

УДК 004.832.32

DOI: 10.21869/2223-1560-2019-23-2-153-173

Исследование и организация приоритетных режимов в сетевой распределенной вычислительной системе с архитектурой в виде облачного сервиса

Мустафа Садек Джафар ✉

ФГБОУ ВО «Пензенский государственный университет», Россия, 440026, Пенза, ул. Красная, 40

✉ e-mail: mustafajaafar221087@gmail.com

Резюме

Актуальность и цель. Объектом исследования является сетевой облачный сервис, построенный на основе реплицированной базы данных. Данные в распределенных вычислительных системах реплицируются в целях обеспечения надежности их хранения, для облегчения доступа к ним, а также для повышения производительности системы хранения данных. В этой связи актуальной и недостаточно полно исследованной является проблема анализа эффективности обработки запросов к реплицированным базам данных в сетевой облачной среде, и, в частности, проблема организации приоритетных очередей запросов на обновление копий баз данных (*update requests*) и запросов на поиск и чтение информации в базах данных (*query requests*). Целью настоящей работы является исследование и организация приоритетных режимов в сетевой распределенной вычислительной системе с архитектурой в виде облачного сервиса.

Материалы и методы. Исследование проведено на основе поведенческих моделей двух видов: модели на основе сетей Петри для описания и проверки правильности функционирования распределенной вычислительной системы с реплицированными базами данных, представленными в виде пула единиц ресурса с несколькими единицами, и модели на основе языка имитационного моделирования GPSS для приближенной оценки времени пребывания запросов каждого типа в очередях в зависимости от приоритета запросов.

Результаты. На основе двух методов имитационного моделирования проведен анализ функционирования облачной системы с репликами баз данных, в которой взаимодействуют две распределенные облачные вычислительные системы: MANET Cloud на основе беспроводной сети и Internet Cloud на основе сети Интернет. В совокупности базы данных являются основой облачного сервиса DBaaS – Data Bases as a Service on Demand (базы данных как сервис, организуемый по требованию пользователя). Для исследования данной системы построены модели двух классов. Модель на основе сети Петри предназначена для проверки моделируемого распределенного приложения на правильность функционирования. Обсуждаются решения по отображению сетей Петри на архитектуру компьютерных сетей. Имитационная статистическая модель используется для сравнения приоритетного и беспriorитетного режимов обслуживания *query*- и *update*-запросов по критерию среднего времени пребывания запросов в очередях.

Выводы. Модели системы на основе сетей Петри прошли проверку, которая показала их живость и безопасность, что позволяет от моделей переходить к построению формализованных спецификаций для сетевых приложений для сетевых облачных сервисов в распределенных вычислительных системах с реплицированными базами данных. Исследование GPSS-модели показало, что в случае приоритетного обслуживания *update*-запросов время ожидания для них сокращается примерно в 2 – 4 раза по сравнению с *query*-запросами в зависимости от интенсивности поступления *query*-запросов. При беспriorитетном режиме условия обслуживания *update*-запросов ухудшаются и время ожидания в очереди для них увеличивается примерно в 2 – 6 раз по сравнению с *query*-запросами в зависимости от интенсивности поступления *query*-запросов.

Ключевые слова: распределенные вычислительные системы; реплицированные базы данных; облачные сервисы; приоритетное обслуживание запросов; модели на основе сетей Петри; GPSS-модель.

© Мустафа Садек Джафар, 2019

Благодарности: Автор выражает глубокую признательность своему научному руководителю Сергею Александровичу Зинкину, доктору технических наук, профессору кафедры «Вычислительная техника» Пензенского государственного университета за полезные обсуждения и консультации при подготовке данной работы.

Конфликт интересов: Автор декларирует отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Для цитирования: Мустафа Садек Джафар. Исследование и организация приоритетных режимов в сетевой распределенной вычислительной системе с архитектурой в виде облачного сервиса // Известия Юго-Западного государственного университета. 2019. Т. 23, № 2. С. 153-173. DOI: 10.21869/2223-1560-2019-23-2-153-173.

UDC 004.832.32

DOI: 10.21869/2223-1560-2019-23-2-153-173

Research and Organization of Priority Modes in a Network Distributed Computing System with Cloud Service Architecture

Mustafa Sadeq Jaafar ✉

Penza State University, 40, Krasnaya str., Penza, 440026, Russian Federation

✉ e-mail: mustafajaafar221087@gmail.com

Abstract

Purpose of research. The object of the study is a network cloud service built on the basis of a replicated database. Data in distributed computing systems are replicated in order to ensure the reliability of their storage, facilitate access to data as well as to improve the storage system performance. In this regard, the problem of analyzing the effectiveness of processing the queries to replicated databases in a network-based cloud environment, and, in particular, the problem of organizing priority query queues for updating database copies (update requests) and for searching and reading information in databases (query-requests). The purpose of this work is to study and organize priority modes in a network distributed computing system with cloud service architecture.

Methods. The study was conducted on the basis of two types of behavioural patterns: models based on Petri nets to describe and verify the functioning of a distributed computing system with replicated databases represented as a pool of resource units with several units, and models based on the GPSS simulation language for possible evaluation of passage of query time of each type in queues depending on the priority of queries.

Results. Based on two simulation methods, the operation of a cloud system with database replicas was analyzed. In this system two distributed cloud computing systems interact: MANET Cloud based on a wireless network and Internet Cloud based on the Internet. These databases together are the basis of the DBaaS (Data Bases as a Service on Demand) cloud service (databases as a service organized at user's query). To study this system the models of two classes were developed. The model based on Petri nets is designed to test the simulated distributed application for proper functioning. The decisions on the mapping of Petri nets on the architecture of computer networks are discussed. The simulation statistical model is used to compare the priority and non-priority maintenance modes of query- and update-requests by the criterion of average passage of time of queries in queues.

Conclusion. System models based on Petri nets were tested, which showed their liveness and security, which makes it possible to move from models to building formalized specifications for network applications for network cloud services in distributed computing systems with replicated databases. The study of GPSS-model showed that in the case of priority service of update-requests, the time of passage for them is reduced by about 2 to 4 times compared with query-requests, depending on the intensity of the query-requests. In the non-priority mode, the serving conditions for update-queries deteriorate and the time of passage in the queue for them increases by about 2 to 6 times as compared with query-requests depending on the intensity of the query-requests.

Keywords: *distributed computing systems; replicated database; Cloud Service; priority queries service; models based on Petri nets; GPSS-model.*

Acknowledgements: *The author expresses deep gratitude to his scientific supervisor Sergey Alexandrovich Zinkin, doctor of technical Sciences, Professor of the Department of "Computer engineering" of Penza state University for useful discussions and consultations in the preparation of this work.*

Conflict of interest. *The Author declare the absence of obvious and potential conflicts of interest related to the publication of this article.*

For citation: Mustafa Sadeq Jaafar Research and Organization of Priority Modes in a Network Distributed Computing System with Cloud Service Architecture. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2019, 23(2): 153-173 (In Russ.). DOI: 10.21869/2223-1560-2019-23-2-153-173.

Введение

Технологии создания сетевого промежуточного программного обеспечения класса *middleware* во многом пересекаются с другими известными в современной международной практике технологиями, такими как Cloud Computing [1 – 4] (облачные вычисления), Grid Computing [5 – 8] (грид-вычисления), Utility Computing [9 – 12] (вычисления как услуги), Services Computing [13 – 15] (сервис-ориентированные вычисления), Peer-to-Peer Computing [16, 17] (пиринговые системы) и Distributed Computing [18 – 21] (распределенные вычисления) в целом. Классификация, учитывающая современные тенденции развития распределенных систем и вычислений, дана в работе [22].

Во многих распределенных вычислительных системах используются крупномасштабные базы данных. Одной из важнейших проблем в подобных системах является проблема распределения данных и, в частности, проблема хранения копий данных (так называемых “реплик”). Данные в распределенных вычислительных системах реплициру-

ются в целях обеспечения надежности их хранения, для облегчения доступа к ним, а также для повышения производительности системы хранения данных. Существует несколько способов создания и согласованного использования реплик баз данных: синхронные и асинхронные, статические и динамические, репликация на уровне транзакций в сетях и другие [23 – 29]. В этой связи актуальной и недостаточно полно исследованной является проблема анализа эффективности обработки запросов к реплицированным базам данных в сетевой облачной среде, и, в частности, проблема организации приоритетных очередей запросов на обновление копий баз данных (Update requests) и запросов на поиск и чтение информации в базах данных (Query requests).

Целью настоящей работы является исследование и организация приоритетных режимов в сетевой распределенной вычислительной системе с архитектурой в виде облачного сервиса. Исследование проведено при помощи поведенческих моделей двух видов: модели на основе сетей Петри для описа-

ния и проверки правильности функционирования распределенной вычислительной системы с реплицированными базами данных, представленными в виде пула единиц ресурса с несколькими единицами и модели для приближенной оценки времени пребывания запросов каждого типа в очередях в зависимости от приоритета запросов.

Материалы и методы решения задачи

В работе [22] приведена общая схема подключения клиентской облачной сети MANET Cloud к ресурсам исполнительной облачной среды Internet Cloud, которую иллюстрирует рисунок 1. Сети MANET [30 – 34] (Mobile Ad

hoc NETworks) представляют собой беспроводные самоорганизующиеся сети со случайными мобильными абонентами, реализующие полностью децентрализованное управление при отсутствии базовых станций или опорных узлов. На рисунке 1 представлена распределенная вычислительная система, в которой взаимодействуют две типовые распределенные облачные вычислительные системы: MANET Cloud и Internet Cloud.

Принято, что клиент-серверная архитектура доступа клиентов к серверам реализуется через составную TCP/IP сеть [35].

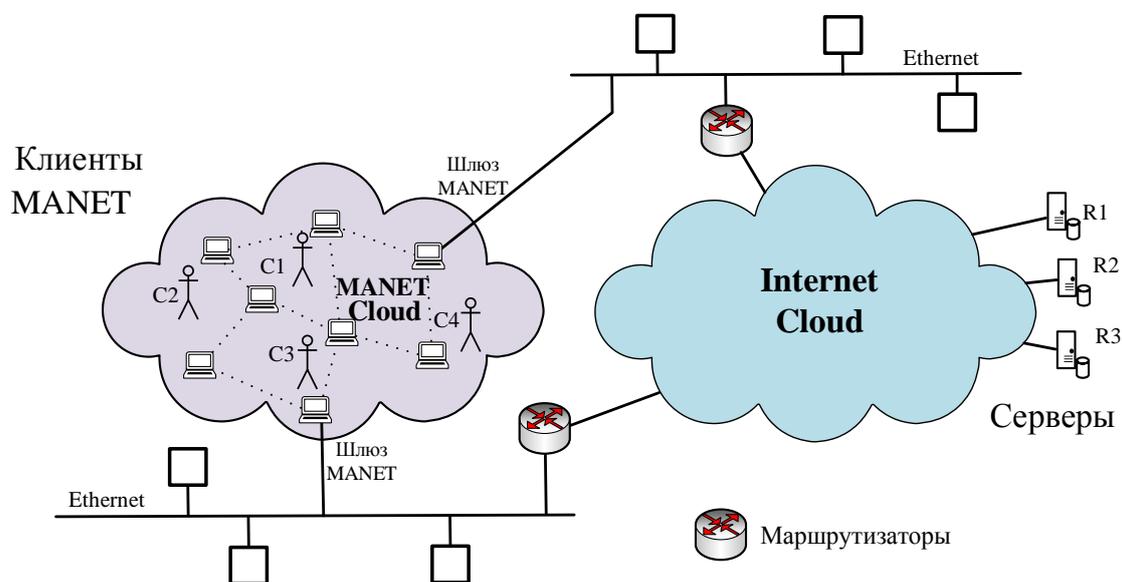


Рис. 1. Взаимодействие двух распределенных облачных вычислительных систем через локальные сети типа Ethernet

Здесь символами C_1, C_2, \dots, C_n обозначены клиентские приложения сети MANET Cloud, а символами R_1, R_2, \dots, R_m – ресурсы (копии баз данных), размещенные на серверах сети Internet

Cloud. Шлюзы и локальные сети типа Ethernet обеспечивают связь между двумя облачными подсистемами.

Пример развертывания ресурсов, например, серверов баз данных, в об-

лачной среде Internet Cloud, реализованной на основе TCP/IP сети, представлен на рисунке 2. На нем представлены реплики баз данных (Databases),

пользовательские станции (Users), серверы (Servers), коммутаторы (Switches) и маршрутизаторы (Routers).

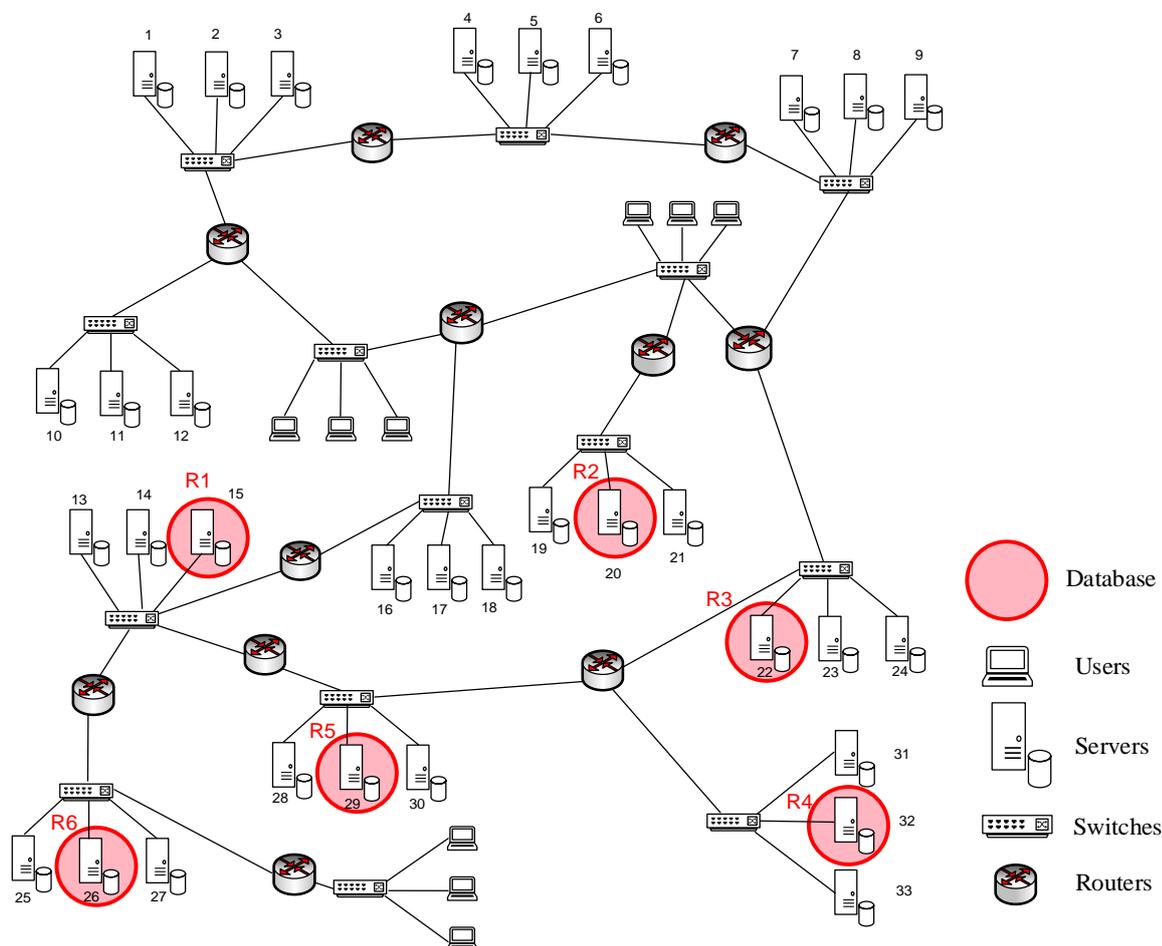


Рис. 2. Пример развертывания мультисерверной системы баз данных в TCP/IP сети

Анализируется пример организации доступа к распределенной мультисерверной системе баз данных с репликацией. Упрощенная схема в виде системы массового обслуживания, моделирующей доступ к базам данных в сети, представлена на рисунке 3. Копии баз данных при этом размещаются на разных узлах, а запросы делятся на два типа – запросы на поиск и чтение данных (Query requests), образующие очередь Q_{One} запросов к одной из баз данных и

запросы на модификацию данных (Update requests), образующие очередь Q_{All} на обращение ко всем базам данных. Запросы формируются в пользовательской сети MANET Cloud (User network) при помощи программных агентов (Agents). Ответы (Query responses и Update responses) на запросы обоих типов возвращаются пользователям.

В совокупности базы данных являются основой облачного сервиса DBaaS – Data Bases as a Service on Demand (базы

данных как сервис, организуемый по требованию пользователя). Система типа DBaaSOD представляет собой один из вариантов реализации концепции построения сетевой облачной системы типа NCaaSOD – Network Computing as a Service On Demand (компьютерная сетевая архитектура как облачный сервис по запросу пользователя). Реализация концепции “Service on Demand” в общем виде предложена в работе [36]. Для исследования данной системы целесообразно построить модели двух классов. Модель на основе сети Петри предназначена для проверки моделируемого распределенного приложения на правильность функционирования, а также, возможно, для получения на ее

основе формализованных исполнимых логико-алгебраических спецификаций. Вопросы отображения сетей Петри на архитектуру компьютерных сетей, в том числе программные реализации таких отображений на языке Java, описаны, например, в работах [37, 38].

Имитационная статистическая модель, построенная, например, при помощи известной системы GPSS World [39], может быть использована для оценки некоторых характеристик моделируемой системы, например, для сравнения приоритетного и беспriorитетного режимов функционирования по критерию среднего времени пребывания запросов в очередях Q_{All} и Q_{One} .

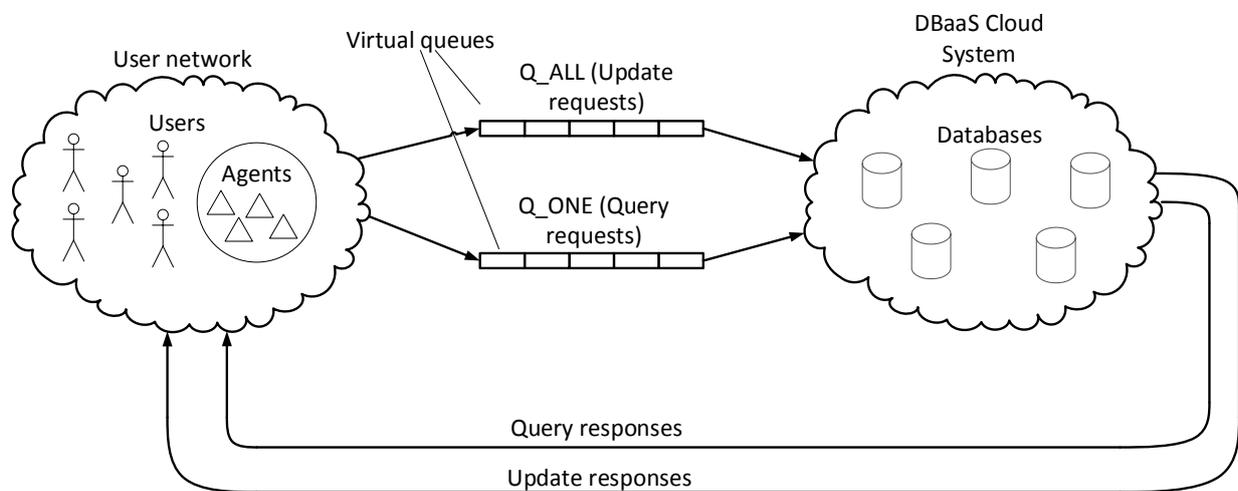


Рис. 3. Облачная система с сервисом DBaaSOD (DataBases as a Service on Demand)

Модель организации обслуживания очередей Q_{All} и Q_{One} для доступа к ресурсу R с несколькими единицами при беспriorитетном режиме обслуживания дана на рисунке 4. Здесь позициями a_1 и b_1 представлены места для ожида-

ний запросов в очереди Q_{All} , а позициями c_1 , d_1 и e_1 – места для ожидания в очереди Q_{One} . Каждый запрос представлен процессом (меткой), обращающимся к пулу единиц r_1 , r_2 и r_3 ресурса R , причем каждому из update-запросов в

позициях a_1 и b_1 требуются все три единицы r_1, r_2 и r_3 ресурса R , а запросам в позициях c_1, d_1 и e_1 требуется по одной единице ресурса R . Здесь метки в позициях r_1, r_2 и r_3 представляют собой реплики базы данных. Некоторые дуги в данной сети Петри являются ингибиторными (сдерживающими). Они оканчиваются не стрелками, а кружками и разрешают срабатывание переходов при отсутствии меток в исходных позициях. Логика срабатывания переходов здесь

проста: переход срабатывает в том случае, если в его входных позициях, связанных с данным переходом простыми дугами, есть по одной метке, а во входных позициях, связанных ингибиторными дугами, метки отсутствуют.

Сетевая модель организации очередей Q_{All} и Q_{One} для доступа к ресурсу R с несколькими единицами при приоритетном режиме обслуживания приведена на рисунке 5.

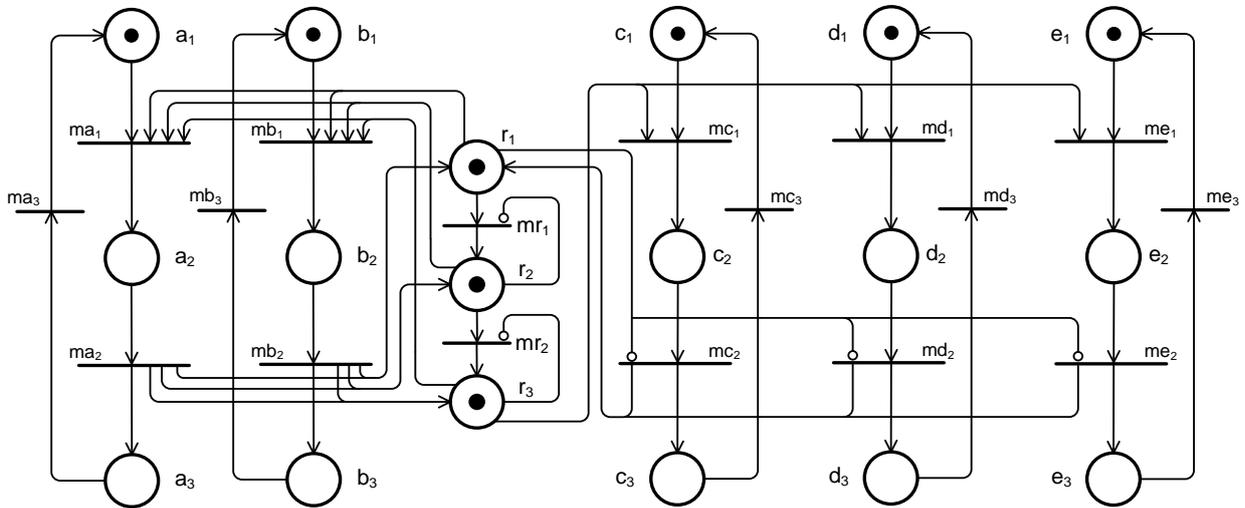


Рис. 4. Модель организации очередей Q_{All} и Q_{One} для доступа к ресурсу с несколькими единицами (бесприоритетный режим)

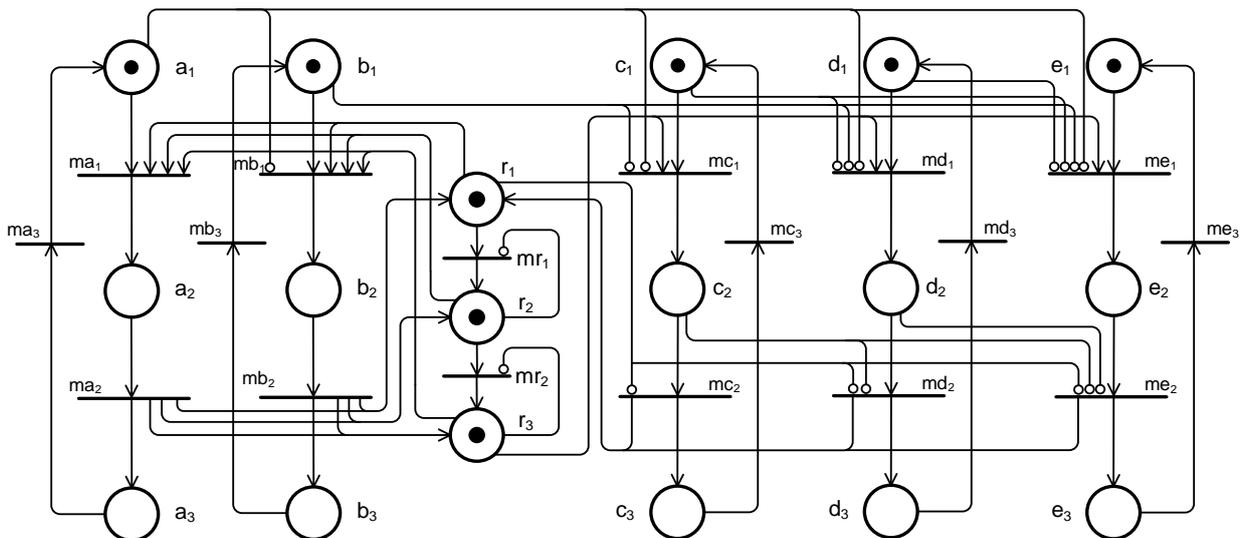


Рис. 5. Модель организации очередей Q_{All} и Q_{One} для доступа к ресурсу с несколькими единицами (приоритетный режим обслуживания запросов очереди Q_{All} по сравнению с запросами из очереди Q_{One})

Данная модель отличается от предыдущей модели наличием дополнительных ингибиторных дуг, запрещающих срабатывание переходов mc_1 , md_1 и me_1 для запросов с низшими приоритетами при наличии запросов с высшими приоритетами, то есть при наличии меток в позициях a_1 или b_1 .

Обе модели, представленные на рисунках 4 и 5, прошли проверку при помощи известной в международной практике системы PIPE [40] (Platform Independent Petri net Editor – независимый от платформы редактор и интерпретатор сетей Петри). Эта система обладает широкими возможностями по исследованию сетей Петри распространенных классов, включая анализ графа достижимых состояний и статистическое имитационное моделирование. Проверка показала живость и безопасность предложенных сетей Петри, что позволяет от моделей переходить к построению формализованных спецификаций для сетевых приложений.

Результаты и их обсуждение

GPSS-модель, представленная на рисунке 6, состоит из двух частей – операторы 1 – 14 моделируют прохождение потока запросов типа “update”, а операторы 15 – 28 – запросов типа “query”. В процессе моделирования производится сбор статистических данных об

очередях Q_ALL для запросов типа “update” и Q_ONE для запросов типа “query”.

Основной ресурс – совокупность баз данных, имитируется в модели при помощи многоканального устройства RESURS (объекта типа “память” в терминах системы моделирования GPSS [39]). Каждая реплика базы данных представлена отдельной единицей (“каналом”) этого устройства.

Оператор 1 генерирует поток *update*-запросов. Оператор 2 имитирует задержку. Оператор 3 собирает статистические данные об очереди Q_ALL *update*-запросов. Оператор 4 позволяет *update*-запросу занять одноканальное устройство FOC_ALL, имитирующее в модели компьютер провайдера облачного сервиса. Оператор 5 при необходимости может понизить приоритет *update*-запроса по сравнению с *query*-запросом путем блокировки дальнейшего продвижения *update*-запросов при наличии *query*-запросов в очереди Q_ONE. Установкой символа звездочки перед оператором это ограничение снимается. Оператор 7 проверяет незанятость всех единиц ресурса RESURS. Оператор 8 позволяет *update*-запросу занять сразу все единицы этого ресурса. Тем самым в модели имитируется занятие всех реплик баз данных для выполнения одинаковых модификаций.

```

*Модель системы с сервисом DBaaSOD
* Запросы типа ALL
1      GENERATE (EXPONENTIAL(1,0,200))
2      ADVANCE
3      QUEUE Q_ALL
4      SEIZE      FOC_ALL
5*     TEST E Q$Q_ONE,0
6      ADVANCE 10
7      GATE SE RESURS
8      ENTER RESURS,8
9      RELEASE  FOC_ALL
10     DEPART Q_ALL
11     ADVANCE 100,50
12     LEAVE RESURS,8
13     TABULATE TAB_ALL
14     TERMINATE 1
* Запросы типа ONE
15     GENERATE (EXPONENTIAL(1,0,125))
16     ADVANCE
17     QUEUE Q_ONE
18     SEIZE      FOC_ONE
19     TEST E Q$Q_ALL,0
20     ADVANCE 10
21     GATE SNF RESURS
22     ENTER RESURS,1
23     RELEASE  FOC_ONE
24     DEPART Q_ONE
25     ADVANCE 100,50
26     LEAVE RESURS,1
27     TABULATE TAB_ONE
28     TERMINATE 1
*
29 RESURS STORAGE 8
30 TAB_ALL  TABLE  M1,0,100,30
31 TAB_ONE  TABLE  M1,0,100,30
32          START   100000

```

Рис. 6. Имитационная GPSS-модель системы типа DBaaSOD

Оператор 9 освобождает устройство FOC_ALL. Оператор 10 собирает статистические данные об очереди Q_ALL при выходе из нее *update*-запроса. Оператор 11 задерживает запрос на время имитации выполнения операции модификации данных. Оператор 12 освобождает все единицы ресурса RESURS. Оператор 13 формирует данные для гистограммы времени обслуживания запроса. Оператор 14 аннулирует запрос.

Вторая часть модели построена таким же образом для *query*-запросов, только здесь каждому запросу для имитации его выполнения требуется лишь одна единица ресурса RESURS.

Приоритет для *update*-запросов задается следующим образом:

```

5*     TEST E Q$Q_ONE,0
. . .
19     TEST E Q$Q_ALL,0

```

а бесприоритетному режиму соответствуют следующие строки в модели:

```

5*      TEST E Q$Q_ONE,0
. . .
19*     TEST E Q$Q_ALL,0

```

то есть приоритет в модели задается простым исключением условия блокировки запроса (звездочка в первой позиции строки превращает строку оператора в строку комментария; номер строки не является частью оператора).

В строке 29 задано число единиц ресурса RESURS – число реплик баз данных в моделируемой системе (то есть число единиц в многоканальном устройстве RESURS). В строках 30 и 31 заданы параметры для гистограмм времени обслуживания запросов соответствующих типов. Строкой 32 задается число пропускаемых через систему запросов при прогоне модели.

Исследование модели проведено для двух режимов управления потоков запросов – при высшем приоритете у *update*-запросов в очереди Q_ALL и при бесприоритетном режиме. Входная нагрузка на систему для *update*-запросов определяется в операторе 1 GENERATE (EXPONENTIAL(1,0,200)), где задается экспоненциальное распределение интервалов между *update*-запросами. Среднее значение интервала задано третьим операндом функции EXPONENTIAL. В данном случае это значение равно $T_U = 200$ мс (интенсивность потока *update*-запросов λ_U при этом равна 5 запросам в секунду).

Входная нагрузка на систему для *query*-запросов аналогично определяется в операторе 15 GENERATE

(EXPONENTIAL(1,0,125)), где задается экспоненциальное распределение интервалов между *query*-запросами. Среднее значение интервала задано третьим операндом функции EXPONENTIAL. В данном случае это значение равно $T_Q = 125$ мс (интенсивность потока *query*-запросов λ_Q при этом равна 8 запросам в секунду).

На рисунках 7 и 8 представлены некоторые результаты статистических экспериментов с GPSS-моделью системы типа DBaaSOD как для приоритетного обслуживания *update*-запросов по сравнению с обслуживанием *query*-запросов (рис. 7), так и для случая равных приоритетов, что эквивалентно в данном случае для бесприоритетного режима обслуживания запросов (рис. 8) при числе реплик баз данных $Res = 8$. Везде полагалось, что интенсивность потока *update*-запросов $\lambda_U = 5$ 1/с, то есть 5 запросам в секунду. В случае приоритетного обслуживания *update*-запросов время ожидания для них сокращается примерно в 2 – 4 раза по сравнению с *query*-запросами в зависимости от интенсивности поступления λ_Q *query*-запросов. При бесприоритетном режиме условия обслуживания *update*-запросов ухудшаются и время ожидания в очереди для них увеличивается примерно в 2 – 6 раз по сравнению с *query*-запросами в зависимости от интенсивности поступления λ_Q *query*-запросов.

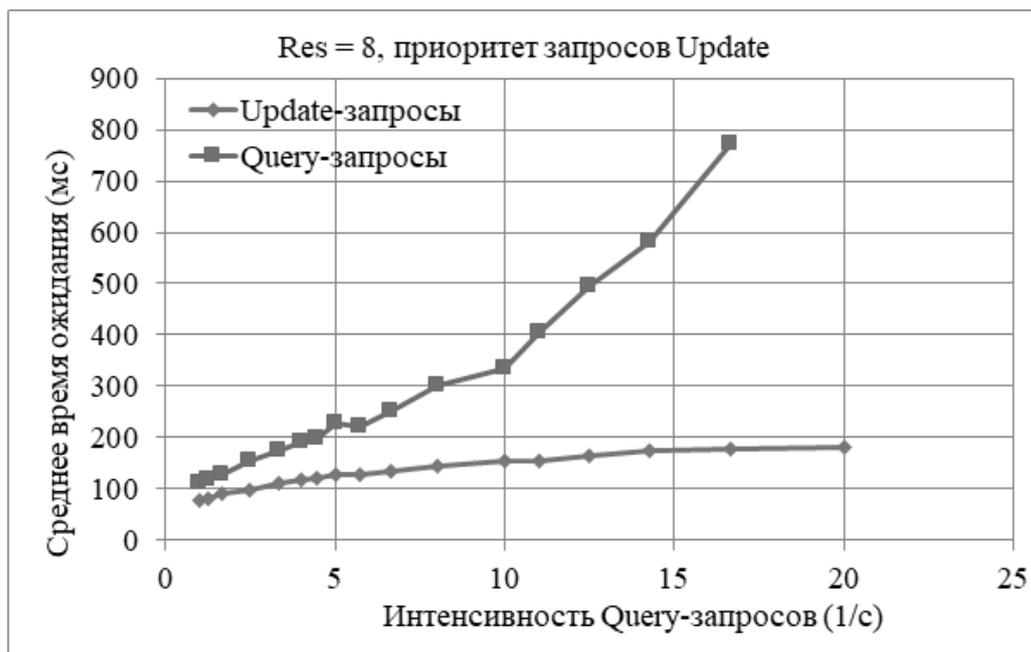


Рис. 7. Зависимости среднего времени ожидания запросов при приоритетном режиме; Res = 8

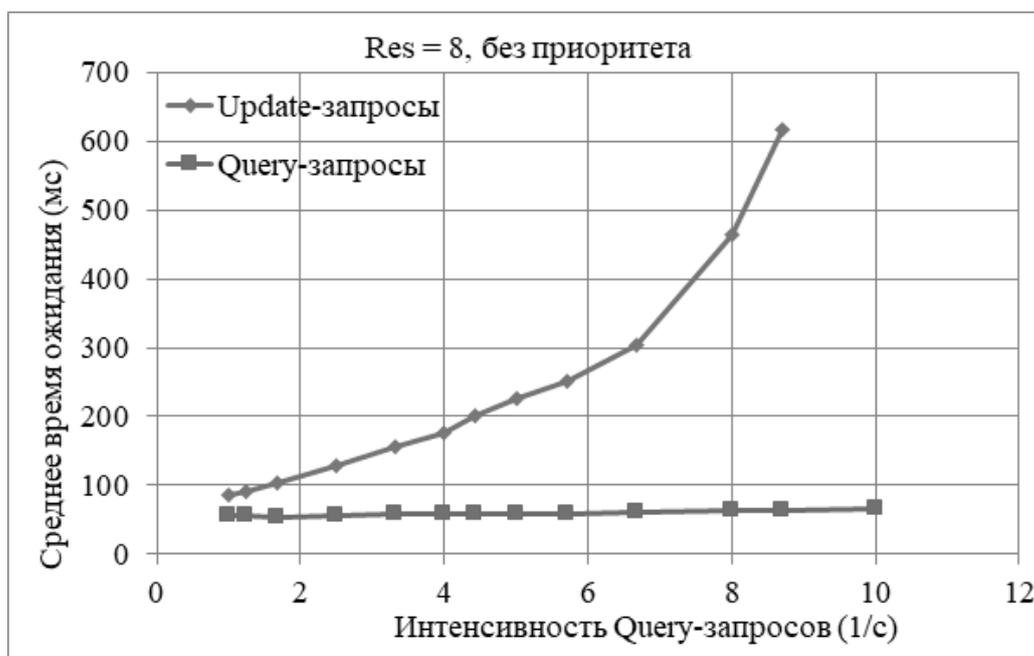


Рис. 8. Зависимости среднего времени ожидания запросов при бесприоритетном режиме; Res = 8

Моделирование системы при другом числе реплик *Res* дало похожие результаты. Таким образом, в результате проведенных статистических экспериментов установлено, что высший приоритет рекомендуется устанавливать для *update*-запросов, так как от них зависит

актуальность хранимой в базах данных информации и поэтому сокращение времени обслуживания для них целесообразно.

Построенная выше GPSS-модель позволяет получить оценку времени ожидания в очередях и дает лишь при-

близительное представление об организации функционирования системы. Реализацию распределенных приложений в системе типа DBaaSOD возможно осуществить путем непосредственного вложения сети Петри в архитектуру компьютерной TCP/IP сети. Метод отображения сети Петри в архитектуру беспроводной мобильной вычислительной сети MANET, управляющей роботизированным производством, предложен в работах [41, 42]. Метод базируется на использовании логико-алгебраических спецификаций. Однако возможно и непосредственное отображение сети Петри в архитектуру компьютерной сети. Пример вложения фрагмента сети Петри в распределенную вычислительную систему на базе TCP/IP сети представлен на рисунке 9. Здесь представлены виртуальные узлы-позиции (Nodes-positions), узлы-переходы (Nodes-transitions) и метки, или маркеры (Markers). Описана ситуация, когда входные позиции p_1, p_2, p_3 , переход t_1 и выходная позиция p_4 и размещены на соответствующих логических узлах $n_{p1}, n_{p2}, n_{p3}, n_{t1}$ и n_{p4} компьютерной TCP/IP сети. Определены следующие унарные функции f_{Node} и $f_{Address}$:

$$f_{Node}: P \cup T \rightarrow Node;$$

$$f_{Address}: Node \rightarrow IPaddress,$$

где $Node$ – множество логических узлов компьютерной сети; $IPaddress$ – множество физических узлов в TCP/IP сети с уникальными IP-адресами; P и T – множества позиций и переходов сети LogNet, реализуемых в TCP/IP сети.

Например, выражение $f_{Address}(f_{Node}(p_1))$ позволяет определить IP-адрес 129.44.192.1 узла $n_{p1} = f_{Node}(p_1)$, на котором реализована позиция p_1 . Дуги, изображенные на рисунке 9 – виртуальные, реально же при реализации сетей Петри в TCP/IP сети используется ее физическая инфраструктура – маршрутизаторы, каналы связи и другие объекты. Метки по физической сети передаются в виде сообщений.

При программной реализации распределенного приложения предлагаются следующие варианты активации перехода в TCP/IP сети.

Программа перехода t_1 , размещенная на узле n_{t1} , посылает запросы к узлам n_{p1}, n_{p2} и n_{p3} , на которых размещены программы, реализующие позиции p_1, p_2 и p_3 соответственно.

В ответ удаленные программы посылают сообщения-маркеры (если они имеются в наличии; в противном случае происходит ожидание их появления) на узел n_{t1} , и после приема этих сообщений программа перехода t_1 отправляет сообщение-маркер со своего узла n_{t1} на узел n_{p4} , где оно принимается программой позиции p_4 .

Второй вариант имеет следующие отличия в реализации: после приема опрашиваемых сообщений, поступающих от перехода к его входным позициям, найденные маркеры блокируются. Узел-переход n_{t1} сработает только в том случае, когда поступит подтверждение о том, что во всех его входных позициях имеются маркеры.

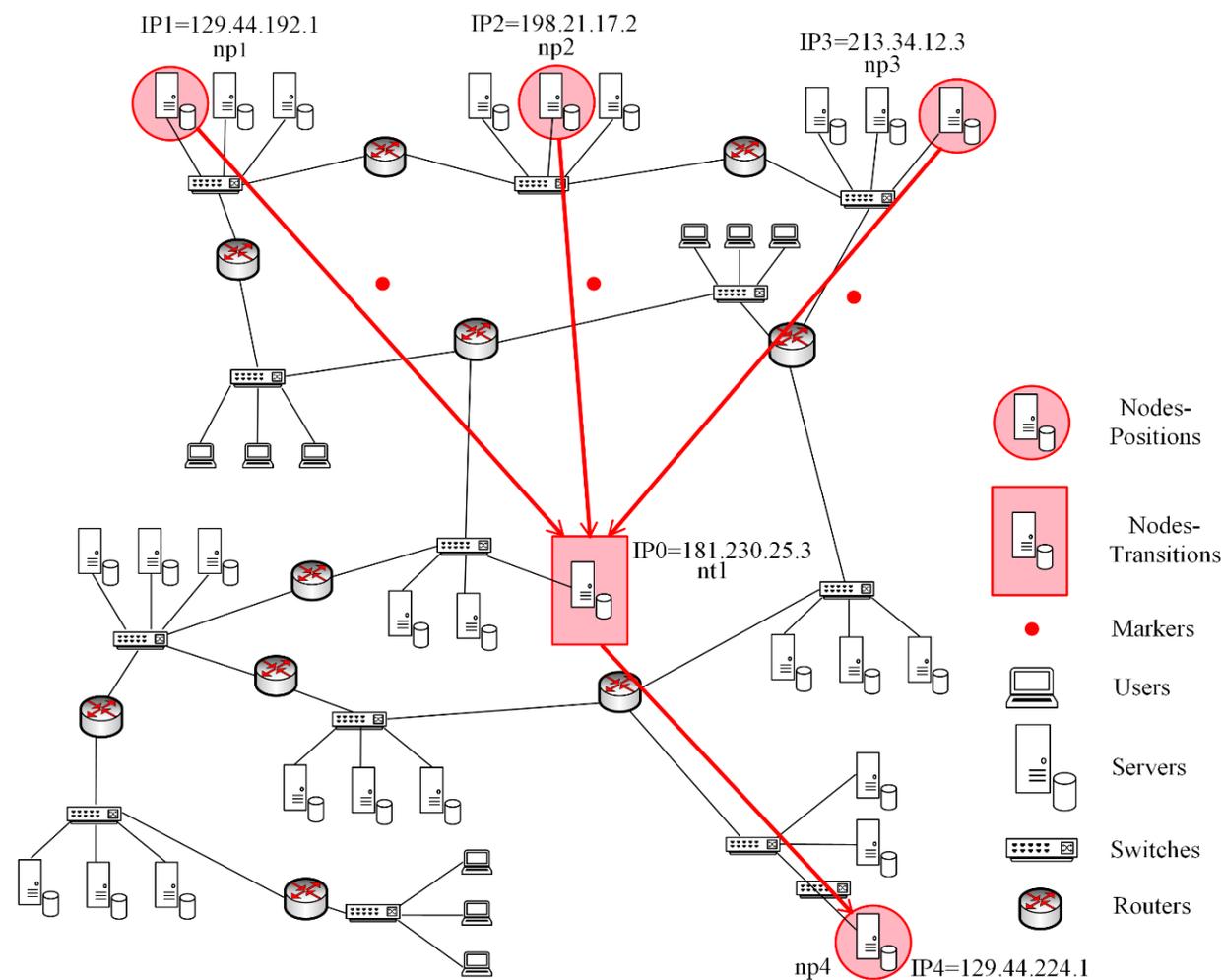


Рис. 9. Пример вложения фрагмента сети Петри в распределенную вычислительную систему на базе TCP/IP сети

В третьем случае все сообщения-маркеры передаются немедленно на узел-переход после их появления на входных узлах-позициях. Получив все необходимые сообщения, узел-переход n_{i1} срабатывает и отправляет выходное сообщение на узел-позицию n_{p4} .

В случае использования в сети Петри ингибиторных и информационных дуг на узел-переход поступают не сообщения-маркеры, а маркеры-уведомления о текущем состоянии входных узлов-позиций, поступающие после предварительного опроса со стороны

узла-перехода. Например, после опроса узла-позиции, связанного с узлом-переходом ингибиторной дугой, поступает маркер-уведомление о том, что в узле-позиции нет ни одного маркера, а после опроса узла-позиции, связанного с узлом-переходом информационной дугой, поступает маркер-уведомление о том, что в узле-позиции есть по крайней мере один маркер.

Возможны и другие варианты реализации процедур переходов в компьютерной сети. В общем случае правило срабатывания перехода имеет следую-

щий вид (в форме модуля-процедуры mt_i некоторого перехода t_i):

$$mt_i: A;[\alpha](B \vee Ret_{\tau,n});C,$$

где операторам A и C соответствуют действия, выполняемые до и после срабатывания перехода t_i в сети; α – условие активизации перехода, определяемое состоянием его входных позиций; B – процедура перехода t_i , выполняемая при истинном условии α и определяющая основные действия (например, поиск и чтение данных, внесение изменений в базу данных), а также операции с содержимым входных и выходных позиций; $Ret_{\tau,n}$ – оператор возврата к проверке условия α , которая может повторяться неоднократно (но не более n раз), символ “;” означает последовательное выполнение, или композицию, операторов.

Алгоритм реализации модуля-процедуры перехода представлен на рисунке 10. Особенностью алгоритма является то, что опрос состояния входных позиций в сети повторяется не более n раз, между последовательными опросами отрабатывается тайм-аут $Delay(\tau)$ на τ единиц реального времени в сети. В случае, когда условие α оказалось ложным после n опросов, выполняется невозможный оператор N , и работа перехода t_i блокируется.

Выводы

На основе двух методов имитационного моделирования проведен анализ функционирования облачной системы с репликами баз данных, в которой взаимо-

действуют две распределенные облачные вычислительные системы: MANET Cloud на основе беспроводной сети и Internet Cloud на основе сети Интернет.

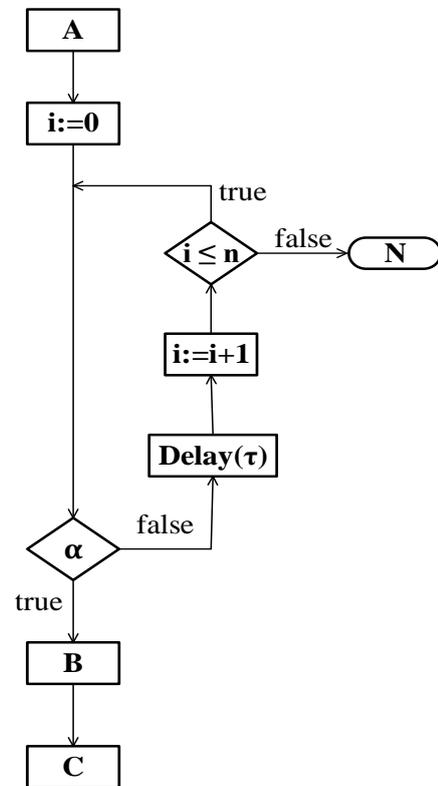


Рис. 10. Алгоритм сетевой реализации модуля-процедуры перехода

В совокупности базы данных являются основой облачного сервиса DBaaS – Data Bases as a Service on Demand (базы данных как сервис, организуемый по требованию пользователя). Для исследования данной системы построены модели двух классов. Модель на основе сети Петри предназначена для проверки моделируемого распределенного приложения на правильность функционирования. Обсуждены решения по отображению сетей Петри на архитектуру компьютерных сетей. Имитационная статистическая модель

использовалась для сравнения приоритетного и бесприоритетного режимов обслуживания query- и update-запросов по критерию среднего времени пребывания запросов в очередях.

Модели системы на основе сетей Петри прошли проверку, которая показала их живость и безопасность, что позволяет от моделей переходить к построению формализованных спецификаций для сетевых приложений для сетевых облачных сервисов в распределенных вычислительных системах с реплицированными базами данных. Ис-

следование GPSS-модели показало, что в случае приоритетного обслуживания update-запросов время ожидания для них сокращается примерно в 2 – 4 раза по сравнению с query-запросами в зависимости от интенсивности поступления query-запросов. При бесприоритетном режиме условия обслуживания update-запросов ухудшаются и время ожидания в очереди для них увеличивается примерно в 2 – 6 раз по сравнению с query-запросами в зависимости от интенсивности поступления query-запросов.

Список литературы

1. Antonopoulos N., Gillam L. (Eds). *Cloud Computing. Principles, Systems and Applications*. Springer. London, Dordrecht, New York, Heidelberg, 2010. 379 p.
2. *Handbook of Cloud Computing* / Furht B., Escalante A. (Eds). Springer. New York, Dordrecht, Heidelberg, London, 2010. 634 p.
3. Foster I., Zhao Y., Raicu I, Lu S. Cloud computing and grid computing 360-degree compared // *Grid Computing Environments Workshop (GCE'08)*, 2008, pp. 1-10.
4. *Apache CloudStack: Open Source Infrastructure as a Service Cloud Computing Platform* / Kumar R., Jain K., Maharwal H., Jain N., Dadhich A. // *International Journal of advancement in Engineering technology, Management and Applied Science (IJAETMAS)*. 2014. Vol. 1, is. 2. P. 111-116.
5. Foster I., Kesselman C., Nick J., Tuecke S. *The Physiology of the Grid: An Open Grid Service Architecture for Distributed Systems Integration*. 30 p. URL: <https://www.dcc.fc.up.pt/~ines/aulas/1314/CG/papers/physiology.pdf>. Дата доступа: 10.10.2018.
6. *Advances in Grid Computing* / Edited by Zoran Constantinescu. Published by InTech, Janeza Trdine 9, 51000 Rijeka, Croatia, 2011. 272 p.
7. Демичев А. П., Ильин В. А., Крюков А. П. Введение в грид-технологии: препринт НИИЯФ МГУ 11/832. М., 2007. 87 с.
8. *The Grid. Blueprint for a new computing infrastructure* / Foster I., Kesselman C. (Eds). San Francisco: Morgan Kaufman, 1999. 677 p.
9. *Cluster computing and applications* / M. Baker, A. Apon, R. Buyya, H. Jin. In A. Kent & J. Williams (Eds). *Encyclopedia of Computer Science and Technology*, 2002, pp. 87-125.

10. Sadashiv N., Kumar S. M. D. Cluster, Grid and Cloud Computing: A Detailed Comparison // The 6th International Conference on Computer Science & Education (ICCSE 2011), August 3-5, 2011. SuperStar Virgo, Singapore, 2011, pp. 477-482.
11. Jaiswal U. C. Study and Applications of Cluster Grid and Cloud Computing // International Journal of Engineering Research and Development. 2012. Vol. 3, No. 1. P. 45-50. e-ISSN: 2278-067X, p-ISSN: 2278-800X.
12. Samah Mawia Ibrahim Omer, Amin Babiker A. Mustafa, Fatema Abdallah Elmahdi Alghali. Comparative study between Cluster, Grid, Utility, Cloud and Autonomic computing // IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE). 2014. Vol. 9, is. 6, Ver. III. P. 61-e-ISSN: 2278-1676, p-ISSN: 2320-3331.
13. Kahanwal B., Singh T. P. The Distributed Computing Paradigms: P2P, Grid, Cluster, Cloud, and Jungle // International Journal of Latest Research in Science and Technology. Vol. 1, is. 2, 2012, pp. 183-187. ISSN (Online): 2278-5299.
14. Jungle Computing: Distributed Supercomputing beyond Clusters, Grids, and Clouds / Frank J. Seinstra, Jason Maassen, Rob V. van Nieuwpoort, Niels Drost, Timo van Kessel, Ben van Werkhoven, Jacopo Urbani, Cerial Jacobs, Thilo Kielmann, Henri E. Bal. Department of Computer Science, Vrije Universiteit, De Boelelaan 1081A, 1081 HV Amsterdam, The Netherlands, 2010, pp. 1-31.
15. Kumar R. Comparison between Cloud Computing, Grid Computing, Cluster Computing and Virtualization // International Journal of Modern Computer Science and Applications (IJMCSA). January, 2015. Vol. 3, is. 1. P. 42-47.
16. Chen Rui, Meng Xiao-jing. Modeling of UDP Hole Punching in P2P Network Using Petri Net // International Proceedings of Economics Development and Research. 2012. Vol. 49. P. 150-154.
17. Park H., Ratzin R. I., Schaar M. Peer-to-Peer Networks: Protocols, Cooperation and Competition // Source Title: Streaming Media Architectures, Techniques, and Applications: Recent Advances, 2011. 33 p.
18. Mittal G., Kesswani N., Goswami K. A Survey of Current Trends in Distributed, Grid and Cloud Computing // International Journal of Advanced Studies in Computer Science and Engineering (IJASCSE). 2013. Vol. 2, no. 3. P. 1-6.
19. Tanenbaum A. S., Maarten Van Steen. Distributed Systems: principles and paradigms. 2nd Edition. Pearson Education, Inc., 2007. 669 p.
20. Distributed and Cloud Computing From Parallel Processing to the Internet of Things / Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra. Elsevier, 2012. 648 p.
21. Nicola R., Gorla D., Pugliese R. Global computing in a dynamic network of tuple spaces // Science of Computer Programming. 2007. № 6. P. 187–204.
22. Зинкин С. А., Мустафа Садек Джафар. Развитие информационно-коммуникационных инфраструктур распределенных вычислительных систем на основе кон-

цепции «Сеть – это компьютер» // Известия Юго-Западного государственного университета. 2018. Т. 22. №4 (79). С. 75-93. DOI: 10.21869/2223-1560-2018-22-4-75-93.

23. Asynchronous Replication Engines // Sombers Associates, Inc., and W. H. Highleyman. November 2006, pp. 1-6.

24. Alireza Souri, Ahmad Habibizad Navin. Consistency of Data Replication Protocols in Database Systems: A Review // International Journal on Information Theory (IJIT). 2014. Vol. 3, no.4. P. 19 – 32.

25. Saadat N., Rahmani A. M. PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids // Future Generation Computer Systems, 2012. Vol. 28. P. 666-681.

26. Wu A.-H., Tan Z.-J., Wang W. // Annotation Based Query Answer over Inconsistent Database // Journal of Computer Science and Technology. 2010. Vol. 25. P. 469-481.

27. Mansouri N., Dastghaibyfar G. H., Mansouri E. // Combination of data replication and scheduling algorithm for improving data availability in Data Grids // Journal of Network and Computer Applications. 2013. Vol. 36. P. 711-722.

28. Mansouri N., Dastghaibyfar G. H. A dynamic replica management strategy in data grid // Journal of Network and Computer Applications. 2012. Vol. 35. P. 1297-1303.

29. Liao X., Jin H., Yu L. A novel data replication mechanism in P2P VoD system // Future Generation Computer Systems. 2012. Vol. 28. P. 930-939.

30. Chiu-Ching Tuan, Yi-Chao Wu. Grid Header Election by Predetermining in Mobile Ad-Hoc Networks // Journal of Applied Science and Engineering. 2012. Vol. 15, no. 1. P. 69-78.

31. Imran Ihsan, Muhammad Abdul Qadir, Nadeem Iftikhar. Mobile Ad-Hoc Service Grid – MASGRID // International Journal of Electrical, Robotics, Electronics and Communications Engineering. 2007. Vol. 1, no. 5. P. 778-781.

32. Bhaskaran R., Madheswaran M. Performance Analysis of Congestion Control in Mobile Ad-hoc Grid Layer // International Journal of Computer Applications. February, 2010. Vol. 1, no. 20. P. 102-110.

33. Li J., Khan S., Li Q. An efficient event delivery scheme in mobile ad hoc communities // Int. J. Communication Networks and Distributed Systems. 2013. Vol. 10, no. 1. P.25–39.

34. An Approach to Ad hoc Cloud Computing / Kirby G., Dearle A., Macdonald A., Fernandes A. // Distributed, Parallel, and Cluster Computing. 2010. P. 1-6.

35. Forouzan B. A. TCP/IP Protocol Suite. McGraw-Hill, 2009. 1024 p.

36. OddCI: On-Demand Distributed Computing Infrastructure / Costa R., Brasileiro F., Filho G. L., Sousa D. M. // Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS'09), November 16th, 2009, Portland, Oregon, USA, Association for Computing Machinery, 2009, pp. 1-10.

37. Mustafa Sadeq Jaafar, Zinkin S. A., Pashchenko D. V. Reconfigurable network models for distributed computing systems // Proceedings of the Fourteenth International

Conference of Science and Technology “New Information Technologies and Systems” (NITIS-2017). Penza, Russia, November 22-24, 2017. P. 92-104.

38. Mustafa Sadeq Jaafar, Zinkin S. A. The Implementation of Global Computing Through the Mapping of Object-Oriented Petri Nets into the Architecture of Distributed Computing Systems // Proceedings of the Fourteenth International Conference of Science and Technology “New Information Technologies and Systems” (NITIS-2017). Penza, Russia, November 22-24, 2017. P. 105-116.

39. GPSS World Reference Manual. URL: http://www.minutemansoftware.com/reference/reference_manual.htm. Дата доступа 10.10.2018.

40. PIPE2: Platform Independent Petri net Editor 2. URL: <http://pipe2.sourceforge.net>. Дата доступа 10.10.2018.

41. Интеграция методов концептуального и поведенческого моделирования дискретно-событийных систем: I. Синтез и анализ концептуальной модели / Мустафа Садек Джафар, С. А. Зинкин, Д. В. Пашенко, У. Н. Пучкова // Кибернетика и программирование. 2016. № 6. С. 83-95.

42. Интеграция методов концептуального и поведенческого моделирования дискретно-событийных систем: II. Логико-алгебраические операционные модели и инфокоммуникационные технологии / Мустафа Садек Джафар, С. А. Зинкин, Д. В. Пашенко, У. Н. Пучкова // Кибернетика и программирование. 2017. № 1. С. 75-93.

Поступила в редакцию 01.03.2019

Подписана в печать 28.03.2019

Reference

1. Antonopoulos N., Gillam L. (Eds). Cloud Computing. Principles, Systems and Applications. Springer. London, Dordrecht, New York, Heidelberg, 2010, 379 p.

2. Furht B., Escalante A. (Eds). Handbook of Cloud Computing. Springer. New York, Dordrecht, Heidelberg, London, 2010, 634 p.

3. Foster I., Zhao Y., Raicu I, Lu S. Cloud computing and grid computing 360-degree compared. Grid Computing Environments Workshop (GCE'08), 2008, pp. 1-10.

4. Kumar R., Jain K., Maharwal H., Jain N., Dadhich A. Apache CloudStack: Open Source Infrastructure as a Service Cloud Computing Platform. *International Journal of advancement in Engineering technology, Management and Applied Science (IJAETMAS)*, 2014, vol. 1, is. 2, pp. 111-116.

5. Foster I., Kesselman C., Nick J., Tuecke S. The Physiology of the Grid: An Open Grid Service Architecture for Distributed Systems Integration. 30 p. URL: <https://www.dcc.fc.up.pt/~ines/aulas/1314/CG/papers/physiology.pdf>. Дата доступа: 10.10.2018.

6. *Advances in Grid Computing*; ed. by Zoran Constantinescu. Published by InTech, Janeza Trdine 9, 51000 Rijeka, Croatia, 2011, 272 p.

7. Demichev A. P., Il'in V. A., Krjukov A. P. *Vvedenie v grid-tehnologii*. Preprint NII-JaF MGU 11/832. Moscow, 2007. 87 p. (In Russ.).

8. Foster I., Kesselman C. (Eds). *The Grid. Blueprint for a new computing infrastructure*. San Francisco, Morgan Kaufman Publ., 1999, 677 p.

9. Baker M., Apon A., Buyya R., Jin H.. In A. Kent & J. Williams (Eds) *Cluster computing and applications*. Encyclopedia of Computer Science and Technology, 2002, pp. 87-125.

10. Sadashiv N., Kumar S. M. D. *Cluster, Grid and Cloud Computing: A Detailed Comparison*. The 6th International Conference on Computer Science & Education (ICCSE 2011), August 3-5, 2011. SuperStar Virgo, Singapore, 2011, pp. 477-482.

11. Jaiswal U. C. *Study and Applications of Cluster Grid and Cloud Computing*. *International Journal of Engineering Research and Development*, 2012, vol. 3, no. 1, pp. 45-50. e-ISSN: 2278-067X, p-ISSN: 2278-800X.

12. Samah Mawia Ibrahim Omer, Amin Babiker A. Mustafa, Fatema Abdallah Elmahdi Alghali. *Comparative study between Cluster, Grid, Utility, Cloud and Autonomic computing*. *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, 2014, vol. 9, is. 6, Ver. III, pp. 61. e-ISSN: 2278-1676, p-ISSN: 2320-3331.

13. Kahanwal B., Singh T. P. *The Distributed Computing Paradigms: P2P, Grid, Cluster, Cloud, and Jungle*. *International Journal of Latest Research in Science and Technology*, 2012, vol. 1, is. 2, pp. 183-187. ISSN (Online): 2278-5299,

14. Frank J. Seinstra, Jason Maassen, Rob V. van Nieuwpoort, Niels Drost, Timo van Kessel, Ben van Werkhoven, Jacopo Urbani, Cerial Jacobs, Thilo Kielmann, Henri E. Bal. *Jungle Computing: Distributed Supercomputing beyond Clusters, Grids, and Clouds*. Department of Computer Science, Vrije Universiteit, De Boelelaan 1081A, 1081 HV Amsterdam, The Netherlands, 2010, pp. 1-31.

15. Kumar R. *Comparison between Cloud Computing, Grid Computing, Cluster Computing and Virtualization*. *International Journal of Modern Computer Science and Applications (IJMCSA)*, January 2015, vol. 3, no.1, pp. 42-47.

16. Chen Rui, Meng Xiao-jing. *Modeling of UDP Hole Punching in P2P Network Using Petri Net*. *International Proceedings of Economics Development and Research*, 2012, vol. 49, pp. 150-154.

17. Park H., Ratzin R. I., Schaar M. *Peer-to-Peer Networks: Protocols, Cooperation and Competition*. Source Title: *Streaming Media Architectures, Techniques, and Applications: Recent Advances*, 2011, 33 p.

18. Mittal G., Kesswani N., Goswami K. *A Survey of Current Trends in Distributed, Grid and Cloud Computing*. *International Journal of Advanced Studies in Computer Science and Engineering (IJASCSE)*, 2013, vol. 2, no. 3, pp. 1-6.

19. Tanenbaum A. S., Maarten Van Steen. Distributed Systems: principles and paradigms. 2nd Edition. Pearson Education, Inc., 2007, 669 p.
20. Distributed and Cloud Computing From Parallel Processing to the Internet of Things. Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra. Elsevier, 2012, 648 p.
21. Nicola R., Gorla D., Pugliese R. Global computing in a dynamic network of tuple spaces. *Science of Computer Programming*, 2007, no. 64, pp. 187–204.
22. Zinkin S. A., Mustafa Sadek Dzhafar. Razvitie informacionno-kommunikacionnyh infrastruktur raspredelennyh vychislitel'nyh sistem na osnove koncepcii "Set' – jeto komp'juter". *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2018, vol. 22, no.4 (79), pp. 75-93 (In Russ.). DOI: 10.21869/2223-1560-2018-22-4-75-93.
23. Asynchronous Replication Engines. Sombers Associates, Inc., and W. H. Highleyman. November 2006, pp. 1-6.
24. Alireza Sourì, Ahmad Habibizad Navin. Consistency of Data Replication Protocols in Database Systems: A Review. *International Journal on Information Theory (IJIT)*, 2014, vol. 3, no.4, pp. 19 – 32.
25. Saadat N., Rahmani A. M. PDDRA: A new prefetching based dynamic data replication algorithm in data grids. *Future Generation Computer Systems*, 2012, vol. 28, pp. 666-681.
26. Wu A.-H., Tan Z.-J., Wang W. Annotation Based Query Answer over In-consistent Database. *Journal of Computer Science and Technology*, 2010, vol. 25, pp. 469-481.
27. Mansouri N., Dastghaibyfar G. H., Mansouri E. Combination of data replication and scheduling algorithm for improving data availability in Data Grids. *Journal of Network and Computer Applications*, 2013, vol. 36, pp. 711-722.
28. Mansouri N., Dastghaibyfar G. H. A dynamic replica management strategy in data grid. *Journal of Network and Computer Applications*, 2012, vol. 35, pp. 1297-1303.
29. Liao X., Jin H., Yu L. A novel data replication mechanism in P2P VoD system. *Future Generation Computer Systems*, 2012, vol. 28, pp. 930-939.
30. Chiu-Ching Tuan, Yi-Chao Wu. Grid Header Election by Predetermining in Mobile Ad-Hoc Networks. *Journal of Applied Science and Engineering*, 2012, vol. 15, no. 1, pp. 69-78.
31. Imran Ihsan, Muhammad Abdul Qadir, Nadeem Iftikhar. Mobile Ad-Hoc Service Grid – MASGRID. *International Journal of Electrical, Robotics, Electronics and Communications Engineering*, 2007, vol. 1, no. 5, pp. 778-781.
32. Bhaskaran R., Madheswaran M. Performance Analysis of Congestion Control in Mobile Ad-hoc Grid Layer. *International Journal of Computer Applications*, February, 2010, vol. 1, no. 20, pp. 102-110.
33. Li J., Khan S., Li Q. An efficient event delivery scheme in mobile ad hoc communities. *Int. J. Communication Networks and Distributed Systems*, 2013, vol. 10, no. 1, pp.25–39.

34. Kirby G., Dearle A., Macdonald A., Fernandes A. An Approach to Ad hoc Cloud Computing. Distributed, Parallel, and Cluster Computing, 2010, pp. 1-6.
35. Forouzan B. A. TCP/IP Protocol Suite. McGraw-Hill, 2009, 1024 p.
36. Costa R., Brasileiro F., Filho G. L., Sousa D. M. OddCI: On-Demand Distributed Computing Infrastructure / Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS'09), November 16th, 2009, Portland, Oregon, USA, Association for Computing Machinery, 2009, pp. 1-10.
37. Mustafa Sadeq Jaafar, Zinkin S. A., Pashchenko D. V. Reconfigurable network models for distributed computing systems. Proceedings of the Fourteenth International Conference of Science and Technology "New Information Technologies and Systems" (NITIS-2017). Penza, Russia, November 22-24, 2017, pp. 92-104.
38. Mustafa Sadeq Jaafar, Zinkin S. A. The Implementation of Global Computing Through the Mapping of Object-Oriented Petri Nets into the Architecture of Distributed Computing Systems. Proceedings of the Fourteenth International Conference of Science and Technology "New Information Technologies and Systems" (NITIS-2017). Penza, Russia, November 22-24, 2017, pp. 105-116.
39. GPSS World Reference Manual. URL: http://www.minutemansoftware.com/reference/reference_manual.htm. Data dostupa 10.10.2018.
40. PIPE2: Platform Independent Petri net Editor 2. URL: <http://pipe2.sourceforge.net>. Data dostupa 10.10.2018.
41. Mustafa Sadek Dzhafar, Zinkin S. A., Pashhenko D. V., Puchkova U. N. Integracija metodov konceptual'nogo i povedencheskogo modelirovanija diskretno-sobytijnyh sistem: I. Sintez i analiz konceptual'noj modeli. *Kibernetika i programirovanie*, 2016, no. 6, pp. 83-95 (In Russ.).
42. Mustafa Sadek Dzhafar, Zinkin S. A., Pashhenko D. V., Puchkova U. N. Integracija metodov konceptual'nogo i povedencheskogo modelirovanija diskretno-sobytijnyh sistem: II. Logiko-algebraicheskie operacionnye modeli i infokommunikacionnye tehnologii. *Kibernetika i programirovanie*, 2017, no. 1, pp. 75-93 (In Russ.).

Received 01.03.2019

Accepted 28.03.2019

Информация об авторах / Information about the Authors

Мустафа Садек Джафар, аспирант,
кафедра вычислительной техники, Пензенский
государственный университет, г. Пенза,
Российская Федерация,
e-mail: mustafajaafar221087@gmail.com

Mustafa Sadeq Jaafar, Post-Graduate Student,
Department of Computer Engineering,
Penza State University, Penza, Russian Federation,
e-mail: mustafajaafar221087@gmail.com