

УДК 004.45

МОДЕЛИРОВАНИЕ СИСТЕМЫ ПЛАНИРОВАНИЯ РАСПРЕДЕЛЕННОГО ВЫСОКОПРОИЗВОДИТЕЛЬНОГО ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА

Д.Ю. Астриков, Д.А. Кузьмин, А.И. Панасюк
Сибирский Федеральный Университет

Эффективное использование вычислительных ресурсов в центрах высокопроизводительных вычислений является актуальной проблемой, важный аспект которой – правильный подход к распределению задач и планированию нагрузки на вычислительные узлы. В статье рассматриваются основные алгоритмы распределения ресурсоемких пользовательских заданий внутри гетерогенных вычислительных систем, такие как First Come First Served, Shortest/Longest Job First, Backfilling, Round-robin и другие. Оценка эффективности алгоритмов распределения проводилась на разработанной авторами статьи модели вычислительной системы, отражающей структуру существующего распределенного вычислительного комплекса СФУ. Исходные экспериментальные данные для моделирования составили реальные задачи, запущенные в вычислительной системе СФУ за последние несколько лет. В качестве платформы для имитации запуска задач были использованы платформы SimGrid и Alea. Как результат – достигнуты максимально приближенные к реальным условия для проведения экспериментов по оценке эффективности применения различных алгоритмов распределения пользовательских задач. Анализ результатов проведенных экспериментов позволяет предложить решения по модернизации существующей вычислительной инфраструктуры.

Ключевые слова: Grid, распределенные системы, Torque, Maui, SimGrid, GridSim, MicroGrid.

Введение

В подавляющем большинстве случаев эффективное использование ресурсов вычислительных систем возможно только при условии применения специализированного программного обеспечения, предназначенного для распределения задач между узлами системы – менеджеров ресурсов и планировщиков очередей заданий. В связи с этим актуальной является задача разработки правил и алгоритмов распределения и запуска заданий, наиболее эффективных в рамках используемой вычислительной инфраструктуры.

1. Проблема распределения заданий в вычислительных системах

В целях обеспечения эффективности использования суперкомпьютерных систем, применяемых для расчетов ресурсоемких заданий, в подавляющем большинстве случаев требуется разработать специализированную политику распределения заданий между вычислительными узлами системы. В общем случае такая политика представляет собой набор правил, параметров и приоритетов, согласно которым задания пользователей распределяются между вычислительными узлами системы, наиболее подходящими с точки зрения наличия необходимого количества аппаратных ресурсов, установленного на узле программного обеспечения и других условий.

Однако нужно понимать, что единственно верного и универсального решения в области распределения задач не существует. Эффективность применения того или иного подхода тесно связана с различными условиями и ограничениями, которые неизбежно возникают в процессе эксплуатации гетерогенных вычислитель-

ных систем, таких как различная скорость передачи данных между вычислительными узлами, разнородность аппаратной и программной конфигурации этих узлов, наличие на узлах необходимого проблемно-ориентированного программного обеспечения и др. В связи с этим в качестве управляющего звена в таких системах используются менеджеры ресурсов и планировщики заданий, которые отвечают за правильное распределение задач в рамках сложившихся в данный момент условий. Планировщики заданий (Maui, Moab и др.) распределяют задачи между вычислительными узлами таким образом, чтобы как можно эффективнее задействовать доступные для использования вычислительные ресурсы. Проектирование и настройка наиболее подходящих политик и алгоритмов распределения задач являются крайне актуальной и приоритетной задачей, решение которой заключается в сравнительном анализе скорости выполнения задач, уровня утилизации ресурсов и других факторов при использовании того или иного алгоритма.

2. Анализ эффективности различных алгоритмов распределения заданий

Существует несколько основных алгоритмов, которые являются основой для любой политики распределения заданий в вычислительных системах.

- **FCFS (First Come First Served).** Расчет задач, поставленных в очередь раньше остальных, будет осуществлен в первую очередь. По завершении выполнения задача покидает очередь. Это наиболее простой алгоритм распределения заданий [1].

- **SJF (Shortest Job First).** Первой рассчитывается задача, для выполнения которой требуется меньше всего процессорного времени. Подход **LJF (Longest Job First)**, напротив, берет за основу правило: задача, требующая больше всего процессорного времени, рассчитывается первой [2].

- **RR (Round Robin).** Задачи распределяются в системе по круговому циклу. При использовании внутри небольших вычислительных систем этот алгоритм схож с FCFS.

Чаще всего эти алгоритмы применяются совместно с дополнительными алгоритмами распределения задач. Среди таких алгоритмов стоит выделить алгоритмы **BF (Backfilling)** и **FS (Fairshare)**.

Алгоритм Backfilling позволяет эффективнее использовать вычислительные ресурсы, «на лету» меняя очередность запуска заданий, запуская задачи с более низким приоритетом при условии, что они не задерживают запуск высокоприоритетных задач [3]. Существуют модификации этого алгоритма – Conservative Backfilling и Aggressive Backfilling.

При использовании алгоритмы Fairshare процессорное время равномерно распределяется между различными пользователями вычислительной системы.

Для того чтобы оценить эффективность работы этих алгоритмов, наиболее удобно использовать подход, заключающийся в разработке соответствующей модели вычислительной системы и последующей симуляции запуска различного вида вычислительных задач.

В качестве средств для проведения такого рода экспериментов можно использовать несколько программных решений. Среди них стоит отметить **SimGrid** [4], **GridSim** [5], **MicroGrid** [6], **Alea** [7]. Все эти пакеты предоставляют исследователю схожие возможности, но при этом довольно сильно отличаются с точки зрения удобства использования. Сравнение основных характеристик этих программных решений представлено в табл. 1.

В качестве средств симуляции запуска ресурсоемких заданий выбраны пакеты SimGrid и Alea в силу того, что они хорошо документированы и обладают развитыми средствами визуализации выходных данных.

Таблица 1 / Table 1

**Сравнение основных характеристик симуляторов SimGrid, GridSim,
MicroGrid и Alea**
**Comparison of the main characteristics of simulators SimGrid, GridSim,
MicroGrid and Alea**

Характеристика/ characteristic	SimGrid	GridSim	MicroGrid	Alea
Язык описания моделей / Models description lan- guage	XML	Java	XML	Java
Язык описания ал- горитмов распре- деления заданий / Language for desc- ribing algorithms assign tasks	C, Python, Ruby, Lua	Java	C	Java
Формат выходных данных / The for- mat of the output data	Paje trace file	Выходные данные сохра- няются в фор- мате CSV	—	Система облада- ет обширными возможностями в плане визуализа- ции результатов экспериментов
Наличие GUI / Availability a GUI	Нет	Нет	Нет	Да
Наличие докумен- тации / Availability of documentation	Проект обладает полноценной и актуальной документацией, большим коли- чеством приме- ров	Документации как таковой нет, вся информация представлена в виде приме- ров использо- вания	Система хорошо документирована	Система постро- ена на базе паке- та GridSim. Кро- ме того, пред- ставлена неболь- шая документа- ция на сайте проекта

За основу модели исследуемой вычислительной системы взят распределенный суперкомпьютерный комплекс Сибирского Федерального Университета. Грид СФУ используется научными группами для расчета ресурсоемких задач в таких областях научного знания, как физика, гидро- и газодинамика, молекулярная химия, математика и др. Локальная вычислительная Грид система СФУ доступна как из корпоративной сети СФУ, так и из глобальной сети Интернет. На рис. 1 показана схема этой системы.

Множество заданий, запускаемых на смоделированной вычислительной системе, формировалось путем анализа логов PBS Torque за прошедшие три года. Анализ производился с помощью специально разработанной программы, написанной на языке python. В результате, в исходное множество вошли 2334 задания, каждое из которых требовало для выполнения различное количество процессорного времени и оперативной памяти. Для проведения экспериментов использовались как полное множество заданий, так и случайным образом взятые «срезы» этого множества.

Также были проанализированы различные алгоритмы распределения заданий – SJF (shortest job first), LJF (longest job first), RR (round-robin), FCFS (first come first serve). При этом эти алгоритмы исследовались в условиях совместной работы

с алгоритмами Backfilling и Fairshare. В табл. 2 представлены результаты работы симулятора при использовании алгоритмов SJF и RR. В соответствующих колонках приведено время (в секундах) исполнения заданного множества задач при использовании алгоритмов SJF и Round-Robin. Алгоритмы LJF, FCFS показали крайне схожие с алгоритмом SJF результаты, поэтому в таблице они не приводятся.

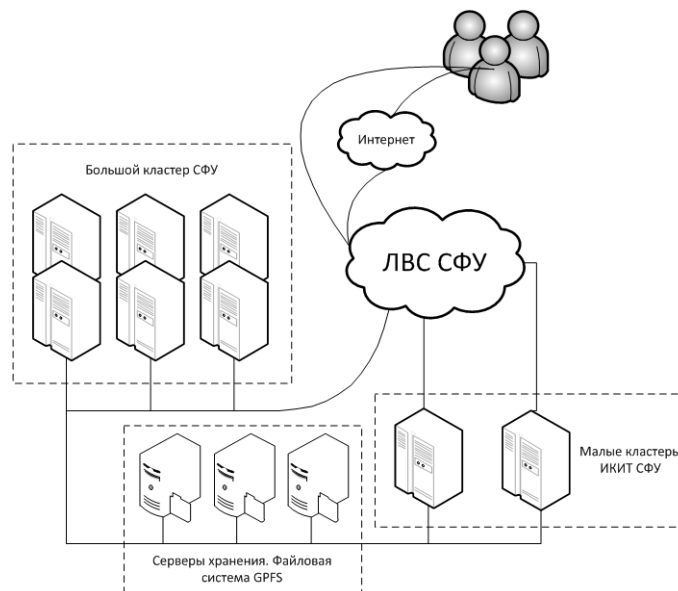


Рис. 1 – Схема локальной Грид-системы Сибирского Федерального Университета

Fig. 1 – Scheme of the local Grid system of Siberian Federal University

Таблица 2 / Table 2

Сравнение результатов работы симулятора на базе различных алгоритмов распределения

Число задач / Tasks number	Shortest Job First	Round-Robin
200	6.37976e+07	6.37976e+07
400	1.39462e+08	2.03258e+08
600	1.39539e+08	2.03496e+08
800	1.3963e+08	2.73934e+08
1000	1.39797e+08	2.96567e+08
1200	1.40139e+08	3.20913e+08
1400	1.40695e+08	3.43273e+08
1600	1.41422e+08	3.76413e+08
1800	1.41827e+08	3.77149e+08
2000	1.42527e+08	4.31322e+08

Однако лучшим образом проявили себя алгоритмы, работавшие совместно с алгоритмами Backfilling и Fairshare. На рис. 2 показаны график утилизации вычислительных ресурсов при использовании алгоритма FCFS + Agressive Backfilling в сравнении с аналогичным графиком при использовании алгоритма FCFS.

На приведенных графиках видно, что в случае применения политики Agressive Backfilling пиковая нагрузка на узлы (показана более темным цветом) сосредоточена

чена в более узком временном промежутке, чем в случае использования обычного алгоритма FCFS. Это означает, что система быстрее справилась с расчетом особенно ресурсоемких заданий, за счет более эффективного распределения нагрузки на вычислительные узлы.

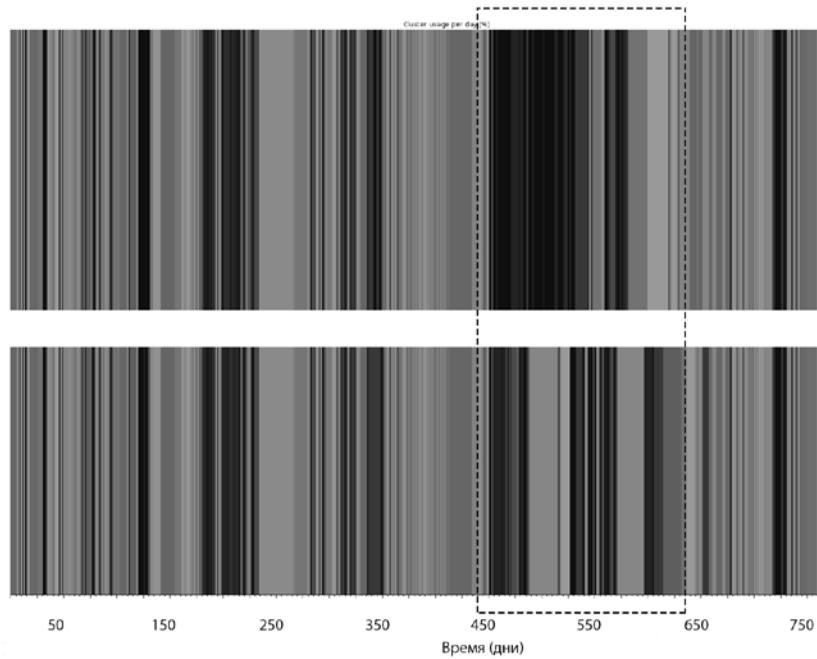


Рис. 2 – Утилизация вычислительных ресурсов при использовании алгоритмов FCFS + Agressive Backfilling (сверху) и FCFS (снизу)

Fig. 2 – Computing resources utilization using the FCFS + Agressive Backfilling algorithm (above) and FCFS (bellow)

Не менее показательными являются и графики использования CPU, представленные на рис. 3. Темно-серый график демонстрирует, что CPU в период пиковой нагрузки использовался более эффективно при применении алгоритма Agressive Backfilling – на протяжении всего этого периода загрузка CPU составляла практически 100 %.

На приведенных графиках можно выделить наиболее интересный период работы суперкомпьютерного комплекса, в течение которого наблюдалась крайне интенсивная нагрузка на вычислительные узлы. На рисунке этот период выделен пунктирной рамкой. В течение этого периода значительно возросло количество заданий в очереди, при этом количество ожидающих задач значительно превышало количество выполняющихся. Соотношение выполняющихся и ожидающих заданий показано на рис. 4.

В подобной ситуации Agressive Backfilling проявил себя наиболее выгодно, за счет того, что планировщик заданий способен изменять приоритеты запуска заданий, находящихся в очереди в зависимости от приоритетов поступающих в очередь новых заданий. Из графиков видно, что в первом случае количество ожидающих задач, при применении алгоритма Agressive Backfilling (показано темно-серым цветом), в 1,5 раза меньше, чем количество задач при применении алгоритма FCFS (показано светло-серым цветом).

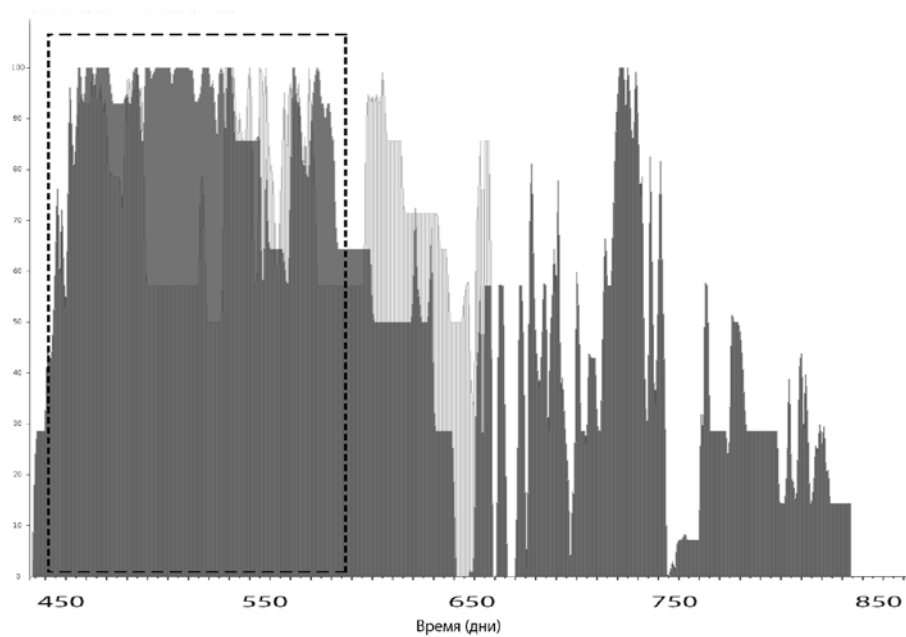


Рис. 3 – Задействование CPU при использовании алгоритмов FSFS + Agressive Backfilling (темно-серый цвет) и FCFS (светло-серый цвет)

Fig. 3 – CPU usage using the FCFS + Agressive Backfilling algorithm (dark-gray color) and FCFS (light-gray color)

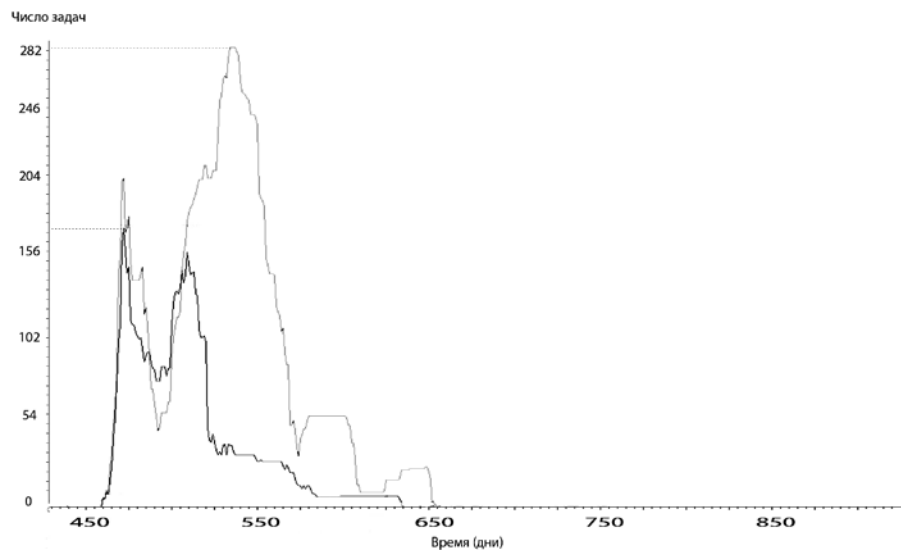


Рис. 4 – Соотношение количества ожидающих задач при использовании алгоритма Agressive Backfilling (показаны темно-серым цветом) и количества ожидающих задач при использовании алгоритма FCFS (показаны светло-серым цветом)

Fig. 4 – Number of waiting tasks using the Agressive backfilling algorithm (dark-gray color) and number of waiting tasks using the FCFS algorithm (light-gray color)

Заключение

На базе разработанной экспериментальной модели, максимально приближенной к существующей вычислительной инфраструктуре Сибирского Федерального Университета, проведен сравнительный анализ различных алгоритмов распределения ресурсоемких заданий в гетерогенных вычислительных системах. В качестве исходных данных использовались задания, реально выполнявшиеся на распределенных вычислительных ресурсах СФУ за последние 3 года.

Полученные в результате проведенных экспериментов данные позволили сформулировать рекомендации по использованию семейства алгоритмов FCFS и SJF совместно с алгоритмами семейства Backfilling (Agressive Backfilling, Easy Backfilling и др.) в целях повышения эффективности использования вычислительных ресурсов СФУ. В частности, использование алгоритма Backfilling способно привести к снижению количества ожидающих задач в 2 раза, что положительно скажется на общем времени, которое потребуется для расчета пользовательских задач.

ЛИТЕРАТУРА

1. **Li K.** Job scheduling and processor allocation for grid computing on metacomputers // Journal of Parallel and Distributed Computing. – 2005. – Vol. 65. – P. 1406–1418.
2. **Li H., Buyya R.** Model-Driven Simulation of Grid Scheduling Strategies // Proceedings of the Third IEEE International Conference on e-Science and Grid Computing. – Washington, 2007. – P. 287–294.
3. Документация планировщика заданий MAUI. – URL: <http://docs.adaptivecomputing.com/maui/8.2backfill.php> (дата обращения: 10.05.2014).
4. Документация платформы моделирования вычислительных систем SimGrid. – URL: <http://simgrid.gforge.inria.fr/documentation.html> (дата обращения: 10.05.2014).
5. Документация платформы моделирования вычислительных систем GridSim. – URL: <http://www.buyya.com/gridsim/> (дата обращения: 10.05.2014).
6. Документация платформы моделирования вычислительных систем MicroGrid. – URL: <http://www-csag.ucsd.edu/projects/grid/mgrid2-user.html> (дата обращения: 10.05.2014).
7. **Klusáček D. and Rudová H.** Alea 2: Job Scheduling Simulator // Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010), ICST, 2010, Malaga, Spain, March 16–18, 2010. – Brussels, Belgium: ICST, 2010. – Art. 61.

SIMULATION OF A SCHEDULING SYSTEM OF THE DISTRIBUTED HIGH-PERFORMANCE COMPUTING SYSTEM

Astrikov D.Y., Kuzmin D.A., Panasyuk A.I.
Siberian Federal University, Krasnoyarsk, Russia

An effective use of computing resources in high-performance computing centers is a vital problem. The main aspect of this problem is a rational approach to task planning and distribution across computing nodes. This article covers the main algorithms of huge task distribution in large computing systems, such as First Come First Served, Shortest/Longest Job First, Backfilling, Round-robin, etc. To evaluate the effectiveness of these algorithms the authors have developed a model of a computer system showing the structure of the available SFU computing system. As initial experimental data the authors used a variety of tasks which have been run in the SFU computing system over the past few years. The SimGrid and Alea platforms are used to simulate starting task execution. The conditions for the evaluation of the effectiveness of various algorithms of user task distribution closest to real conditions have been found. Using experimental results obtained, the authors have proposed some solutions aimed at modernizing the existing computing infrastructure.

Keywords: Grid, distributed system, Torque, Maui, SimGrid, GridSim, MicroGrid.

REFERENCES

1. Li K. Job scheduling and processor allocation for grid computing on metacomputers. *Journal of Parallel and Distributed Computing*, 2005, vol. 65, pp. 1406-1418.
2. Li H., Buyya R. Model-Driven Simulation of Grid Scheduling Strategies. *Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*, Washington, 2007, pp. 287-294.
3. MAUI Scheduler official documentation. Available at: <http://docs.adaptivecomputing.com/maui/8.2backfill.php> (accessed 10.05.2014)
4. SimGrid simulation tools official documentation. Available at: <http://simgrid.gforge.inria.fr/documentation.html> (accessed 10.05.2014)
5. GridSim simulation tools official documentation. Available at: <http://www.buyya.com/gridsim/> (accessed 10.05.2014)
6. MicroGrid simulation tools official documentation. Available at: <http://www-csag.ucsd.edu/projects/grid/mgrid2-user.html> (accessed 10.05.2014)
7. Klusáček D. and Rudová H. Alea 2: Job Scheduling Simulator. *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010)*, ICST, 2010, Malaga, Spain, March 16–18, 2010. Brussels, Belgium, ICST, 2010, art. 61.

СВЕДЕНИЯ ОБ АВТОРАХ



Астриков Дмитрий Юрьевич – родился в 1989 году, аспирант второго года обучения Сибирского Федерального Университета. Область научных интересов: информационные технологии, программирование. Опубликовано 8 научных работ. (Адрес: 660130, Российская Федерация, Красноярск, Елены Стасовой, 40А. Email: astrikov.d@gmail.com).

Astrikov Dmitry Yuirievich (b. 1989) – second year postgraduate student, Siberian Federal University. His research interests are currently focused on IT and programming. He is author of 8 scientific papers. (Address: Russian Federation, Krasnoyarsk, 40A Eleni Stasovoy St., Email: astrikov.d@gmail.com).



Кузьмин Дмитрий Александрович – родился в 1968 году, канд. техн. наук, заведующий кафедрой «Высокопроизводительные вычисления» в Сибирском Федеральном Университете. Область научных интересов: высокопроизводительные системы.

Kuzmin Dmitry Alexandrovich (b. 1968) – PhD, head of high-performance computing department. His research interests are currently focused on high performance computing.



Панасюк Александр Иванович – родился в 1989 году, аспирант третьего года обучения Сибирского Федерального Университета. Область научных интересов: информационные технологии, программирование.

Panasyuk Alexandr Ivanovich (b. 1989) – third-year postgraduate student, Siberian Federal University. His research interests are currently focused on IT and programming.

Статья поступила 31 июля 2014 г.
Received July 31, 2014

To Reference:

Astrikov D.Yu., Kuz'min D.A., Panasyuk A.I. Modelirovanie sistemy planirovaniya raspredelen-nogo vysokoproizvoditel'nogo vychislitel'nogo kompleksa [Simulation of a scheduling system of the distributed high-performance computing system]. *Doklady Akademii Nauk Vysshei Shkoly Rossiiskoi Federatsii – Proceedings of the Russian Higher School Academy of Sciences*, 2014, no. 2-3 (23-24), pp. 34-41.