

---

---

# СОВРЕМЕННЫЕ ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

---

---

УДК 004.75

## БАЛАНСИРОВКА НАГРУЗКИ В ОБЛАЧНЫХ ВЫЧИСЛЕНИЯХ

Е. Н. Десятирикова\*, Хадж Али Муса\*\*, Ходар Алмосана\*,  
Алькади Усама\*, Раджаб Хаян\*\*\*

\*Воронежский государственный университет

\*\*Университет Дамаска, Сирия

\*\*\*Университет Тишрин, Сирия

Поступила в редакцию 07.09.2017 г.

**Аннотация.** Облачные вычисления – это развивающаяся интернет-технология, в том числе, для коммерческих вычислений. Балансировка нагрузки помогает улучшить их производительность. В работе с помощью инструмента cloud analyst анализируются различные алгоритмы балансировки нагрузки, проводится их сравнительный анализ.

**Ключевые слова:** облачные вычисления, балансировка нагрузки, облачный аналитик, моделирование, алгоритм Round Robin, алгоритм AMLB, Throttled алгоритм балансировки нагрузки.

**Annotation.** Cloud computing is the emerging internet based technology which emphasizes commercial computing. Load balancing helps in improving the performance of the centralized server. In the present work, various algorithms are analyzed using an analysis tool, namely, cloud analyst. Comparison is also made for algorithms load balancing.

**Keywords:** Cloud Computing, Load balancing, Cloud Analyst, Simulation, Round Robin Algorithm, AMLB Algorithm, Throttled Load Balancing Algorithm.

*Cloud computing* – «облачные вычисления» – концепция «вычислительного облака», согласно которой программы запускаются и выдают результаты работы в окно стандартного веб-браузера на локальном ПК, при этом все приложения и их данные, необходимые для работы, находятся на удаленном сервере в Интернете.

Концепция «облачных вычислений» зародилась в 1960 г., когда Джон Маккарти высказал предположение, что когда-нибудь компьютерные вычисления будут производиться с помощью общенародных утилит.

Термин *cloud computing* стал широко употребляться в США с 2005 года после запуска компанией Amazon.com проекта Elastic Compute Cloud (Amazon EC2) и широко распространился в бизнесе, среди поставщиков

информационных технологий и в научно-исследовательской среде.

К преимуществам облачных вычислений можно отнести следующие: снижаются требования к вычислительной мощности ПК, возрастают отказоустойчивость и безопасность, многократно увеличивается скорость обработки данных, снижаются затраты на аппаратное и программное обеспечение, на обслуживание, электроэнергию, экономится дисковое пространство.

Облачные вычисления – это технологии которые помогают обмениваться данными и предоставлять много ресурсов пользователям. Пользователи платят только за те ресурсы, которые они использовали. Облачные вычисления хранят данные и распределенные ресурсы в открытой среде, и объем хранения данных очень быстро увеличивается. Таким образом, балансировка нагрузки является основной задачей в облачной среде. Баланси-

---

© Десятирикова Е. Н., Хадж Али Муса, Ходар Алмосана, Алькади Усама, Раджаб Хаян, 2017

ровка нагрузки помогает распределить динамическую рабочую нагрузку между несколькими узлами, чтобы гарантировать, что ни один узел не будет перегружен.

Данное исследование в основном сосредоточено на анализе производительности облачных вычислений и сравнении различных алгоритмов балансировки нагрузки с использованием сетевого симулятора Cloud Analyst.

## 1. БАЛАНСИРОВКА НАГРУЗКИ

Балансировка нагрузки используется для распределения большей нагрузки на более мелкие узлы обработки для повышения общей производительности системы [1]. В облачной вычислительной среде распределение нагрузки необходимо распределить динамическую локальную рабочую нагрузку равномерно между всеми узлами.

Балансировка нагрузки помогает в справедливом распределении вычислительных ресурсов для достижения высокого уровня удовлетворенности пользователей и надлежащего использования ресурсов. Высокое использование ресурсов и правильная балансировка нагрузки помогают минимизировать потребление ресурсов. Это помогает реализовать отказоустойчивость, масштабируемость и избегать узких мест.

Балансировка нагрузки – это метод, который помог сетям и ресурсам, обеспечивать максимальную пропускную способность с минимальным временем отклика.

Балансировка нагрузки выполняется на двух уровнях в облачных вычислениях [2]:

- Уровень виртуальной машины, сопоставление сделано между приложениями, которые загружаются в облаке на виртуальную машину. Балансировщик нагрузки назначает запрошенную виртуальную машину физическим компьютерам, который уравнивает нагрузку множества приложений среди ПК.

- Уровень хоста, сопоставление между виртуальной машиной и хост-ресурсами, которые позволяют обрабатывать несколько поступающих запросов приложения.

## 2. СУЩЕСТВУЮЩИЕ ПОЛИТИКИ БАЛАНСИРОВКИ НАГРУЗКИ

Существуют различные алгоритмы балансировки нагрузки, используемые в облачных вычислениях. В данном исследовании изучены следующие три алгоритма, которые могут быть реализованы в симуляторе Cloud Analyst [3].

### 2.1. Round Robin алгоритм (RR)

Это самый простой алгоритм, который использует понятие кванта времени или интервала. Здесь время делится на несколько секторов, и каждому узлу задается определенный квант времени или временной интервал, и в этом кванте узел будет выполнять свои операции (рис. 1). В Round Robin планирование кванта времени играет очень важную роль, потому что если квант времени очень большой, то алгоритм планирования Round Robin такой же, как и FCFS планирование [3].

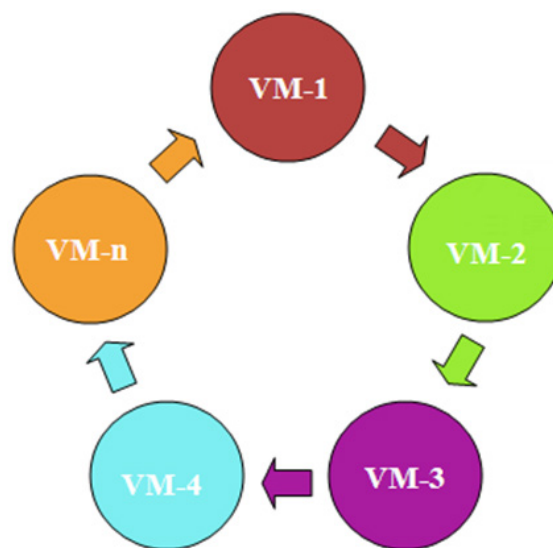


Рис. 1. Round Robin алгоритм (RR)

Недостатком метода является то, что, хотя алгоритм очень простой, но для определения размера кванта он генерирует дополнительную нагрузку на планировщик. Кроме того, он имеет более высокие переключатели контекста, которые увеличивают время оборота, и низкую пропускную способность [3].

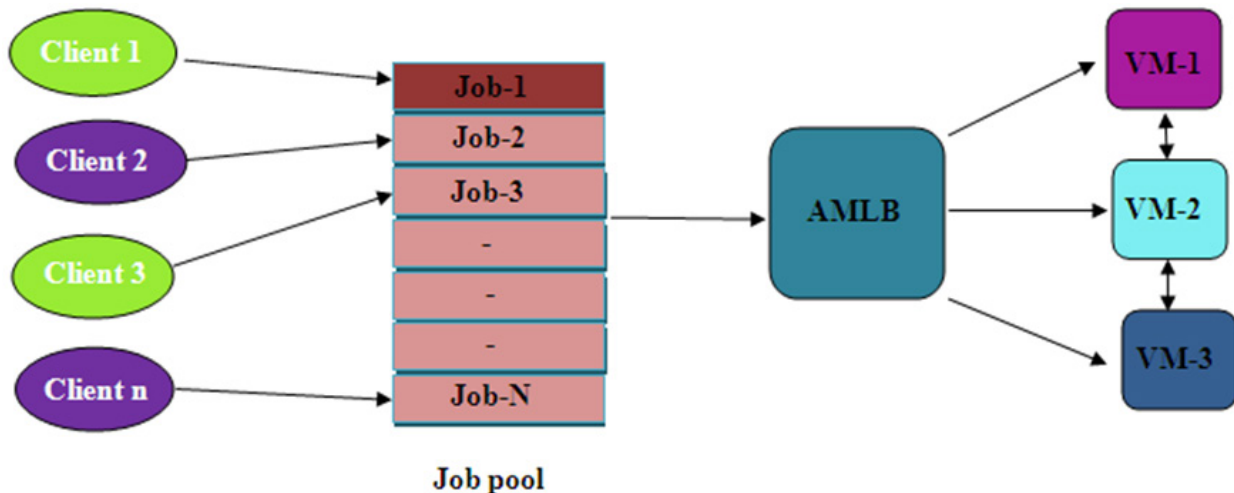


Рис. 2. AMLB алгоритм

## 2.2. Активный Мониторинг Балансировки Нагрузки (AMLB)

Этот алгоритм имеет динамический характер. Он хранит информацию о каждой виртуальной машине VM и количестве запросов, которые в настоящее время назначены для каждой VM. Когда запрос распределяется новой VM и если имеется несколько VM, то выбирается первая распознанная, и AMLB возвращает идентификатор VM в контроллер центра данных. Контроллер центра данных предупреждает AMLB о новом распределении и отправляет запрос в виртуальную машину, известную под этим идентификатором VM (рис. 2).

Недостаток алгоритма в том, что AMLB всегда находит наименее загруженную VM для назначения нового входящего запроса, но не проверяет, использовалась она ранее или нет (поэтому некоторая VM используется интенсивно, а некоторые по-прежнему не задействованы).

## 2.3. Throttled Load Balancing Algorithm (TLB)

В этом алгоритме балансировщик нагрузки поддерживает таблицу индексов виртуальных машин, а также их состояния (Доступно или Занято). Клиент/сервер сначала делает запрос в центр обработки данных, чтобы найти подходящую виртуальную машину (VM), для выполнения рекомендуемо-

го задания (рис. 3). Центр обработки данных запрашивает балансировщик нагрузки для распределения виртуальной машины. Балансировщик нагрузки сканирует индексную таблицу сверху до тех пор, пока не будет найдена первая доступная виртуальная машина или индексная таблица не будет полностью отсканирована.

Если виртуальная машина найдена, центр обработки данных передает запрос в виртуальную машину, идентифицированную идентификатором. Кроме того, центр данных подтверждает балансировку нагрузки нового распределения, и центр обработки данных соответствующим образом пересматривает индексную таблицу.

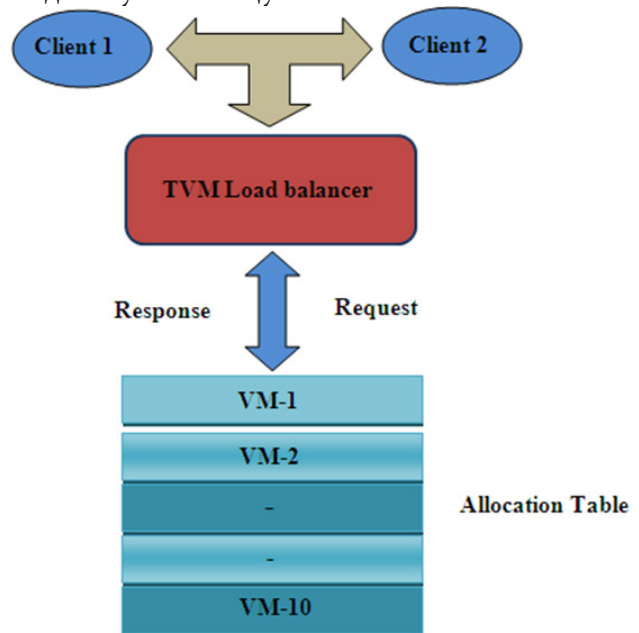


Рис. 3. Throttled алгоритм

При обработке запроса клиента, если соответствующая VM не найдена, балансировщик нагрузки возвращает «-1» в центр обработки данных. Запрос центра обрабатывается центром обработки данных.

Недостаток метода в том, что нагрузка на первые машины больше, чем на остальные в том случае, когда количество запросов невелико [3].

### 3. CLOUD ANALYST СИМУЛЯТОР

Моделирование и анализ производительности трех указанных алгоритмов балансировки нагрузки проведены в работе с помощью инструмента «Cloud Analyst» [4]. Он позволяет пользователю выполнять многократные симуляции с небольшими изменениями параметров, а также позволяет настраивать расположение пользователей, которые создают приложение, и расположение центров обработки данных [5]. Укажем терминологию эмулятора (рис.4):

– *регион*: в Cloud Analyst мир поделен на 6 регионов, которые совпадают с 6 основными континентами в мире;

– *база пользователей*: User Base рассматривается как единое целое, и используется для генерации трафика;

– *центр обработки данных*: брокерское обслуживание определяет, какой центр должен принять и обработать запрос, который приходит от каждой пользовательской базы;

– *VmLoadBalancer*: он отвечает за распределение нагрузки на доступный центр данных. VmLoadBalancer распределяет нагрузку в центре обработки данных на основе политики балансировки нагрузки.

В процессе моделирования использовано программное обеспечение CloudSim 4.0 [6].

### 4. МОДЕЛИРОВАНИЕ И ЭКСПЕРИМЕНТ

Моделирование и виртуальный эксперимент являются лучшим способом проверки алгоритма в облачных вычислениях. Рассмотрим работу каждого из трех алгоритмов балансировки нагрузки на примере социальной сети Facebook, имеющей более 200 миллионов зарегистрированных пользователей по всему миру (табл. 1).

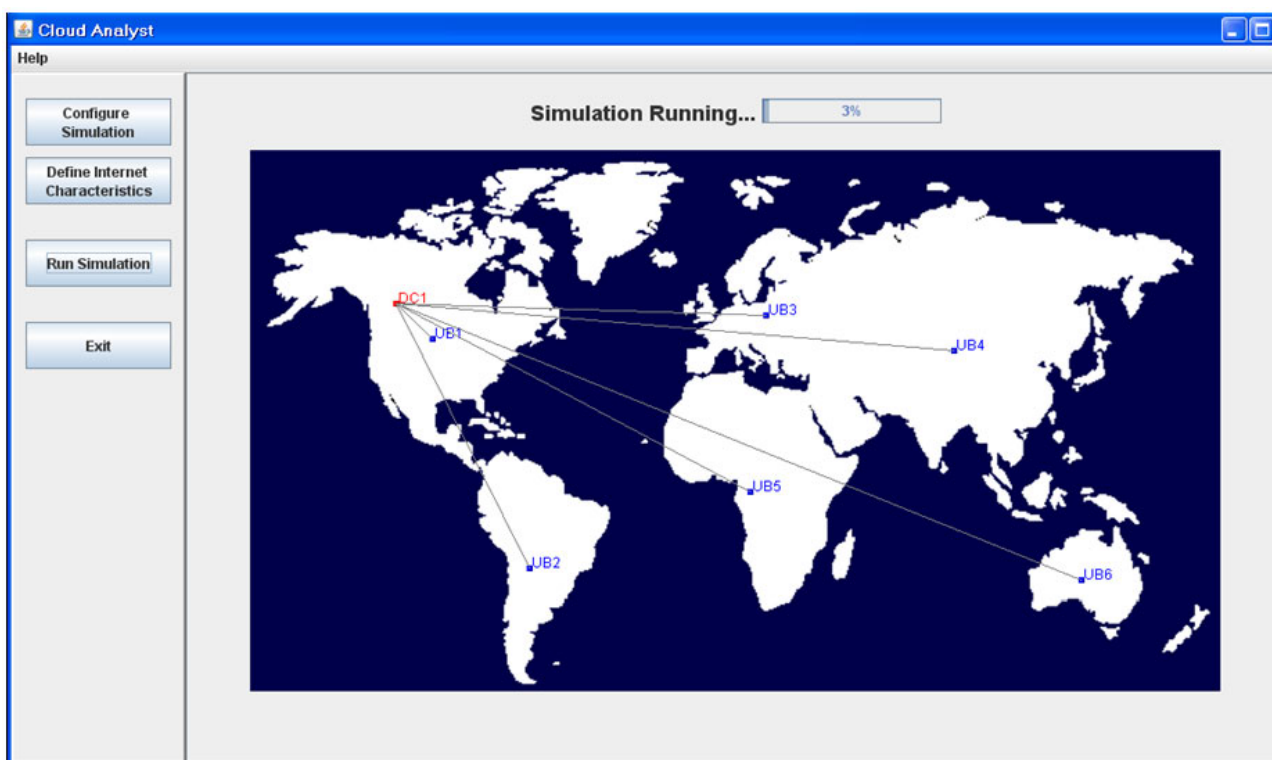


Рис 4. Cloud Analyst симулятор

Таблица 1

Зарегистрированные пользователи  
Facebook по всему миру

Регион	ID регион	Пользователи
Северная Америка	0	80 миллионов
Южная Америка	1	20 миллионов
Европа	2	60 миллионов
Азия	3	27 миллионов
Африка	4	5 миллионов
Океания	5	8 миллионов

Таблица 2

Параметры пользовательских баз

База	Регион	Часовой пояс	Час пик (Местное время)	Час пик (GMT)	пользователи онлайн в часы пик	пользователи онлайн в не пиковых часах
UB1	0	GMT - 6.00	7.00-9.00 pm	13:00-15:00	400,000	40,000
UB2	1	GMT - 4.00	7.00-9.00 pm	15:00-17:00	100,000	10,000
UB3	2	GMT + 1.00	7.00-9.00 pm	20:00-22:00	300,000	30,000
UB4	3	GMT + 6.00	7.00-9.00 pm	01:00-03:00	150,000	15,000
UB5	4	GMT + 2.00	7.00-9.00 pm	21:00-23:00	50,000	5,000
UB6	5	GMT+10.00	7.00-9.00 pm	09:00-11:00	80,000	8,000

Таблица 3

Параметры центра обработки данных

Name	Region	Arch	OS	VMM	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/GB	Physical Hw Units
Dc1	0	X86	Linux	Xen	0.1	0.05	0.1	0.1	20

**Configure Simulation**

Main Configuration | **Data Center Configuration** | Advanced

Data Centers:

Name	Region	Arch	OS	VMM	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/Gb	Physical HW Units
DC1	0	X86	Linux	Xen	0.1	0.05	0.1	0.1	2

Add New  
Remove

Рис. 5. Конфигурация центра обработки данных



Для нашего моделирования предположим, что мы имеем аналогичную систему, но в масштабе 1/10. Определим 6 пользовательских баз, представляющих указанные выше 6 регионов, со следующими параметрами (табл. 2).

Также определим центр обработки данных (табл. 3), который должен обработать запрос, приходящий от каждой пользовательской базы пользователей со следующими параметрами (рис. 5).

Ограничим модель тем, что каждая пользовательская база содержится в одном часовом поясе, и предположим, что большинство пользователей используют приложение по вечерам после работы около 2 часов.

Положим также, что 5 % из зарегистрированных пользователей будут on-line в пиковое время одновременно и только одна десятая из этого числа будут в сети в течение непиковых часов. Положим, что каждый пользователь делает новый запрос каждые 5 минут, когда находится on-line.

## 5. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Нами три раза проведено моделирование в соответствии с предыдущими параметрами. Каждый раз, мы меняли алгоритм ба-

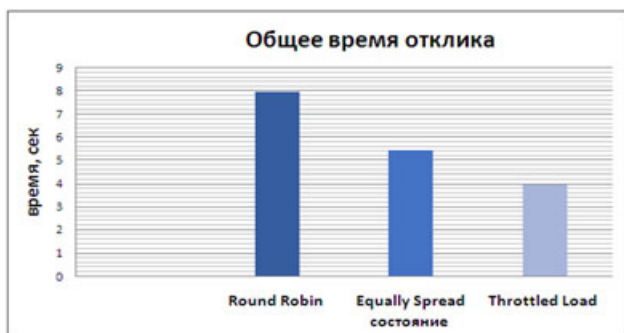


Рис. 6. Общее время отклика

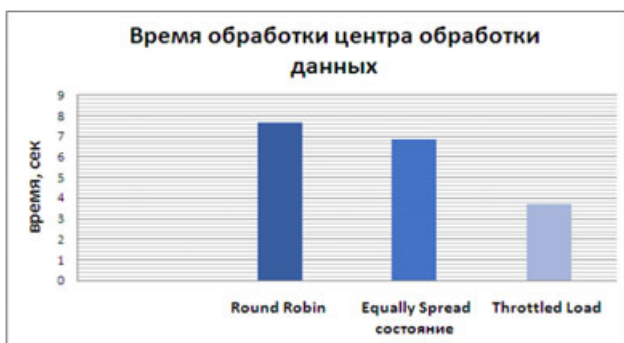


Рис. 7. Время обработки центра данных

лансировки нагрузки, который и анализировался. Сравнение результатов проводилось по критериям: общее время отклика (рис. 6), время центра обработки данных (рис. 7), почасовая нагрузка центра обработки данных (рис. 8–10) и стоимость обработки (табл. 4).

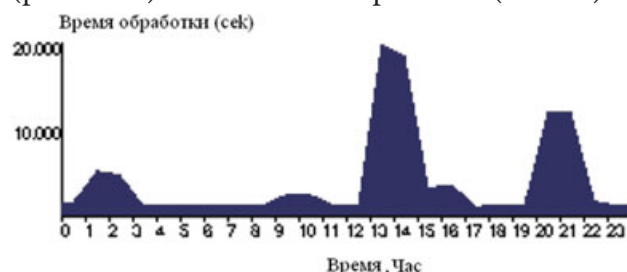


Рис. 8. Почасовая нагрузка центра обработки данных в (RR)

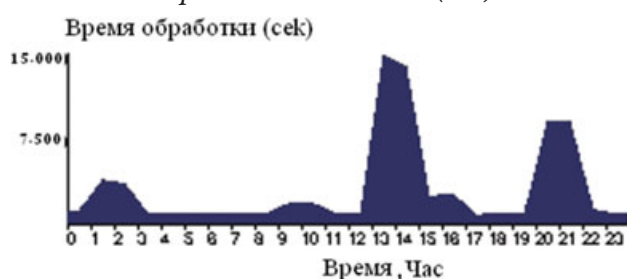


Рис. 9. Почасовая нагрузка центра обработки данных в (AMLB)

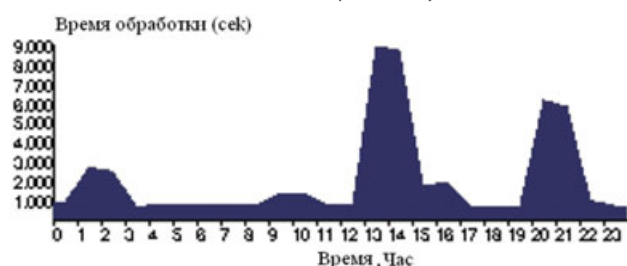


Рис. 10. Почасовая нагрузка центра обработки данных в (TLB)

Таблица 4

Стоимость обработки

Алгоритм	VM стоимость, \$	Полная стоимость, \$
RR	120.5	632.34
AMLB	112.8	580.67
TLB	99.1	501.92

## ЗАКЛЮЧЕНИЕ

Из вышеприведенного обзора балансировки нагрузки и трех существующих политик симулятора Cloud Analyst можно сделать вывод, что балансировка нагрузки является сложной задачей в облачных вычислениях.

Сравнивая результаты, полученные с использованием различных алгоритмов балансировки нагрузки, можно сделать вывод, что общее время отклика в Throttled алгоритме лучше, чем в других алгоритмах, также лучше и время центра обработки данных.

### Перспективы исследования

Эта работа может быть продолжена с использованием различных алгоритмов оптимизации «роя частиц». Кроме того, мы планируем расширить исследование методов балансировки нагрузки, используя различные политики брокерских услуг.

### СПИСОК ЛИТЕРАТУРЫ

1. *Бууа R. A.* Survey of Load Balancing Policies / R. Buuya, R. Ranjan, R. N. Calheiros // Int. Conf. High Perform. Comput. Simul. – USA: IEEE, 2009. – p. 1–11.

**Десятирикова Е. Н.** – д-р экон. наук, профессор, кафедра информационных систем, факультет компьютерных наук, Воронежский государственный университет.  
E-mail: science2000@ya.ru

**Хадж Али Муса** – д-р техн. наук, профессор, доктор компьютерных наук, кафедра информационных систем и сетей, факультет информатики, Университет Дамаска, Сирия.  
E-mail: hajalim@hotmail.com

**Ходар Алмосана** – магистр технических наук, аспирант кафедры информационных систем, факультет компьютерных наук, Воронежский государственный университет.  
E-mail: Mothana-sy@hotmail.com

**Алькади Усама** – магистр технических наук, аспирант кафедры информационных систем, факультет компьютерных наук, Воронежский государственный университет.  
E-mail: oalkadee@gmail.com

**Раджаб Хаян** – магистр технических наук, аспирант кафедры информационных систем и сетей, факультет информатики, Университет Тишрин, Сирия.  
E-mail: hayyan-r@hotmail.com

2. *Calheiros R. N.* CloudSim : A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services / R. N. Calheiros. Eprint: Australia, 2009. – p. 9–17.

3. *Simar P. S.* Analysis of Load Balancing Algorithms using Cloud Analyst / P.S Simar // International Journal of Grid and Distributed Computing. – 2016. – p. 11–24.

4. *Maguluri S. T.* Stochastic models of load balancing and scheduling in cloud computing clusters / S.T. Maguluri, R. Srikant, L. Ying // INFOCOM, 2012 Proc. IEEE. – USA: IEEE, 2012. – p. 702–710.

5. *Bhathiya W. A.* CloudAnalyst: A CloudSim based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments / W.A. Bhathiya. – UNIVERSITY OF MELBOURNE : Australia, 2009. – p. 21–35.

6. CloudSim web page at GitHub. – режим доступа : <http://www.cloudbus.org/cloudsim/> (дата обращения 15.06.2017).

**Elena Desyatirikova** – Professor, Doctor of economic Sciences, Department of Information Systems, Faculty of Computer Science, Voronezh State University.  
E-mail: science2000@ya.ru

**Moussa Hajali** – Professor, Doctor of Computer Science, Department of Information Systems and Networks, Faculty of Informatics, Damascus University, Syria.  
E-mail: hajalim@hotmail.com

**Almothana Khodar** – Master of Technical Sciences, Postgraduate student of the Department of Information Systems, Faculty of Computer Science, Voronezh State University.  
E-mail: Mothana-sy@hotmail.com

**Osama Alkaadi** – Master of Technical Sciences, Postgraduate student of the Department of Information Systems, Faculty of Computer Science, Voronezh State University.  
E-mail: oalkadee@gmail.com

**Hayyan Rajab** – Master of Technical Sciences, Postgraduate student of the Department of Information Systems and Networks, Faculty of Informatics, Tishrin University; Syria.  
E-mail: hayyan-r@hotmail.com