

Исследование алгоритмов планирования и управления распределением ресурсов в открытых облачных системах *

И.П. Болодурина, Д.И. Парфёнов

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Оренбургский государственный университет"

Проводимое исследование направлено на повышение эффективности использования высоконагруженных информационных систем, развернутых в облачной системе. Для этого планируется разработать модели, описывающие основные особенности обслуживания потоков с учетом топологии системы, сетевых сервисов и существующих систем планирования задач, а также методы управления потоками данных между процессами вычислительных задач. В рамках данной статьи решается задача исследования облачной системы и оценка эффективности схем управления с учетом различных алгоритмов планирования. С этой целью разработаны: модель облачной системы, метрики эффективности и методика экспериментального исследования алгоритмов планирования и методов управления потоками данных. Модели определяют функционалы вычислительных узлов и связанных потоков между сервисами всей системы в целом. Методика экспериментального исследования предполагает оценку эффективности совместной работы виртуальных машин с учетом алгоритмов планирования и методов управления потоками данных по описанным метрикам. Предложенные в рамках данной статьи решения являются основой разработанного симулятора облачной системы.

1. Введение

Современные интегрированные информационные системы (ИС) кроме обработки данных, как правило, поддерживают ряд сервисов и приложений, потребляющих в больших объемах вычислительные ресурсы. При этом рост потребления выделенных ресурсов, в отличие от изменения их доступного объема, имеет экспоненциальный характер [1]. Кроме интенсивного роста потребления ресурсов другой проблемой является неоднородность поступающих потоков запросов пользователей, возникающей из-за многофункциональности современных информационных систем, выраженной в поддержке различных мультимедийных сервисов на единой платформе. Это требует существенных изменений к подходу в организации самой архитектуры приложений и управлению вычислительными ресурсами.

На сегодняшний день наиболее востребованной является концепция облачных вычислений. Основным преимуществом, которой является возможность динамического управления ресурсами с учетом зависимости от их востребованности конечными пользователями.

Наиболее гибкой является архитектура облачных вычислений, построенная по принципу «инфраструктура как услуга» (IaaS). Использование данной архитектуры предоставляет пользователям возможность не только обработки и хранения данных, но и управления пулом вычислительных ресурсов, включая запуск операционных систем и прикладных программных комплексов, а также создание виртуальных машин и сетей [3].

Отличительной особенностью облачных систем уровня инфраструктуры является применение виртуализации для реализации серверов. Преимуществом использования виртуальных машин является консолидация ресурсов. На одном физическом вычислительном узле могут размещаться сразу несколько виртуальных машин, обслуживающих различные приложения или сервисы. Облачные вычисления позволяют достигать существенной экономии средств за счет обеспечения высокой плотности загрузки серверов.

Однако объединение вычислительных мощностей и сокращение накладных расходов на инфраструктуру не снимает существующие ограничения по производительности облачных си-

* Представленная работа выполнена при поддержке Российского фонда фундаментальных исследований проект №13-07-00198.

стем. Для эффективного использования построенной облачной инфраструктуры необходима разработка методов и алгоритмов, позволяющих осуществлять управление основными объектами вычислительной системы. Для этого следует определить связи между ключевыми объектами и описать ее структуру [3].

2. Модель виртуализации вычислительных ресурсов облачной системы

Концепция виртуализации вычислительных ресурсов строится на базе абстракций, представляющих собой кортежи отношений, связанных между собой элементами подмножеств. В настоящем исследовании нами разработана модель вычислительных ресурсов облачной системы.

Облачную систему можно представить в виде связанных между собой множеств объектов, включающих в себя: вычислительные узлы $Snode$, систем хранения данных $Sstg$, сетевых хранилищ данных $Snas$ и серверов расписаний $Srasp$. Число объектов в каждом множестве и их состав может отличаться [4].

На каждом вычислительном узле одновременно запущено сразу несколько экземпляров виртуальных машин, описываемых кортежем $Snode_i = \{VM_{i,1}, VM_{i,2}, \dots, VM_{i,k}\}$, где k количество виртуальных машин на i -ом одном вычислительном узле, $i=1..l$ (l - общее число узлов). При этом каждая виртуальная машина содержит набор приложений $VM_j = \{App_{j,1}, App_{j,2}, \dots, App_{j,m}\}$, обеспечивающих, поддержку работы определенных сервисов, $j=1..m$ (m число виртуальных машин). В состав сетевого хранилища данных входит набор готовых образов виртуальных машин $Snas_y = \{VMimg_{y,1}, VMimg_{y,2}, \dots, VMimg_{y,z}\}$, $y=1..z$ (z число хранилищ данных), содержащих требуемую операционную систему с предустановленным программным обеспечением, и заданными параметрами виртуального аппаратного обеспечения $VMimg_{y,1} = \{OS_1, OS_2, \dots, OS_r\}$. Работа всей облачной системы строится с использованием системы планирования выполнения каких-либо операций $Srasp = \{Rtask_1, Rtask_2, \dots, Rtask_f\}$, задаваемых посредством сервера расписаний. В распределенной системе хранения данных, состоящей, как правило, из отказоустойчивых RAID массивов $Sstg_f = \{Rdisk_1, Rdisk_2, \dots, Rdisk_d\}$, осуществляется размещение необходимой для работы мультимедийных сервисов информации $Rdisk_d = \{Data_1, Data_2, \dots, Data_g\}$ [5]. Помимо этого, в облачной системе также присутствуют виртуальные и физические коммутаторы, соединяющие между собой все компоненты в единую сеть. Каждый из компонентов, входящих в множество узлов $Shcn = (Snode, Snas, Srasp, Sstg, VM, \dots)$, облачной системы обладает следующими характеристиками:

$$Shcn = (State, Mem, Desk, Diskn, Core, Lan), \quad (1)$$

где, $State \in \{\text{«on»}, \text{«off»}\}$ – состояние объекта облачной системы;

$Mem \in N$ — объем оперативной памяти установленной для узла облачной системы;

$Disk \in N$ — объем дискового пространства устройств хранения данных, установленных для узла облачной системы;

$Diskn \in N$ — количество устройств хранения данных, установленных для узла облачной системы;

$Core \in N$ — количество вычислительных ядер процессора узла облачной системы;

$Lan \in N$ — максимальная пропускная способность сетевого адаптера узла облачной системы.

При этом для виртуальных машин из множества $Snode$, может быть создано подмножество $VMnode = \{Snode, Snas, Sstg, \dots\}$, включающее в себя все компоненты облачной системы, что позволяет масштабировать архитектуру и изолировать вычислительные ресурсы для различных сервисов друг от друга. Декомпозиция ресурсов облачной системы и ее основных компонентов представлена на рис. 1.

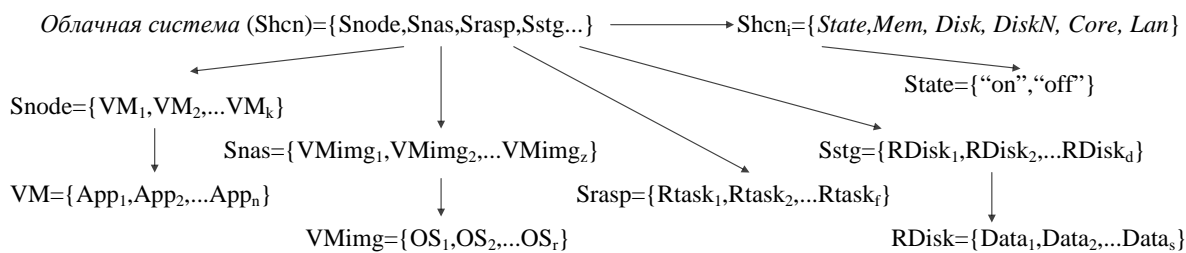


Рис. 1. Декомпозиция ключевых ресурсов облачной системы

Кроме того, так как облачная система является динамическим объектом, изменяющимся в моменты времени t , ее состояние может быть формализовано в виде ориентированного графа вида:

$$Shcn(t) = (Node(t), Connections(t), App(t)), \quad (2)$$

где вершины — $Node(t) = \{Node_1, Node_2, \dots, Node_\lambda\}$ представляют собой активные элементы, входящие в одно из множеств $Snode_i$, $Sstg_j$, $Snas_k$, $Srasp_m$;

$Connections(t) = \{Connections_1, Connections_2, \dots, Connections_v\}$ — активные подключения пользователей к виртуализованным приложениям;

$App(t) = \{App_1, App_2, \dots, App_n\}$ — активные экземпляры приложений, запущенные на виртуальных ресурсах.

Таким образом, нами определен состав вычислительных ресурсов, описаны структура связей и механизмы взаимодействия основных компонентов облачной системы. С точки зрения широкополосного трафика одновременное обслуживание разнородных запросов пользователей такой системы является нетривиальной задачей. При этом также стоит учитывать ограниченность вычислительных ресурсов. Для оптимизации механизма предоставления доступа к ресурсам информационной системы необходимо проанализировать основные потоки передаваемых данных в облачной системе.

3. Модель потоков данных информационных систем, построенных на базе облачных вычислений

Потоки передаваемых данных в информационных системах, как правило, обрабатываются в несколько этапов, называемых фазами обслуживания. При этом на каждой фазе может применяться несколько однотипных элементов, обеспечивающих балансировку и распределение нагрузки между элементами информационной системы. Количество элементов в каждой фазе зависит от функционала информационной системы и количества приложений, входящих в ее состав. Представим информационную систему в виде кортежа:

$$IS = \{S_1, \dots, S_r\}, \quad (3)$$

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

Количество фаз обслуживания f на пути потока запросов пользователей информационной системы зависит от ее архитектуры. Архитектура облачной системы представляет собой трехфазную систему массового обслуживания с соответствующими законами распределения интервалов длительностей обслуживания для каждого типов заявок на каждой фазе. Фактически облачная система представляет собой модель с 3 классами заявок G3/G3/M, где M - число виртуальных машин.

Учитывая ранее описанную архитектуру облачной системы, включающую в себя рекурсивные объекты, усложняющие ее структуру, модель может быть модифицирована для работы с большим количеством фаз. В дальнейшем потоки данных, формируемые в информационной облачной системе, будем рассматривать для системы дистанционного обучения, построенной с использованием современных средств телекоммуникаций, так как хранимые в ней данные позволяют промоделировать потоки поступления и обслуживания заявок пользователей с различной плотностью вероятностей.

Определим назначение каждой из фаз обслуживания запросов относительно их расположения в последовательности обработки запросов пользователей:

- первая фаза представлена группой элементов в виде вычислительных узлов контроллера, управляющего распределением потоков данных между ресурсами информационной системы, расположенной в облаке;
- вторая фаза представлена группой элементов в виде виртуальных машин, размещенных на вычислительных узлах информационной системы, обеспечивающих динамическое масштабирование вычислительных ресурсов в облаке;
- третья фаза представлена в виде приложений, запущенных на виртуальных машинах, выполняющих обработку данных пользователя.

Кроме того, к элементам третьей фазы относят узлы системы хранения данных и системы управления базами данных, участвующие в обслуживании запросов пользователей, как неотъемлемые части информационной системы обеспечивающей доступ к мультимедийным сервисам в облаке.

Детализируем кортеж элементов, входящих в состав информационной системы, в виде:

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

где S_i^j - i -й элемент j -й фазы;

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

Введем также входные S_i^0 элементы, передающие поток данных в информационную систему, и выходные S_i^4 элементы, принимающие потоки данных из облачной инфраструктуры. Следовательно, кортеж, описывающий элементы информационной системы, преобразуется к виду:

$$IS = \{S_1^0, \dots, S_l^0, S_1^1, \dots, S_n^1, S_1^2, \dots, S_m^2, S_1^3, \dots, S_k^3, S_1^4, \dots, S_p^4\}, \quad (5)$$

где $m \in N, n \in N, k \in N$ — 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.

Каждый элемент S_i^j информационной системы в любой момент времени может обслуживать несколько запросов, поступивших от разных пользователей. При этом в процессе выполнения запроса пользователя образуются потоки данных на входе и выходе из элемента системы, обладающие индивидуальными характеристиками, изменяющимися во времени. 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. Каждый поток запросов можно описать набором характеристик. Пусть, для элемента системы S_i^j существует l_i^j входящих потоков, и p_i^j выходящих. Тогда для входящего v -го потока, $v=1..l_i^j$, введем набор характеристик:

$$X_i^{(j,v)}(t) = (x_{1,i}^{(j,v)}(t), \dots, x_{k,i}^{(j,v)}(t))^T, \quad (6)$$

где $x_{1,i}^{(j,v)}$ — интенсивность поступления заявок в каждом v -м потоке на входе элемента S_i^j ;

$x_{2,i}^{(j,v)}$ — время обслуживания v -го потока заявок в элементе S_i^j ;

$x_{3,i}^{(j,v)}$ — интенсивность обслуживания v -го потока заявок в элементе S_i^j .

$x_{4,i}^{(j,m)}$ - дисциплина обслуживания v -го потока заявок в элементе S_i^j , определяющая порядок обслуживания в соответствии с алгоритмом расстановки приоритетов в информационной системе;

$x_{5,i}^{(j,v)}$ — класс обслуживания v -го потока заявок в элементе S_i^j ;

$x_{6,i}^{(j,v)}$ — количество заявок, поступивших из v -го потока в элемент S_i^j .

Для выходящего μ -го потока, $\mu=1..p_i^j$ элемента системы S_i^j набор характеристик включает в себя:

$$Y_i^{(j,\mu)}(t) = (y_{1,i}^{(j,\mu)}(t), \dots, y_{k,i}^{(j,\mu)}(t))^T, \quad (7)$$

где $y_{1,i}^{(j,\mu)}$ — интенсивность поступления заявок в каждом μ -м потоке на выходе элемента S_i^j ;

$y_{2,i}^{(j,\mu)}$ — интенсивность обслуживания μ -го потока заявок на выходе элемента S_i^j ;

$y_{3,i}^{(j,\mu)}$ — дисциплина обслуживания μ -го потока заявок на выходе элемента S_i^j , определяющая порядок передачи данных в соответствии с алгоритмом расстановки приоритетов в информационной облачной системе;

$y_{4,i}^{(j,\mu)}$ — класс обслуживания заявок μ -го потока заявок на выходе элемента S_i^j ;

$y_{5,i}^{(j,\mu)}$ — количество обслуженных заявок в μ -м потоке на выходе элемента S_i^j .

Интенсивность обслуживания рассматриваемых потоков заявок информационной системы, запрашиваемых пользователями, зависит от класса обслуживания, определяемого уровнем конечного приложения. В общем случае интенсивность поступления и обслуживания потока заявок для каждого класса приложений определяется функцией распределения, описывающей следующие законы распределения [6]:

- для приложений, осуществляющих обработку потоковых данных (передача видео трафика) характерно распределение Парето;
- для приложений, осуществляющих обработку статических данных (передача бинарного трафика) характерно распределение Вейбулла;
- для приложений, осуществляющих обработку статических данных (передача трафика данных наибольшего размера) характерно распределение Хи-квадрат.

При обслуживании заявок в информационной системе, построенной с использованием облачной архитектуры, траектория прохождения фаз для каждого потока может изменяться динамически. При этом количество уникальных потоков напрямую зависит от числа элементов на каждой фазе.

Множество всех потоков, входящих в элементы на каждой j -й фазе, включая входящие потоки пользователей информационной системы, можно представить в виде:

$$X^j = \bigcup_{i=0}^{n_j} X_i^j, \quad (8)$$

где j номер фазы обслуживания, n_j количество элементов на j фазе. Следовательно, все входящие потоки информационной системы, реализованной с использованием облачных вычислений, можно представить в виде:

$$X = \bigcup_{j=0}^f X^j, \quad (9)$$

где f количество фаз обслуживания.

Для выходных потоков справедливо аналогичное условие: $Y^j = \bigcup_{i=0}^{n_j} Y_i^j \Rightarrow Y = \bigcup_{j=0}^f Y^j$, где n количество элементов на j фазе, $j=1..f$.

Для эффективного обслуживания запросов пользователей, образующих потоки данных в информационной системе, должно существовать однозначное отображение вида $R: X \rightarrow Y$.

При этом для обслуживания каждой заявки в любой момент времени в облачной информационной системе строится матрица переходов H между фазами обслуживания в зависимости от класса запроса и текущей загруженности элементов системы. Граф перехода между фазами может быть получен с использованием функции $Y_e^{j-1} = R(X_i^{j,\nu})$, $Y_e^{j-1} \in Y$, где e – номер элемента, принадлежащего фазе $j-1$ и направляющего поток данных ν к элементу S_i^j фазы j , $\nu=1..l_i^j$.

Тогда множество всех входных потоков для любого элемента S_i^j , полученных от элемента S_i^{j-1} информационной системы, находящегося на предыдущей фазе, представим в виде:

$$X_i^{j,j-1} = R_j^{-1}[Y_i^{j-1} \cap R(X_i^j)], \quad (10)$$

где j — фаза обслуживания.

Тогда выходящие потоки элемента S_i^j направленные к элементу S_i^{j+1} представим в виде:

$$Y_i^{j,j+1} = Y_i^j \cap R(X_i^{j+1}), \quad (11)$$

где j — фаза обслуживания.

Тогда выражениями $X^{j*} = \bigcup_{i=0}^n X_i^j$ и $Y^{j*} = \bigcup_{i=0}^m Y_i^j$ можно описать соответственно входные и выходные потоки фазы j (Рис. 2).

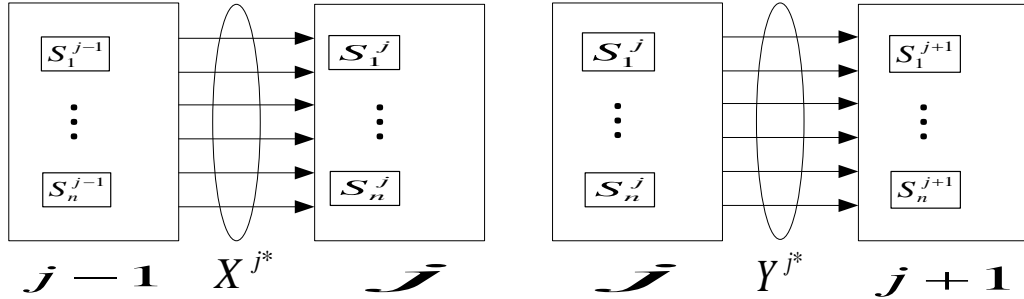


Рис. 2. Графическое представление потоков данных, циркулирующих на каждой фазе

В реальных информационных системах, как правило, выходные потоки могут пересекаться и попадать на обслуживание на один и тот же вычислительный узел или выполняться в рамках одного экземпляра приложения, что приводит к образованию внутренних очередей на каждой из фаз обслуживания. Для описания данного процесса необходимо определить связи выходных потоков данных элемента S_i^j фазы j со всеми элементами фазы $j+1$. С учетом этого множество Y^{j*} расширяется и принимает вид:

$$Y^{j*} = \bigcup_{S_i^j} \left[Y_i^{j,0} \cup \left(\bigcup_{S_i^{j+1}} Y_i^{j,j+1} \right) \right], \quad (12)$$

При этом для заключительной фазы обслуживания необходимо также учитывать передачу потока данных пользователю.

Для описания пересекающихся входящих в фазу потоков введем две функции:

$$X^{j,j+1} = Q_x^j(Y^{j*}), \quad (13)$$

$$Y^{j,j+1} = Q_y^j(Y^{j*}), \quad (14)$$

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

Аналогичным образом можно определить множество входных потоков поступающих в фазу обслуживания. При этом также справедливо пересечение потоков однотипных запросов пользователей. Следовательно, входящий поток данных, поступающий на элемент S_i^j , принадлежащий фазе j , от всех элементов фазы $j-1$ можно представить в виде:

$$X^{j*} = \bigcup_{S_i^j} \left[X_i^{j,0} \cup \left(\bigcup_{S_i^{j-1}} X_i^{j,j-1} \right) \right], \quad (15)$$

Для описания пересекающихся выходящих из фазы потоков введем две функции:

$$X^{j,j-1} = P_x^j(X^{j*}), \quad (16)$$

$$Y^{j,j-1} = P_y^j(X^{j*}), \quad (17)$$

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

Таким образом, функции (13) и (17) описывают потоки данных между фазами обслуживания информационной системы, расположенной в облаке. Для описания всей многофазной информационной облачной системы формализуем описание потоков каждой фазы в виде отобра-

жения $R^j : X^j \rightarrow Y^j$. Тогда потоки данных информационной системе, расположенной в облаке можно представить в виде:

$$Y_i^j = R^j(X_i^j) = \begin{cases} R(X_i^j), & X_i^j \in X^j \\ P_Y^j(X^{j*}), & X^{j*} \in \bigcup_{S_i^j} \left[X_i^{j,0} \cup \left(\bigcup_{S_i^{j-1}} X_i^{j,j-1} \right) \right] \\ Q_X^j(Y^{j*}), & Y^{j*} \in \bigcup_{S_i^j} \left[Y_i^{j,0} \cup \left(\bigcup_{S_i^{j+1}} Y_i^{j,j+1} \right) \right] \end{cases} \quad (18)$$

Так как потоки данных и их характеристики могут изменяться во времени, следует дополнить полученное описание потоков данных в информационной системе параметром времени t . Кроме того, так как на информационную систему оказывают влияние не только внутренние процессы и потоки передаваемых данных, но и внешние факторы описание системы можно дополнить параметром внешнего воздействия F . Тогда потоки передаваемых данных в облачной информационной системе можно описать выражением вида: $Y_i^j = R^j(X_i^j, t, F)$.

4. Алгоритм управления виртуальными ресурсами облачной системы

Представленные модели позволяют определять наиболее подходящие вычислительные узлы информационной системы и виртуальные машины, содержащие требуемые экземпляры мультимедийных приложений. Однако для предоставления доступа к вычислительным ресурсам необходимо осуществить запуск всей инфраструктуры облачной системы, при этом основной задачей является обеспечение требуемого количества узлов для бесперебойной работы пользователей [7]. В условиях физической ограниченности ресурсов, необходимо осуществлять эффективное управление виртуальными ресурсами и приложениями.

Основной задачей управления вычислительными ресурсам является выбор их количества в каждый последующий момент времени, то есть составление плана. При организации доступа к высоконагруженным информационным системам эффективное планирование особенно актуально, так как создаваемая нагрузка на сервисы может изменяться в широких пределах за достаточно короткие интервалы времени. Для того, чтобы предотвратить исчерпание выделенных для работы виртуальных машин ресурсов в уже запущенных приложениях и подготовить дополнительные вычислительные мощности требуется динамически составлять план потребления ресурсов и применять его для оптимизации структуры облачной системы.

В отличие от других информационных систем, поток запросов пользователей в образовательной среде, можно предсказать, за счет организации подписки на мультимедийные сервисы. Алгоритм управления, применяемый для предоставления доступа пользователям к виртуальным информационным ресурсам, работает в двух связанных друг с другом процессах.

Одним из процессов является планирование. Во время работы с выделенными ресурсами высоконагруженной информационной системы алгоритм управления осуществляет сбор данных о поступающих в систему запросах и производит анализ, классифицируя их согласно уровням, определяемым согласно критичности приложений для бизнес-процессов. При этом входными данными для алгоритма управления выступают приложения информационной системы, описанные в виде выделенного шаблона, включающего в себя образ виртуальной машины с заданными параметрами конфигурации аппаратного и программного обеспечения и характеристиками, обеспечивающими учет сессий пользователей.

На основе полученного шаблона и данных приведенного анализа истории подключений к вычислительным ресурсам информационной системы, расположенной в облаке, алгоритм управления осуществляет расчет конфигурации для развертывания требуемого сервиса. При этом в случае идентичности виртуальных машин по программному обеспечению предлагаются варианты уже созданных ранее образов, размещенных на сетевом хранилище данных. Для оптимизации использования выделенных вычислительных ресурсов в условиях физических огра-

ничений аппаратного обеспечения, алгоритм формирует три варианта конфигурации виртуальных машин.

Первый вариант — обеспечит запас производительности в случае непредвиденного увеличения количества пользователей. Коэффициент масштабирования при этом рассчитывается динамически. Облачной системой будет производиться масштабирование в рамках всех доступных на текущий момент времени ресурсов вычислительной системы. Кроме того, алгоритм управления виртуальными ресурсами облачной системы, используя данные о текущей конфигурации облачной системы и ее характеристиках, формирует диапазон количества экземпляров виртуальных машин, доступных для запуска с учетом ограниченности ресурсов.

Второй вариант — обеспечит заведомо меньшую производительность виртуальной машины по сравнению с заданным количеством пользователей. Однако такой подход является наиболее эффективным при использовании сервисов для узкоспециализированной аудитории пользователей. Он позволяет сократить накладные расходы при сравнительно малом количестве пользователей относительно рассчитанного числа подписчиков. Как и в случае с первым вариантом конфигурации, алгоритм формирует диапазон доступного количества экземпляров виртуальных машин, что позволяет обеспечить параллельный запуск приложений на нескольких вычислительных узлах при увеличении нагрузки на приложение. Таким образом, производится разделение на независимые узлы общей вычислительной задачи, что также позволяет обеспечить соответствующее качество обслуживания для пользователей использующих приложения.

Третий вариант — создается с использованием заданных в конфигурации пользователем характеристик, включающих в себя фиксированный коэффициент масштабирования, и фиксированное количество экземпляров виртуальных машин, которые будут запущены независимо от количества пользователей сразу после выполнения алгоритма. При этом в рамках данного процесса алгоритм используется лишь для расчета максимального числа виртуальных машин, доступных в выбранной пользователем конфигурации для обеспечения ограничения по вычислительным ресурсам.

Вторым процессом, в рамках предлагаемого алгоритма, является непосредственное обслуживание запросов пользователей и масштабирование ресурсов во время работы приложений. Алгоритм управления виртуальными ресурсами и приложениями учитывает общее количество запросов от каждого из источников, что дает возможность прогнозировать нагрузку на запущенные в облачной инфраструктуре приложения. На основе полученных данных алгоритм управления виртуальными ресурсами в соответствии с заданным планом осуществляет миграцию виртуальных машин между физическими вычислительными узлами, масштабируя тем самым работу приложений.

Для эффективного использования ресурсов в рамках описанных процессов в сетевом хранилище образов формируются дополнительные экземпляры виртуальных машин, поддерживающие работу приложений для обеспечения доступа минимального количества пользователей. При этом, в случае прогнозирования увеличения нагрузки на какой-либо сервис, алгоритм управления виртуальными ресурсами разворачивает полноценный образ мультимедийного ресурса и анализирует входящие запросы пользователей. Если в течение заданного интервала времени нагрузка не превышает количества запросов стандартного потока в режиме «stand by», алгоритм переключает нагрузку на соответствующий образ, выключая виртуальную машину.

Подход, применяемый в предложенном алгоритме управления виртуальными ресурсами и приложениями, позволяет учитывать физические ограничения вычислительных ресурсов, и организовывать работу информационной системы с учетом входящего потока запросов пользователей регулируя при этом число запущенных экземпляров приложений.

5. Экспериментальная часть

Для оценки эффективности алгоритма управления вычислительными ресурсами облачной системы, построенного с учетом представленных моделей, нами проведено исследование работы информационной системы с различными параметрами. При этом в качестве эталонных данных для сравнения в эксперименте использовались типовые алгоритмы, применяемые в облачной системе Openstack. Для экспериментального исследования создан прототип облачный сре-

ды, включающей в себя основные узлы информационной системы ВУЗа, а так же разработанные алгоритмы модифицирующие обработку запросов пользователей. Схема комплексного подхода, использующая облачные вычисления представлена на рисунке 1 [6].

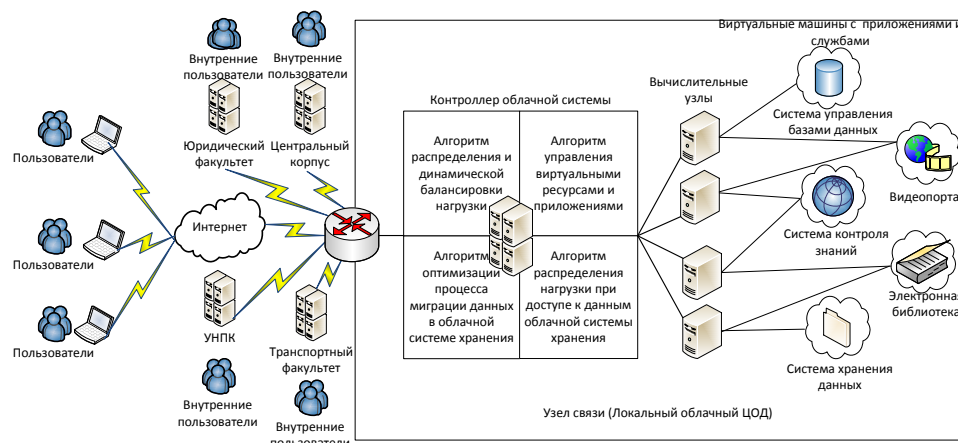


Рис. 3. Укрупненная схема оптимизации доступа к ИС ВУЗа на базе облачных вычислений

В облачной системе OpenStack [8] реализован модуль, применяющий разработанные алгоритмы для рационального использования вычислительных ресурсов и эффективного распределения виртуальных машин, а так же связанных с ними данных. В ходе эксперимента для анализа данных создан поток запросов, аналогичный реальному трафику информационной системы дистанционного обучения. Для этого с применением СУБД Oracle восстановлены данные о подключениях пользователей к информационным ресурсам ВУЗа за последние 3 года. 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 10000, что соответствует максимальному числу потенциальных пользователей системы. Все запросы классифицированы на 6 групп, характеризующих типичное поведение пользователей информационной системы. Группы экспериментов с первой по третью направлены на интенсивное поступление запросов к выделенному сервису информационной системы, с одновременным использованием других приложений. Группы экспериментов с 4 по 6 описывают работу выделенного приложения в информационной системе, испытывающего недостаток вычислительных ресурсов в виду превышения количества одновременных запросов. Для каждой из группы пользователей, задана интенсивность использования компонентов системы (видеопортала, системы тестирования, электронной библиотеки), а также объем востребованных данных. Время эксперимента составило один час, что соответствует наиболее длительному периоду времени пиковой нагрузки системы, зафиксированному в реальном трафике. В таблице 1 представлены основные показатели, полученные в результате экспериментов.

Таблица 1. Эффективность обслуживания запросов пользователей

Подсистемы СДО (тип данных)	контроль знаний (текст)	электронная библиотека (файлы)	видеопортал (видео- и аудио)	контроль знаний (текст)	электронная библиотека (файлы)	видеопортал (видео- и аудио)
Эксперимент	1			3		
Общее кол-во запросов	8000	1000	1000	1000	1000	8000
Объем данных	32650	9330	10340	4750	8210	92300
Кол-во обслуженных запросов (без балансировки нагрузки)	5443 (4352)	622 (418)	517 (356)	592 (465)	643 (512)	4320 (3985)
Интенсивность обслуживания зп/с.	90,71 (72,53)	10,36 (6,96)	8,61 (5,93)	9,8 (7,75)	10,71 (8,5)	72 (66,4)

Таблица 2. (продолжение) Эффективность обслуживания запросов пользователей

Подсистемы СДО (тип данных)	контроль знаний (текст)	электронная библиотека (файлы)	видеопортал (видео- и аудио)	контроль знаний (текст)	электронная библиотека (файлы)	видеопортал (видео- и аудио)
Эксперимент	2			4	5	6
Общее кол-во за- просов	1000	8000	1000	10000	10000	10000
Объем данных	4250	67200	10670	41700	87600	108000
Кол-во обслужен- ных запросов (без балансировки нагрузки)	632 (525)	5384 (4625)	560 (376)	6753 (5642)	6351 (5215)	5860 (4129)
Интенсивность обслуживания зп/с	10,5 (4,2)	89,73 (77,08)	9,3 (6,26)	112,5 (94,03)	105,85 (89,91)	97,6 (68,81)

Полученные данные подтверждают целесообразность применения алгоритмов управления виртуальными ресурсами для запуска различных типов мультимедийных приложений. По результатам проведенных экспериментов можно сделать вывод о снижении на 12-15% количества отказов в обслуживании при доступе к мультимедийным сервисам в условиях ограниченности ресурсов.

Кроме этого в рамках экспериментального исследования в облачной системе OpenStack проведено сравнение потребления выделенных виртуальных ресурсов по количеству виртуальных серверов для каждой из подсистем. На рис. 4 представлен график, показывающий эффективность работы системы управления вычислительными ресурсами облачной системы.



Рис. 4. График эффективности использования вычислительных ресурсов облачной системы

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 алгоритмов оптимизации возможно высвобождение от 20 до 30% выделенных ресурсов (виртуальных серверов).

Заключение

Таким образом, оценивая общий результат работы алгоритма управления виртуальными ресурсами облачной системы можно получить прирост производительности от 12 до 15% по сравнению со стандартными средствами, что является весьма эффективным при большой интенсивности запросов. Кроме того, сокращение числа выделенных виртуальных ресурсов позволяет масштабировать более эффективно масштабировать облачную систему, и обеспечить запас надежности при резком увеличении интенсивности использования выделенных приложений.

Литература

1. Парфёнов, Д. И. Распределение ресурсов в информационной системе дистанционной поддержки образовательного процесса / Д. И. Парфёнов, И. П. Болодурина, В. Н. Решетников // Программные продукты и системы – 2012. - № 3. - С. 151-155.
2. 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.
3. 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.
4. Wuhib F., Stadler, R., Lindgren H. Dynamic resource allocation with management objectives - Implementation for an OpenStack cloud // Network and service management, 2012 8th international conference and 2012 workshop on systems virtualization management, 2012. – С. 309-315.
5. 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.
6. Болодурина И.П., Парфёнов, Д.И. Управление потоками запросов при доступе к широкополосным мультимедийным образовательным ресурсам системы дистанционного обучения / Д.И. Парфёнов, // Проблемы управления. – 2013. – № 6. - С. 24-30.
7. Полежаев П.Н., Шухман А.Е., Ушаков Ю.А., Л. В. Легашев, Тарасов В.Н. Модельное исследование эффективного планирования задач для облачных вычислительных систем// Интеллект. Инновации. Инвестиции, № 4, 2013. – С. 140-145.
8. OpenStack Open Source Cloud Computing Software. [Электронный ресурс]. - Режим доступа: <http://www.openstack.org/>