Июнь, № 4 **Физико-математические науки** **2014**

УДК 004.75

ЕВГЕНИЙ ЕВГЕНЬЕВИЧ ИВАШКО
кандидат физико-математических наук, научный сотрудник, Институт прикладных математических исследований Карельского научного центра РАН, старший преподаватель кафедры прикладной математики и кибернетики математического факультета, Петрозаводский государственный университет (Петрозаводск, Российская Федерация)
ivashko@krc.karelia.ru

МАРИЯ СЕРГЕЕВНА ИЦКАРЬ
студент 4-го курса математического факультета, Петрозаводский государственный университет (Петрозаводск, Российская Федерация)
ickar@cs.karelia.ru

МАРИЯ ВЛАДИМИРОВНА КОЛЧИНА
студент 4-го курса математического факультета, Петрозаводский государственный университет (Петрозаводск, Российская Федерация)
kolchina@cs.karelia.ru

ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ПРИЛОЖЕНИЯ ПО АНАЛИЗУ ДАННЫХ НА БАЗЕ BOINC-ГРИД*

Представлены результаты тестирования производительности программного обеспечения (ПО), предназначенного для нахождения ассоциативных правил в больших наборах данных с помощью BOINC-грид. Приведены результаты экспериментов по выявлению зависимости времени выполнения вычислений от количества подзаданий. Для класса задач обработки больших наборов данных описана структура накладных временных расходов, привносимых BOINC при организации вычислительного процесса.

Ключевые слова: тестирование, производительность, BOINC

ВВЕДЕНИЕ

Стремительный научно-технический прогресс, развитие информационных технологий, появление электронной коммерции и социальных сетей, а также развитие технологий записи и хранения данных привели к бурному росту объемов собираемой и анализируемой информации. В связи с этим наблюдается повышенный интерес к технологиям класса Big Data. Согласно [3], Big Data – это серия подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объемов и значительного многообразия для получения воспринимаемых человеком результатов. Накопленная информация для многих организаций является важным активом, однако обрабатывать ее и извлекать из нее пользу с каждым днем становится все сложнее и дороже.

Для целей интеллектуальной обработки данных были разработаны специальные алгоритмы и подходы, объединенные термином Data Mining. Data Mining – это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности [6]. Одним из популярных методов Data Mining является нахождение ассоциативных правил. Ассоциативные правила позволяют выявлять скрытые закономерности между связанными событиями.

Анализ больших объемов данных требует привлечения специальных технологий и средств выполнения высокопроизводительных вычислений. Одним из таких средств является BOINC. BOINC – это открытая программная платформа для организации систем распределенных вычислений, разработанная в университете Беркли [9]. Платформа BOINC отличается простотой в установке, настройке и администрировании, а также обладает хорошими возможностями по масштабируемости, простоте подключения вычислительных узлов, использованию дополнительного ПО, интеграции с другими грид-системами и др.

Платформа BOINC имеет архитектуру «клиент-сервер», при этом клиентская часть может работать на компьютерах с различными аппаратными и программными характеристиками. Ключевым объектом системы является проект – автономная сущность, в рамках которой производятся распределенные вычисления. BOINC-сервер поддерживает одновременную работу большого числа независимых проектов; каждый вычислительный узел может одновременно выполнять вычисления для нескольких BOINC-проектов [2].

При расчетах на BOINC большое внимание уделяется оптимизации приложений, так как эффект масштаба на большом числе вычислительных узлов приводит к большим потерям производительности неоптимизированных BOINC-приложений. Вынужденные потери производительности связаны также с необходимостью дублирования подзаданий для обеспечения достоверности результатов, а также со сложностью планирования подзаданий.

Вопросам планирования подзаданий посвящены, например, статьи [7] и [8]. В работе [8] рассматриваются три подхода к выбору ресурсов, которые позволяют достичь наилучшей производительности: приоритезация ресурсов, исключение ресурсов и дублирование заданий. В частности, было выявлено, что установка приоритетов в использовании ресурсов является малоэффективным подходом. Более действенным методом оказывается дублирование подзаданий. Данный подход позволяет исключить из вычислений ненадежные хосты, которые часто являются причиной задержки расчетов.

В работе [7] представлены сочетания четырех политик планирования ресурсов:

- планирование ресурсов центрального процессора: какие из подзаданий запускать на расчет в первую очередь;
- запрос подзаданий: когда запрашивать у проекта новые подзадания, у какого проекта их запрашивать и как много;
- отправка подзаданий: какие подзадания серверу следует отправлять клиенту в ответ на запрос;
- оценка времени завершения: как оценить время, оставшееся для расчета подзадания.

В результате были выявлены наиболее эффективные из них: например, при определенном сочетании политик планирования можно добиться снижения на 90 % накладных расходов, связанных с запросом подзадания.

В статье [10] представлены результаты тестирования платформы RT-BOINC, предназначенной для проведения вычислений в режиме реального времени. На примере игр (шахматы и го) было показано, что данная платформа превосходит BOINC по показателям времени отклика и масштабируемости. Успех достигается с помощью установки таймера дедлайна и создания механизма контроля входных данных. Благодаря специальным политикам и планированию процесса выполнения подзаданий достигается снижение накладных расходов, привносимых BOINC для организации процесса расчетов. Однако изучению структуры и объемов этих накладных расходов не уделяется большого внимания.

В современном мире ПО играет огромную роль практически во всех сферах человеческой деятельности, будь то наука, образование, медицина или промышленность. Как правило, для широко используемых программ, обрабатывающих большие объемы данных или имеющих требования к срокам получения результата, проводится оценка (или тестирование) производительности. Цель тестирования производительности – определить, как быстро работает система или ее часть под определенной нагрузкой. Данный вид тестирования служит для проверки и подтверждения таких качеств системы, как:

- масштабируемость;
- надежность;
- потребление ресурсов [4].

Тестирование позволяет выявлять и устранять причины потерь производительности, что особенно важно для высокомасштабируемых программ, выполняемых в рамках грид.

Представленная статья посвящена исследованию накладных расходов, привносимых платформой BOINC при организации процесса вычислений. За основу взята работа [2], в которой описано приложение по анализу больших массивов данных в гетерогенной Desktop Grid на базе BOINC: описана реализация алгоритма, предназначенного для использования в распределенной среде, и представлены результаты экспериментов по оценке производительности разработанного ПО на тестовых наборах данных. Задача представленного исследования – провести тестирование производительности программной системы, оценить время нахождения ассоциативных правил при различных характеристиках наборов данных и вычислительных узлов, а также определить структуру накладных расходов.

ОПИСАНИЕ ТЕСТИРУЕМОЙ СИСТЕМЫ

Тестируемая программная система представляет собой ПО, реализующее алгоритм Partition на базе грид-платформы BOINC. Данный алгоритм предназначен для решения задачи поиска ассоциативных правил [1].

Работа алгоритма проходит в три этапа, два из которых выполняются параллельно на вычислительных узлах грид-сети. На завершающем этапе происходит объединение промежуточных результатов. На рис. 1 представлена общая схема выполнения алгоритма Partition в BOINC.

Рассмотрим работу алгоритма Partition в BOINC подробнее.

Этап I. Программа получает на входе исходный файл транзакций (например, записей за определенный период о покупках клиентов в супермаркете или платежах в банке). Файл разбивается на заданное количество частей (подзаданий), затем планировщик BOINC-сервера распределяет подзадания клиентам (вычислительным узлам) BOINC-грид. Затем каждый BOINC-клиент загружает с сервера входные файлы подзаданий, находит часто встречающиеся наборы в имеющихся исходных данных, после чего загружает выходные файлы на сервер и отчитывается о выполнении подзадания.

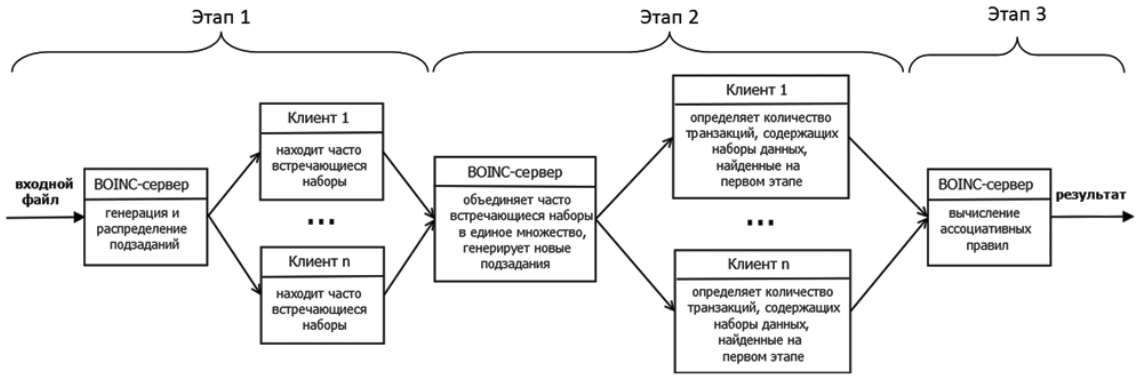

Рис. 1. Алгоритм Partition в BOINC

Этап II. Выходные файлы проверяются сервером, а затем все полученные часто встречающиеся наборы объединяются в единое множество. На основе этого множества формируются новые подзадания, которые снова распределяются между клиентами. Клиенты, в свою очередь, определяют количество транзакций, содержащих наборы данных, найденные на первом этапе, и отправляют результаты на сервер.

Этап III. На основе результатов, полученных от клиентов, на сервере вычисляются ассоциативные правила.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Задача тестирования заключалась в оценке времени нахождения ассоциативных правил при различных характеристиках наборов данных и вычислительных узлов и определении структуры накладных расходов. Соответственно тестирование проводилось в два этапа.

1. *Зависимость времени выполнения от количества подзаданий.*

Увеличение числа подзаданий позволяет повысить масштабируемость приложения. Кроме того, увеличение числа подзаданий приводит к уменьшению размеров исходных данных, а значит, снижается время обработки одного подзадания, что помогает нивелировать разницу в производительности узлов гетерогенной грид. Однако с ростом числа подзаданий увеличивается доля накладных расходов, привносимых BOINC при организации процесса вычислений. На данном этапе тестирования была поставлена задача определения характера зависимости производительности приложения от того, на сколько частей разбивается исходный файл (то есть от количества подзаданий).

Эксперименты проводились на одном вычислительном узле (клиенте). В качестве исходных данных использовались файлы различного объема: 1 Гб, 5 Гб и 10 Гб. Каждый файл был разбит на фиксированное количество частей – от 10 до 70. Результаты экспериментов представлены на рис. 2.

Из рис. 2 видно, что с увеличением числа подзаданий увеличивается время, необходимое для выполнения расчетов. Причем в зависимости от размера исходного файла это увеличение поначалу незначительно, но при достижении некоторого порога начинает расти достаточно быстро. Происходит это из-за того, что при увеличении числа подзаданий все более существенными становятся накладные расходы, которые платформа BOINC вносит для организации вычислений. Таким образом, увеличивать число подзаданий необходимо в соответствии с масштабом сети и размером исходного файла транзакций.

2. *Оценка накладных расходов.*

Второй этап тестирования связан с оценкой накладных расходов времени, требуемых для организации процесса вычислений в грид-сети. Для этого прежде всего необходимо определить, из чего складываются накладные расходы.

Расчет подзадания со стороны клиента состоит из следующих этапов:

- запрос подзадания от сервера;
- время ожидания (от ответа планировщика до начала скачивания подзадания);
- загрузка подзадания;
- проведение расчетов;
- время ожидания (от окончания расчетов до начала загрузки результата);
- отправка результата на сервер;
- время ожидания (от окончания отправки результата до ответа планировщика о получении результата).

Для того чтобы выяснить, сколько времени занимает каждый из этапов, был проведен ряд экспериментов. Вычисления проводились на

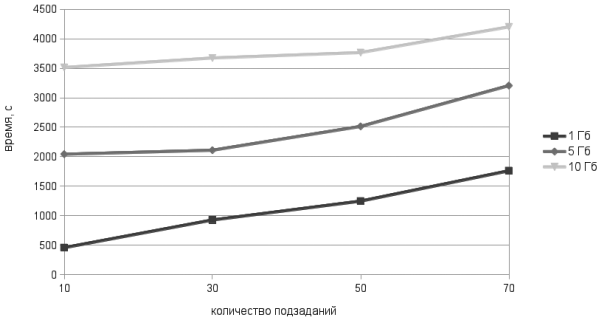

Рис. 2. Время расчета подзаданий

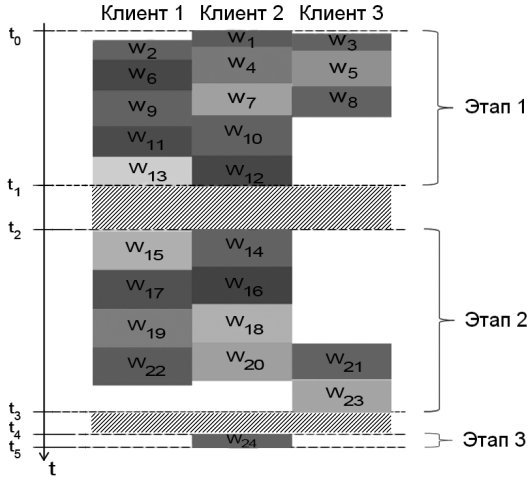

Рис. 3. Хронометраж расчета подзаданий тремя клиентами

трех вычислительных узлах (клиентах). В качестве исходных данных использовался файл объемом 1 Гб, разделенный на 10 частей (подзаданий). На рис. 3 представлен результат анализа лог-файлов клиентов. Каждый блок обозначает отдельное подзадание, штриховкой выделен этап генерации новых подзаданий на сервере. Так как исходный файл делился на 10 частей, в итоге должно было получиться 21 подзадание. Но из-за того, что на первом этапе результаты расчетов некоторых подзаданий были утеряны и рассчитывались повторно, в общей сложности получилось 24 подзадания.

Оценка структуры накладных расходов также проводилась на основе анализа лог-файлов клиентов, для чего была разработана специальная программа. Результаты представлены на рис. 4. На рисунке видно, что основное время выполнения анализа данных занимают непосредственно расчеты, но значительную часть составляет время, затрачиваемое на загрузку подзаданий и ожидание ответа планировщика о принятом результате.

На рис. 5а показана структура накладных расходов в процентном соотношении. Как показали дальнейшие эксперименты, при увели-

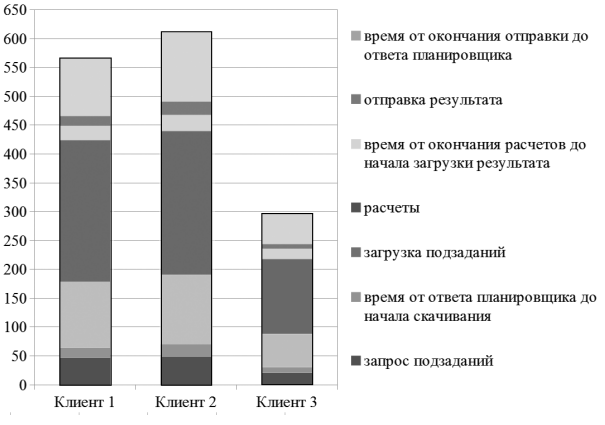

Рис. 4. Структура накладных расходов

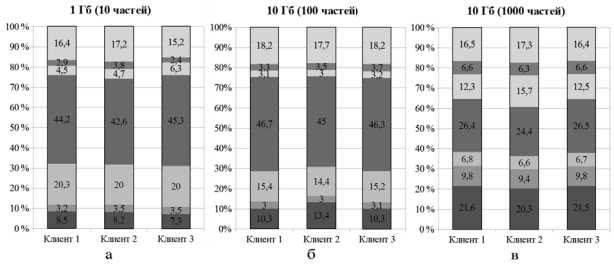

Рис. 5. Структура накладных расходов в процентном выражении

чении на порядок и размера файла (до 10 Гб), и количества подзаданий (до 100) структура накладных расходов сохраняется (см. рис. 5б), так как размер каждого подзадания остается тем же, что и на рис. 5а.

В следующем эксперименте исходный файл объемом 10 Гб был разделен уже на 1000 частей (то есть по сравнению с предыдущим экспериментом объем файла остался неизменным, а количество подзаданий увеличилось в 10 раз). Эксперимент показал, что при увеличении числа подзаданий структура накладных расходов изменяется довольно существенно. Структуру накладных расходов, полученную в результате проведенного эксперимента, можно увидеть на рисунке 5в.

На основании проведенных экспериментов был сделан вывод о том, что в структуре накладных расходов существенную часть занимает время, затрачиваемое на взаимодействие клиента и сервера. При этом при пропорциональном увеличении как объема файла, так и числа подзаданий структура накладных расходов сохраняется.

ЗАКЛЮЧЕНИЕ

В работе представлены результаты тестирования производительности программного обеспечения, предназначенного для нахождения ассоциативных правил в больших наборах данных с помощью BOINC-грид. При расчетах на BOINC большое внимание уделяется оптимизации приложений, так как эффект масштаба на большом числе вычислительных узлов приводит к большим потерям производительности неоптимизированных BOINC-приложений. Увеличение числа подзаданий позволяет повысить масштабируемость приложения. Кроме того, оно приводит к уменьшению размеров исходных данных, а значит, снижается время обработки одного подзадания, что помогает нивелировать разницу в производительности узлов гетерогенной грид. Однако с ростом числа подзаданий увеличивается доля накладных расходов, привносимых BOINC при организации процесса вычислений.

В ходе работы было проведено тестирование, направленное на выявление зависимости времени выполнения вычислений от количества подзаданий. С увеличением числа подзаданий уве-

личивается время, необходимое для выполнения расчетов, что обусловлено ростом накладных временных расходов. Представлена структура накладных расходов для класса задач обработки больших наборов данных. В структуре накладных расходов существенную часть занимает время, затрачиваемое на взаимодействие клиента и сервера BOINC.

* Работа выполнена в рамках Программы стратегического развития ПетрГУ и поддержана грантами РФФИ 12–07–31147 «мол_а» и 13–07–00008 «а».

СПИСОК ЛИТЕРАТУРЫ

1. Головин А. С. Реализация алгоритмов Data Mining в гетерогенной грид на базе платформы BOINC: Дис. … магистра прикладной математики и информатики / Петрозаводский государственный университет. Петрозаводск, 2012. 42 с.
2. Ивашко Е. Е., Головин А. С. Вычислительная эффективность BOINC-GRID // Proceedings of 2nd International Conference on High Performance Computing HPC-UA. 2012. С. 183–187.
3. Термины и определения [Электронный ресурс]. Режим доступа: http://www.ipiran.ru/niap/index_3.html
4. Тестирование и анализ производительности с помощью сервера приложений WebSphere [Электронный ресурс]. Режим доступа: http://www.ibm.com/developerworks/ru/library/wes-1208_hare
5. Шнайер Б. Секреты и ложь. Безопасность данных в цифровом мире. СПб.: Питер, 2003. 368 с.
6. Data Mining – добыча данных [Электронный ресурс]. Режим доступа: http://www.basegroup.ru/library/methodology/data_mining
7. Kondo D., Anderson D. P., McLeod J. Performance Evaluation of Scheduling Policies for Volunteer Computing // E-SCIENCE ’07 Proceedings of the Third IEEE International Conference on e-Science and Grid Computing. 2007. P. 415–422.
8. Kondo D., Chien A. A., Casanova H. Resource Management for Rapid Application Turnaround on Enterprise Desktop Grids // ACM/IEEE Conf. Supercomputing (SC ’04). 2004. P. 17–19.
9. Open-source software for volunteer computing and grid computing [Электронный ресурс]. Режим доступа: http://boinc.berkeley.edu
10. Yi S., Jeannot E., Kondo D., Anderson D. P. Towards Real-Time, Volunteer Distributed Computing // CCGRID-2011. P. 154–163.

Ivashko E. E., Institute of Applied Mathematical Research, Karelian Research Centre of RAS, Petrozavodsk State University (Petrozavodsk, Russian Federation)
Itskar’ M. S., Petrozavodsk State University (Petrozavodsk, Russian Federation)
Kolchina M. V., Petrozavodsk State University (Petrozavodsk, Russian Federation)

PERFORMANCE TESTING OF BOINC-BASED DATA ANALYSIS SOFTWARE

Results of the software performance testing designed for finding association rules in large data sets using BOINC-grid are presented. The results of experiments detecting computing runtime dependent on the number of sub-tasks are presented. The structure of temporary overhead costs brought by BOINC during computing processing is described for the class of large data sets aimed at processing problems.

Key words: testing, performance, BOINC

REFERENCES

1. Golovin A. S. *Realizatsiya algoritmov Data Mining v geterogennoy grid na baze platformy BOINC. Diss. ... magistra tekhnicheskikh nauk* [Implementation of Data Mining algorithms in a heterogeneous grid based on a BOINC platform]. Petrozavodsk, 2012. 42 p.
2. Ivashko E. E., Golovin A. S. Computational efficiency of BOINC-GRID [Vychislitel’naya effektivnost’ BOINC-GRID]. *Proceedings of 2nd International Conference on High Performance Computing HPC-UA*. 2012. P. 183–187.
3. *Terminy i opredeleniya* [Terms and definitions]. Available at: http://www.ipiran.ru/niap/index_3.html
4. *Testirovanie i analiz proizvoditel’nosti s pomoshch’yu servera prilozheniy WebSphere* [Testing and analysis using an application server WebSphere]. Available at: http://www.ibm.com/developerworks/ru/library/wes-1208_hare
5. Shnayer B. *Sekrety i lozh’. Bezopasnost’ dannykh v tsifrovom mire* [Secrets and lies. Digital security in a networked world]. St. Petersburg, Piter Publ., 2003. 368 p.
6. *Data Mining – dobycha dannykh* [Data mining]. Available at: http://www.basegroup.ru/library/methodology/data_mining/
7. Kondo D., Anderson D. P., McLeod J. Performance Evaluation of Scheduling Policies for Volunteer Computing // E-SCIENCE ’07 Proceedings of the Third IEEE International Conference on e-Science and Grid Computing. 2007. P. 415–422.
8. Kondo D., Chien A. A., Casanova H. Resource Management for Rapid Application Turnaround on Enterprise Desktop Grids // ACM/IEEE Conf. Supercomputing (SC ’04). 2004. P. 17–19.
9. Open-source software for volunteer computing and grid computing. Available at: http://boinc.berkeley.edu
10. Yi S., Jeannot E., Kondo D., Anderson D. P. Towards Real-Time, Volunteer Distributed Computing // CCGRID-2011. P. 154–163.

Поступила в редакцию 09.01.2014