-oriented applications, as well as definition of approaches to optimization of their placement in a storage system. The authors developed a generalized model of a service-oriented cloud application that includes three basic layers: an application model, a service model, and a resource model. As a resource in the model they consider some kind of object that describes the behavior and characteristics of individual elements of an infrastructure depending on the current status and settings of the virtual data center of a software-defined infrastructure. A distinctive feature of the model is that it is a universal resource that allows us to consider it from user's point of view (for whom it is a closed system) and from the point of view of the software-defined infrastructure of a virtual data center (for which it is an open system). The novelty of the model is simultaneous describing data placement, associated applications, as well as the state of the virtual environment, taking into account the network topology. To work out the resource model in the virtual data center of software-defined infrastructure in detail the authors developed a model of a software-defined storage as a separate element. It allows defining data parameters and applications' connections with virtual and physical nodes of a data storage.

Keywords: cloud applications, cloud services, computing resources, virtual data centers, software-defined storage devices, cloud computing, data mining, OpenStack.

Acknowledgements. The work has been done with support from RFBR, scientific projects no. 16-37-60086 мол_a_дк, 16-07-01004.

References

1. Gribova V.V., Kleshev A.S., Krylov D.A., Moskalenko F.M., Timchenko V.A., Shalfeyeva E.A. Basic technology of intellectual services development on the cloudy IACPaaS platform. Part 2. Agents and message templates development. *Programmaya inzheneriya* [Program Engineering]. 2016, no. 1, pp. 14–20 (in Russ.).
2. Mayer-Schönberger V., Cukie K. *Big Data: A Revolution That Will Transform How We Live, Work and Think*. John Murray Publ., 2013, 256 p. (Russ.ed.: Moscow, Mann, Ivanov i Ferber Publ., 2014, p. 240 p.).
3. Bolodurina I., Parfenov D., Shukhman A. Approach to the effective controlling cloud computing resources in data centers for providing multimedia services. *Proc. 2015 Int. Siberian Conf. on Control and Communications, SIBCON 2015*. 2015, pp. 714–717.
4. Bolodurina I.P., Parfenov D.I. Complete solution of efficient data access in a hybrid cloud systems according to multimedia services characteristics. *Vestnik Orenburgskogo gos. univ.* [Bulletin of the Orenburg State Univ.]. 2015, no. 13, pp. 114–118 (in Russ.).
5. Bolodurina I.P., Parfenov D.I. The flow of control in high loaded information systems based on cloud computing. *Sistemy upravleniya i informatsionnye tekhnologii* [Control systems and information technologies]. 2015, no. 1.1 (59), pp. 111–118 (in Russ.).
6. Tarasov V.N., Polezhaev P.N., Shukhman A.E., Ushakov Ju.A., Konnov A.L. Mathematical models of cloud computing datacenter based on openflow. *Vestnik Orenburgskogo gos. univ.* [Bulletin of the Orenburg State Univ.]. 2012, no. 9, pp. 150–155 (in Russ.).
7. Ustimenko O.V. Prospects for the development of ultra-large data volumes distributed storage. *Materialy XVIII mezhdunar. nauch. konf. obyedineniya molodykh uchenykh i spetsialistov OIJaI (OMUS)* [Proc. 18th Int. scientific conf. of the Association of Young Scientists and Specialists (AYSS)]. 2014, pp. 198–207 (in Russ.).
8. Wuhib F., Stadler R., Lindgren H. Dynamic resource allocation with management objectives – Implementation for an OpenStack cloud. Network and service management. 2012. *Proc. 8th Int. conf. and 2012 workshop on systems virtualization management*. 2012, pp. 309–315.
9. Bocchi E., Drago I., Mellia M. Personal Cloud Storage Benchmarks and Comparison. *IEEE Transactions on Cloud Computing*. 2015, vol. 99.
10. Darabseh A., Al-Ayyoub M., Jararweh Y., Benkhelifa E., Vouk M., Rindos A. SDStorage: A Software Defined Storage Experimental Framework. *Proc. of Cloud Engineering (IC2E)*. Tempe, IEEE Press, 2015, pp. 341–346.
11. Plakunov A., Kostenko V. Data center resource mapping algorithm based on the ant colony optimization. *Proc. of Science and Technology Conf. (Modern Networking Technologies) (MoNeTeC)*. Moscow, IEEE Press, 2014, pp. 1–6.
12. Wu X., Kumar V., Ross J. Top 10 algorithms in data mining. *Knowledge and Information Systems*. 2008, iss. 14. Available at: <http://www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf> (accessed July 12, 2016).
13. Singh A., Korupolu M., Mohapatra D. Server-storage virtualization: integration and load balancing in Data Centers. *Proc. 2008 ACM/IEEE Conf. on Supercomputing*. Austin, IEEE Press, 2008, pp. 1–12.
14. Bolodurina I.P., Reshetnikov V.N., Parfenov D.I. Resource distribution in an information system related to educational process remote support. *Programmnye produkty i sistemy* [Software & Systems]. 2012, no. 3, pp. 151–155 (in Russ.).