УДК 004.75

Оптимизация поддержки вычислительных ресурсов на железнодорожном транспорте

Николай ИГНАТОВ

Игнатов Николай Александрович *– аспирант кафедры «Вычислительные системы и сети» Московского государственного университета путей сообщения (МИИТ), Москва, Россия.*

Nickolay A. IGNATOV

Статья посвящена особенностям предоставления виртуальных ресурсов в системах, основанных на облачных технологиях, но с учетом выполнения требований QoS. Описан адаптивный механизм, а также проведен сравнительный анализ работы адаптивного и статических механизмов предоставления ресурсов с помощью имитационных моделей. Рассмотрены такие вычислительные показатели для моделей, как среднее время обработки запроса, уровень отказа обслуживания, значение общего использования ресурсов системы при различных входных параметрах.

Ключевые слова: управление, информационные сети, качество обслуживания, облачные вычисления, виртуализация, моделирование, распределенные системы.

Сегодня ОАО «Российские железные дороги» представляет собой холдинг с разнообразными направлениями деятельности, функционирование которых невозможно без применения сложных информационно-вычислительных систем.

Каждая из 16 железных дорог, входящих в состав РЖД, имеет свои особенности, которые проявляются и в сфере внедрения вычислительных систем. Наряду с этим, ОАО «РЖД» является единой информационно-вычислительной инфраструктурой, объединенной в крупнейшую в стране волоконно-оптическую магистральную цифровую сеть связи, состоящую из более чем 900 узлов доступа в большинстве регионов России.

Корпоративная программа информатизации предполагает работу по таким направлениям, как: управление сбытом грузовых перевозок, управление сбытом и организацией пассажирских перевозок, управление перевозочным процессом, оптимизация управления содержанием инфраструктуры и подвижного состава, оптимизация управления финансовыми, трудовыми и материальными ресурсами, управление инвестициями и инновациями,

унификация и интеграция автоматизированных систем.

При этом важным является выбор технологии, с помощью которой будет построена инфраструктура для решения информационно-вычислительных задач, а также механизма ресурсного обеспечения системы, совместное использование которых позволит сократить общую стоимость информационно-вычислительной системы за счет исключения простоя ресурсов и при условии соблюдения требований к качеству обслуживания системы.

В данной работе рассмотрена облачная инфраструктура, основанная на технологиях виртуализации, в которой для ресурсного обеспечения используется механизм, автоматически адаптирующийся к изменениям рабочей загрузки.

Виртуализация – одна из самых современных технологий, которая допускает выполнение отдельных экземпляров операционной системы в среде виртуальной машины. Часто технологии виртуализации используются для создания облачной инфраструктуры. Однако существуют значительные проблемы с расчетом эффективного обеспечения и распределения ресурсов, которые используются системами, основанными на технологиях виртуализации. Данные проблемы затрагивают различные этапы работы системы, такие, как прогнозирование рабочей загрузки, виртуализация, моделирование производительности, ввод в эксплуатацию и мониторинг приложений в виртуальных средах. Если эти проблемы будут решены, тогда приложения смогут работать более эффективно, с уменьшением финансовых затрат, перераспределением неиспользуемых ресурсов и увеличением производительности в минуты наибольшей загрузки.

МОДЕЛЬ СИСТЕМЫ

Условные обозначения, принятые

P– множество инфраструктур облачных вычислений;

C_i – i-е ЦОД из P;

n – количество ЦОД;

s_j – j-й экземпляр приложения;

v_j – j-я виртуальная машина;

m – количество экземпляров виртуальных машин, предоставленных приложению;

G– рабочая нагрузка приложения;

r_l – l-й запрос конечного пользователя для G_s;

h– количество запросов, заданий или рабочих единиц, из которых состоит рабочая нагрузка;

r_l– время прибытия запроса r_l к поставщику приложения;

T_r – время отклика на запрос конечного пользователя;

T_s – максимально допустимое значение времени отклика на запрос конечного пользователя, удовлетворяющее требованиям QoS;

λ – ожидаемое количество запросов, которые поступят к поставщику приложения;

λ_{si}– ожидаемое количество запросов, которые поступят на экземпляр приложения;

$Rej(G_s)$– количество запросов из G_s, получивших отказ на обслуживание.

Системы, основанные на облачных вычислениях, объединяют центры обработки данных (далее – ЦОД) как сети виртуальных IaaS (вычислительные серверы, базы данных, сети) и PaaS (механизмы стабилизации загрузки и автоматического масштабирования) таким образом, что поставщики имеют возможность разворачивать приложения (SaaS) из любой точки мира с уровнем затрат, определяемых требованиями QoS. Обозначим $P = \{C_1,.., C_n\}$ как облачную инфраструктуру, где $C_1,..,$ C_n – ЦОД, из которых состоит P.

При развертывании приложение компонуется m экземплярами виртуальных машин (далее – ВМ) $\{v_1,.., v_m\}$, где значение m любое фиксируемое или изменяемое во времени, основанное на текущей рабочей загрузке и запрашиваемой производительности. Экземпляры приложений, являющиеся примерами SaaS программ, которые могут принадлежать предприятиям маленького или среднего масштаба, а также государственным организациям, использующим приложения, работающие через Облака. В рассматриваемом механизме приложения и платформы предоставляются одной организацией, а ресурсы, основанные на облачных технологиях, используются различными организациями.

Предположим, что возможно распределение один к одному между экземпляром

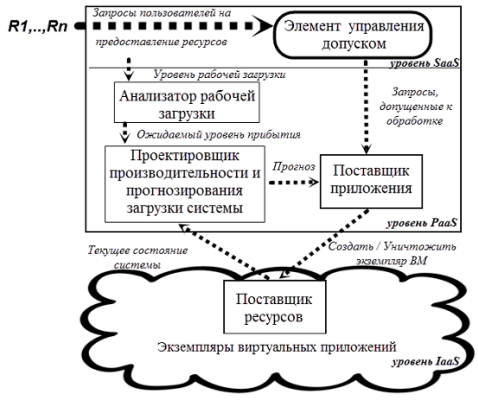

Рис. 1. Архитектура адаптивного механизма предоставления виртуальных ресурсов.

Pic. 1. The architecture of the adaptive mechanism of virtual resources provisioning.

приложения s_j и v_j экземпляром ВМ, а следовательно, можно говорить об их тождественности в рамках описываемого механизма.

Сценарий работы приложений в таком механизме связан с выполнением набора различных действий или функциональных возможностей, экземпляром приложения s_j для конечного пользователя. Особенности выполнения или функциональные характеристики приложения зависят от его модели. Адаптивный механизм предоставления ресурсов может быть реализован в рамках построения таких систем, как общественные компьютерные сервисы, например, folding@home [10], которые предоставляют функциональные возможности для выполнения математических расчетов в распределенной системе, и организации доступа к web-ресурсу, требующему балансировки загрузки.

Вычисление объемов ресурсов, необходимых для приложения, состоит из набора независимых задач, которые могут быть смоделированы как запросы на обслуживание, отправляемые конечными пользователями на экземпляры виртуальных приложений.

Как в системе распределенной обработки математических моделей, так и в организации доступа к web-ресурсу рабочую загрузку G_s можно представить набором из h независимых запросов $\{r_1,.., r_h\}$, которые принимаются от экземпляров приложений во время $\{t_1,.., t_h\}$.

Первостепенная задача механизма предоставления виртуальных ресурсов заключается в выполнении требований QoS. Основными параметрами QoS системы являются: T_s – максимально допустимое значение времени отклика на запрос конечного пользователя и $Rej(G_s)$ – количество запросов из G_s, получивших отказ в обслуживание. Соблюдение данных параметров важно, поскольку они имеют определяющее воздействие на мнение пользователя о SaaS приложении. Если время реагирования станет очень большим или запросы будут отклоняться, то пользователи прекратят применять приложение, что приведет к снижению его рентабельности.

АДАПТИВНЫЙ МЕХАНИЗМ

С целью выполнения требований QoS системы в рамках изменяющегося характера рабочей загрузки, а также с учетом сложности прогнозирования объема виртуальных ресурсов, характерных для облачных технологий, был разработан адаптивный механизм. Архитектура его представлена на рис. 1.

Администрирование различных программных компонентов архитектуры осуществляется поставщиком услуг. Уровень SaaS механизма содержит элемент управления допуском, основная задача которого заключается в допуске к дальнейшей обработке поступающих в систему запросов конечных пользователей.

Принцип работы элемента управления допуском связан с параметром k – размером очереди и учитывает отношение максимально допустимого значения времени откликов на запрос конечного пользователя T_s и T_r, в соответствии с равенством 1. Если число запросов к ВМ превышает k, то запрос отбрасывается элементом управления допуском и не отправляется поставщику приложения. Это гарантирует, что запросы или будут отброшены, или обслужены в момент, согласованный с клиентом:

$$k = \frac{T_s}{T_r}. \qquad (1)$$

Принятые запросы передаются на уровень PaaS, который включает несколько основных компонентов.

1) *Поставщик приложения* является ключевым элементом механизма, который принимает запросы конечных пользователей от элемента управления допуском

и предоставляет виртуальные машины и экземпляры приложений, основываясь на данных, поступающих от анализатора рабочей загрузки и проектировщика производительности и прогнозирования загрузки системы (далее – ППЗС). Если использование ресурсов ЦОД снижается, то поставщик приложений предлагает остановить некоторые экземпляры приложений. В подобной ситуации первыми будут остановлены экземпляры, не выполняющие обработку запросов. Если количество бездействующих экземпляров виртуальных приложений меньше, чем количество экземпляров, которые запланировано остановить, тогда экземпляры с низкой долей выполняемых запросов будут остановлены после завершения обработки принятых ими запросов.

В другом случае, когда ожидается рост количества поступающих запросов или не выполняются требования QoS, поставщик приложений приказывает создать больше экземпляров виртуальных приложений.

2) *Анализатор рабочей загрузки* это компонент, предназначенный для прогнозирования количества запросов, которые поступят в систему. Подобная информация используется при вычислении точного количества экземпляров приложений, требуемых для обеспечения целей QoS и распределения ресурсов.

Кроме оценки будущей загрузки системы анализатор оповещает о наступлении события, когда количество запросов на обслуживание изменится. Такое сообщение содержит информацию об ожидаемом уровне прибытия запросов, а также фиксирует время до изменения рабочей загрузки системы. То есть ППЗС предлагает иметь время для вычисления изменений в системе, а поставщик приложения – время, чтобы развернуть дополнительные или освободить используемые ВМ.

3*) Проектировщик производительности и прогнозирования загрузки системы*, который ведет аналитическое моделирование и обозначает ожидаемый уровень запросов. Функции ППЗС заключаются в вычислении количества экземпляров виртуальных приложений, соответствующих задачам QoS. Проектировщик строит модель системы как сети массового обслуживания, где параметры базируются

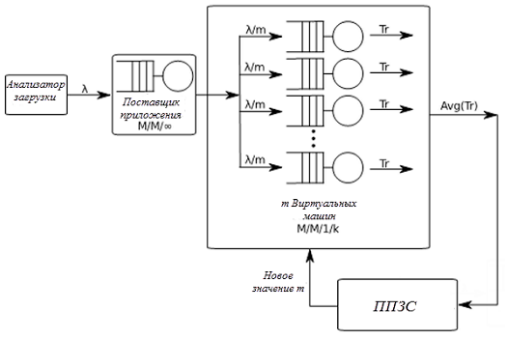

Рис. 2. Модель обработки очередей запросов ЦОД.

Pic. 2. The model of requests processing in the data processing centers.

на информации, получаемой при формализации результатов мониторинга «Облака» [6, с. 298]. Система массового обслуживания, обрабатываемая ППЗС, представлена на рис. 2. Конечные пользователи в модели изображены как генераторы запросов, в то время как поставщик и экземпляры приложений – в виде обрабатывающих узлов для идущих запросов.

Поставщик – это узел, имеющий $M/M/\infty$ очередей запросов [2, с.68]. С другой стороны, каждый экземпляр виртуального приложения может рассчитывать на $M/M/1/k$ очередей [2, с. 38].

В разработанной модели предполагается, что экземпляры приложения получают одинаковые программные и аппаратные настройки, за счет этого они обеспечивают равные объемы производительности. Эти параметры задаются на этапе конфигурирования ВМ. Виртуальные машины с различными возможностями могут также разворачиваться в системе.

Когда анализатор рабочей загрузки обновляет значение оценки уровня прибытия, то ППЗС проверяет, достаточно ли текущее количество запущенных экземпляров виртуальных приложений для выполнения QoS. Фиксируемое при этом время обслуживания учитывается при прогнозировании общего времени отклика на запрос, уровня отказа в обработке, объема требуемых ресурсов и максимального количества ВМ. Если время отклика или уровень отказа оцениваются как понижающие QoS, либо прогнозируется уровень общего объема ресурсов ниже минимального порогового значения, то количество экземпляров ВМ, обслуживающих приложения, обновляется. [6, с. 299].

Совместно три компонента рассматриваемого адаптивного механизма предоставления ресурсов – поставщик приложений, анализатор рабочей загрузки и ППЗС – способны динамично адаптировать количество экземпляров виртуальных приложений для поддержания нужного уровня QoS в ситуациях, когда значения времени обработки и уровня отказа в обслуживании будут находиться ниже оговоренного предела.

ИМИТАЦИОННЫЕ МОДЕЛИ ОБЛАЧНОЙ СРЕДЫ

Для построения имитационных моделей облачных систем использовался пакет моделирования CloudSlim [7, с. 24].

В рамках исследования базовой стала модель ЦОД, содержащего 1000 хостов, каждый хост имеет по два четырехъядерных центральных процессора (далее – ЦП) и 16 Гб оперативной памяти. Элементами имитационной модели являются, кроме хостов ЦОД, экземпляры приложений и адаптивный механизм предоставления ресурсов, а также блок, генерирующий запросы, которые направляются группой пользователей.

Виртуальные машины, выделяемые приложениям, должны содержать ядро ЦП и два Гб оперативной памяти. Для распределения физических ресурсов ЦОД применялась простая политика балансировки загрузки, при которой новые экземпляры ВМ создаются на хосте с наименьшим числом запущенных экземпляров виртуальных приложений. При запуске экземпляры ВМ используют простаивающие ядра физического хоста.

ВМ-часы – суммарное время работы каждого экземпляра приложения от его запуска до остановки.

В процессе проведения экспериментов тестировались два сценария использования облачных сервисов. Для каждого из них берутся следующие выходные показатели: 1) минимальное и максимальное числа виртуальных приложений, запущенных в единицу времени; 2) уровень отказа в обслуживании и объемы ресурсов ЦОД; 3) ВМ-часы; 4) среднее время отклика системы и стандартное отклонение.

Имитация каждого сценария выполнялась 10 раз, после чего было подсчитано среднее значение каждого выходного параметра.

В первом сценарии моделируется рабочая загрузка web-сервиса с большим количеством запросов, обработка которых требует своего объема производительной мощности от экземпляров приложения (например, поиск web-страниц). Данный сценарий обозначается как «Web».

Модель «Web» G_s формируется по результатам анализа загрузки доступа к web-странице, опубликованным в работе [3]. Изменение уровня входящих запросов здесь зависит от дня недели или времени суток. Имитация была запущена как для ЦОД с адаптивным механизмом предоставления ресурсов, так и ЦОД со статическими механизмами предоставления, состоящими из 50, 75, 100, 125 и 500 экземпляров виртуальных приложений.

Каждый запрос требует 100 мс для обработки на сервере. С учетом их неоднородности время обработки запросов добавлялось равномерно – генерируемые значения увеличивались в диапазоне от 0 до 10%. Максимально допустимое время отклика было установлено 250 мс, а максимальный отказ обработки сводился к 0%, т. е. системе следовало обрабатывать все запросы. Минимальное использование ресурсов было на уровне 80%. Имитация эксперимента заключалась в сборе статистики за одну неделю запросов в ЦОД, начиная с 00.00 часов понедельника.

Во втором сценарии моделируется рабочая загрузка приложений, с небольшим количеством запросов, обработка которых нуждается в значительном объеме производительной мощности от экземпляра приложения. Такой сценарий получил название «Научный». В нем T_r является большим, чем в сценарии «Web», а сам эксперимент демонстрирует особенности предоставления виртуальных ресурсов для научных приложений (например, обработка изображений и моделирование свертывания белка).

Рабочая загрузка «Научной модели» заключается в направлении поставщику услуг запросов на выполнение вычислительно сложных задач. Поступление запросов моделируется, основываясь на результатах анализа рабочей загрузки для GRID-приложения «Bag-of-Task» (далее – BoT), описанного в [4].

Для анализа рабочей загрузки имитация была запущена как для ЦОД с адаптивным механизмом предоставления ресурсов, так и ЦОД со статическими механизмами, состо-

Таблица 1/Table 1

Результаты рабочей загрузки имитационной модели «Web»
Results of workload simulation for model «Web»

Механизм/Параметр *Mechanism/parameter*	**Мин. ВМ** *min.VM*	**Макс. ВМ** *max.VM*	**Отказ в обслуживании, %** *service denial,%*	**Общее использ. ресурсов, %** *general resource use,%*	**ВМ-часы** *VM-hours*	**Среднее время отклика, мс** *average response time, ms*
Адаптивный/*adaptive*	55	153	0	82	18900	108
Статический 50 ВМ/*static 50 VM*	50	50	56	76	8400	106
Статический 75 ВМ/ *static 75 VM*	75	75	40	70	12600	106
Статический 100 ВМ/ *static 100 VM*	100	100	22	69	16800	105
Статический 125 ВМ/ *static 125 VM*	125	125	2	69	21000	104
Статический 150 ВМ/ *static 150 VM*	150	150	0	59	25200	105
Требования QoS/ *Requirements of QoS*	-	-	0	≥**80**	-	≤**250**

*VM-virtual machine

Таблица 2/Table 2

Результаты имитации рабочей загрузки «Научной модели»
Results of workload simulation for «scientific model»

Механизм/Параметр *Mechanism/parameter*	**Мин. ВМ** *min.VM*	**Макс. ВМ** *max.VM*	**Отказ в обслуживании, %** *service denial,%*	**Общее использ. ресурсов, %** *general resource use,%*	**ВМ-часы** VM-hours	**Среднее время отклика, с** *average response time, sec*
Адаптивный/ *adaptive*	13	80	0	78	980	320
Статический 15 ВМ/ *static 15 VM*	15	15	69	99	360	490
Статический 30 ВМ/ *static 30 VM*	30	30	51	80	720	500
Статический 45 ВМ/ *static 45 VM*	45	45	34	73	1080	490
Статический 60 ВМ/ *static 60 VM*	60	60	16	64	1440	480
Статический 75 ВМ/ *static 75 VM*	75	75	0	42	1800	325
Требования QoS *Requirements of QoS*	-	-	0	≥**80**	-	≤**700**

*VM-virtual machine

ящими из 15, 30, 45, 60 и 75 экземпляров приложений.

Каждый запрос требует 300 секунд для обработки на сервере. С учетом эффекта неоднородности время обработки запросов было добавлено равномерно – генерируемые значения увеличивались в диапазоне от 0 до 10%. Максимальное время отклика 700 с, а максимальный отказ обработки – 0%. Минимальное использование ресурсов было установлено на уровне 80%. Имитация эксперимента заключалась в одном дне обработки запросов, направляемых в ЦОД, начиная с 00.00 часов.

Во всех экспериментах применялось одинаковое правило управления допуском: запросы либо обслуживались вовремя, либо отклонялись. Выходные параметры накапливались одинаково как в процессе имитационного прогона статического, так и адаптивного механизмов.

РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

В таблице 1 представлены результаты модели «Web». В среднем каждый имитационный прогон эксперимента генерирует 500,12 миллиона запросов за одну моделируемую неделю работы.

Число экземпляров виртуальных приложений в ЦОД с адаптивным механизмом варьируется от 55 до 153; количество экземпляров ВМ, создаваемых адаптивным механизмом в пики нагрузки (153), превышает реально нужное, так как в статическом ЦОД со 150 ВМ уровень отказов обработки был тоже 0%; количество ВМ-часов, потребовавшихся адаптивному механизму за одну неделю предоставления ресурсов с уровнем отказа, равным 0%, эквивалентно поддержанию 113 экземпляров виртуальных приложений в активном режиме, в то время как уровень отказа статического механизма со 125 экземплярами постоянно работающих ВМ оказался равен 2%, что не удовлетворяет требованиям QoS.

Чтобы статический механизм имел значения показателей среднего времени обработки запроса и уровня отказа в обслуживании, тождественные соответствующим параметрам адаптивного механизма, ему необходимо 150 постоянно работающих экземпляров ВМ. Но в данном случае значение показателя общего использования ресурсов статического механизма составляет 60%, а адаптивного механизма, при выполнении требований QoS – 82%, что снижает потенциал системы на 22% (в значении ВМ-часов).

В таблице 2 представлены результаты «Научной модели». В среднем в каждой имитации эксперимента генерируется 8286 запросов за один день.

Число экземпляров виртуальных приложений в ЦОД с адаптивным механизмом варьируется от 13 ВМ в моменты снижения нагрузки и до 80 ВМ в периоды ее повышения; количество ВМ-часов, потребовавшихся адаптивному механизму за один день предоставления ресурсов с уровнем отказа, равным 0%, эквивалентно поддержанию 41 экземпляра виртуальных приложений в активном режиме, в то время как уровень отказа статического механизма, содержащего 45 экземпляров постоянно работающих ВМ, оказался равен 34%, что не удовлетворяет требованиям QoS.

Чтобы статический механизм имел значения показателей среднего времени обработки запроса и уровня отказа в обслуживании, тождественные соответствующим параметрам адаптивного механизма, ему необходимо 75 постоянно работающих экземпляров ВМ. Но в данном случае значение показателя общего использования ресурсов статического механизма составляет 42%, а показатель общего использования ресурсов адаптивного механизма, при выполнении требований QoS – 78%, что снижает системы на 36% (в значении ВМ-часов). Из таблицы 2 видно, что использование вычислительных ресурсов ЦОД адаптивного механизма было немного ниже нужного уровня (78% по сравнению с установленными 80%).

Из таблиц 1 и 2 следует, что в большинстве статических механизмов с различным количеством постоянно работающих ВМ отмечаются высокий уровень отказа обработки запросов и низкий уровень использования ресурсов. Этот факт объясняется тем, что часть экземпляров ВМ, запущенных для предоставления ресурсов, простаивает в периоды снижения нагрузки. Причем общего количества экземпляров не хватает для обработки запросов, поступающих в периоды повышенной нагрузки.

Компонент управления допуском адаптивного механизма, применяемый в рассмотренных сценариях, удачно предотвратил нарушение требований QoS.

ВЫВОДЫ

Архитектура адаптивного механизма предоставления ресурсов предполагает запуск приложений в системе на виртуальных машинах, которые расположены на множестве объединенных серверов, позволяющих совместное использование ресурсов в облачной инфраструктуре.

Результаты моделирования, проводимого на основе параметров рабочей загрузки, демонстрируют, что адаптивный механизм, который отслеживает изменения интенсивности загрузки за определенный период времени, выделяет необходимое количество ресурсов для достижения требуемого уровня обеспечения качества обслуживания и эффективного распределения ресурсов в облачной системе.

В имитационных моделях адаптивный механизм исключает отказ обработки запросов посредством динамического увеличения числа экземпляров виртуальных приложений

в периоды повышенной нагрузки и снижение в периоды ее спада.

Из этого можно сделать вывод, что использование адаптивного механизма предоставления ресурсов является эффективным при организации доступа к web-ресурсу, однако в системе распределенной обработки математических моделей наблюдается избыточное выделение ресурсов.

Наряду с проведенной работой интерес для дальнейшего исследования представляет разработка управляющего правила перерасчета количества ресурсов ВМ при решении вычислительных задач большой размерности.

Вместе с тем важно и внедрение алгоритмов приоритизации в элементе управления допуском, реализованном в адаптивном механизме.

ЛИТЕРАТУРА

1. Бородакий В. Ю., Окороченко Г. Е. Анализ средств имитационного моделирования распределенных информационных систем//Научная сессия МИФИ-2007. Сборник научных трудов. В 15 томах. – Т. 12. Компьютерные системы и технологии. – М.: МИФИ, 2007. – С. 129–130.

2. Вишневский В. М. Теоретические основы проектирования компьютерных сетей. – М.: Техносфера, 2003. – 506 с.

3. Гнеденко Б. В., Коваленко И. Н. Введение в теорию массового обслуживания. – М.: Наука, 1966. – 432 с.

4. A. Iosup, O. Sonmez, S. Anoep, and D. Epema, The performance of bags-of-tasks in large-scale distributed systems, in Proceedings of the 17th International Symposium on High Performance Distributed Computing (HPDC'08), 2008;

5. G. Urdaneta, G. Pierre, and M. van Steen, Wikipedia workload analysis for decentralized hosting, Computer Networks, vol. 53, no. 11, 2009. P. 1830–1845;

6. R. N. Calheiros, Rajiv Ranjany, and Rajkumar Buyya Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments. 2011 International Conference on Parallel Processing, 2011. P. 295–304;

7. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Software: Practice and Experience, 41 (1), 2011. P. 23–50;

8. Tanveer Ahmed, Yogendra Singh Analytic Study Of Load Balancing Techniques Using Tool Cloud Analyst. International Journal of Engineering Research and Applications (IJERA), 2012. P. 1027–1030;

9. Y. C. Lee and A. Zomaya, «Rescheduling for reliable job completion with the support of clouds,» Future Generation Computer Systems, vol. 26, no. 8, 2010. P. 1192–1199;

10. Folding@home. [Электронный ресурс]. 2013. Дата обновления: 22.10.2013. URL: http://folding.stanford.edu/home/papers (дата обращения: 22.10.2013).

OPTIMIZATION OF SUPPORT OF IT-RESOURCES OF RAILWAYS

Ignatov, Nickolay A. – *Ph. D. student at the department of computer systems and networks of Moscow State University of Railway Engineering (MIIT), Moscow, Russia*

ABSTRACT

In this article the author focuses on peculiarities of providing virtual resources, used in cloud computing systems for guaranteed quality of service, taking into account the requirements of QoS. The article contains the description of adaptive mechanism and comparative analysis of static and adaptive mechanisms to provide resources through simulation models. Moreover it covers such computing figures for models as the average time for request processing, the level of service denial, the significance of general application of system resources with different inbound parameters.

ENGLISH SUMMARY

Background. *Cloud computing is one of the most modern technology, which provides a variety of options for working with data as a network service. Users are released from worries about commissioning or systems administration. However, there are significant problems with the calculation and allocation of resources, primarily, in cloud computing systems. If they are resolved, the work of applications will be able more effective, with reduced financial costs, reallocation of unused resources and increase in productivity in moments of high load.*

Objective. *The author examines the mechanism of automatic adaptation to changes in workload related to the tasks that are performed by applications in the cloud environment.*

Methods. *The author uses specific methods of analysis of IT-systems.*

Results. *System based on cloud computing combines data processing centers (hereinafter – DPC) as a network of virtual IaaS (computing servers, databases, networks) and PaaS (stabilization mechanisms of load (download) and auto-scaling), so that providers are able to open applications (SaaS) from anywhere in the world with a level of costs, determined by the requirements of QoS.*

Copies of the applications are examples of SaaS programs, which may belong to small or medium-sized enterprises, as well as public organizations, using applications running through the clouds. In this mechanism the application itself and platform are provided by one organization, and resources based on cloud technologies are used by different organizations.

Calculation of resources needed for the application consists of a set of independent tasks that can be modeled as service requests sent by end-users for copies of virtual applications.

An adaptive mechanism (its architecture is shown in Pic.1) was developed in order to meet the QoS requirements of the system within the changing nature of the workload, as well as the difficulty of forecasting the volume of virtual resources specific to cloud technology.

It consists of three levels. The service provider administers various software components of the mechanism architecture. SaaS level of the mechanism