

На правах рукописи

Никитина Наталья Николаевна

**Математические модели управления заданиями  
и защиты информации  
в высокопроизводительных вычислительных системах**

05.13.18 – Математическое моделирование, численные методы и комплексы программ

**АВТОРЕФЕРАТ**  
диссертации на соискание ученой степени  
кандидата технических наук

Петрозаводск – 2014

Работа выполнена в *федеральном государственном бюджетном учреждении науки Институте прикладных математических исследований Карельского научного центра Российской академии наук.*

Научный руководитель:

**Мазалов Владимир Викторович,**  
доктор физико-математических наук,  
профессор

Официальные оппоненты:

**Буре Владимир Мансурович,**  
доктор технических наук, доцент, профессор кафедры математической теории игр и статистических решений факультета прикладной математики — процессов управления Санкт-Петербургского государственного университета

**Гуртов Андрей Валерьевич,**  
кандидат технических наук, ведущий научный сотрудник кафедры распределенных и мобильных облачных систем Хельсинкского института информационных технологий

Ведущая организация:

федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук

Защита состоится «26» декабря 2014 года в 12 часов на заседании диссертационного совета Д 212.190.03 при ФГБОУ ВПО «Петрозаводский государственный университет», по адресу: 185910, г. Петрозаводск, пр. Ленина, 33.

С диссертацией можно ознакомиться в научной библиотеке Петрозаводского государственного университета и на сайте [petrsu.ru](http://petrsu.ru).

Автореферат разослан «\_\_\_\_\_» \_\_\_\_\_ 2014 г.

Отзывы и замечания по автореферату в двух экземплярах, заверенные печатью, просьба высылать по вышеуказанному адресу на имя ученого секретаря диссертационного совета.

Ученый секретарь  
диссертационного совета

*Воронов Роман Владимирович*

## Общая характеристика работы

**Актуальность темы.** В ряде отраслей фундаментальной и прикладной науки возникают задачи, требующие выполнения значительных объемов вычислений, а также обработки, хранения, передачи и визуализации больших объемов данных. Для решения таких задач используются высокопроизводительные вычислительные системы, значительно превосходящие традиционные вычислительные средства по техническим характеристикам. Для эффективного использования высокопроизводительной вычислительной системы необходимо управлять очередью заданий, то есть определять порядок их запуска и распределять между ними вычислительные ресурсы — например, процессорное время, оперативную память, дисковое пространство.

Критериями эффективности при управлении очередью заданий могут являться общее время выполнения набора заданий, среднее время ожидания задания в очереди, средняя загрузка системы в единицу времени и др., а также комбинации таких критериев. Выбор характеристик заданий и критериев эффективности во многом зависит от особенностей операционной системы и архитектуры вычислительной системы в целом, а также от специфики решаемых задач. В связи с этим разработке и оценке алгоритмов планирования заданий посвящено множество исследований на протяжении нескольких десятков лет.

Эффективное управление ресурсами вычислительной системы включает в себя обеспечение безопасности информации. Сетевая атака типа DoS (от англ. Denial of Service — отказ в обслуживании) заключается в создании таких условий, в которых вычислительная система становится не в состоянии своевременно обслуживать все задания. При этом алгоритм управления заданиями, функционирующий в отсутствие атаки, становится малоэффективным или практически неприменимым.

**Цель диссертационной работы** заключается в построении и исследовании свойств математических моделей управления ресурсами высокопроизводительных вычислительных систем и защиты информации в них. Достижение поставленной цели требует решения следующих **задач**:

1. Задача повышения эффективности алгоритма Backfill для управления заданиями на вычислительном кластере. В качестве критериев эффективности принимаются количество ошибок при применении алгоритма и изменение среднего времени ожидания задания в очереди.
2. Задача минимизации нагрузки на сервер в централизованной системе распределенных вычислений типа Desktop Grid.
3. Задача определения момента начала DoS-атаки на вычислительную систему с последующим изменением протокола обслуживания заданий.

**Методы исследований.** В диссертационной работе применяются методы теории вероятностей, теории оптимизации и теории некооперативных игр.

Для имитационного моделирования и программной реализации разработанных алгоритмов применяются методы вычислительной математики и прикладного программирования.

**Научная новизна работы** заключается в следующем:

1. Разработана модификация алгоритма Backfill с принятием решения на основе аналитического выражения вероятности ошибки.
2. Построена теоретико-игровая модель управления заданиями в централизованной системе распределенных вычислений, предназначенной для проведения виртуального скрининга лекарств. Найдены аналитические выражения оптимальных стратегий и функций выигрыша узлов.
3. Найдено аналитическое решение разностного уравнения, полученного целочисленным приближением уравнения, описывающего динамику изменения среднего количества ошибок I или II типа при обнаружении момента DoS-атаки на вычислительную систему методом кумулятивных сумм.
4. Разработан алгоритм обслуживания заданий в присутствии DoS-атаки, позволяющий выбирать из очереди регулярные (ассоциированные с нормальной активностью пользователей) задания, игнорируя искусственные (ассоциированные с атакой).

**Практическая значимость работы** заключается в следующем:

1. Реализована программа обслуживания заданий с применением модифицированного алгоритма Backfill.
2. Реализована программа поиска решений задачи минимизации нагрузки на сервер в системе распределенных вычислений методом прямого перебора.
3. Реализован программный пакет для управления заданиями на основе построенной модели, предназначенный для системы распределенных вычислений. Программный пакет апробирован в системе, реализованной в ходе выполнения совместного научно-исследовательского проекта между Институтом прикладных математических исследований КарНЦ РАН и Любекским Институтом экспериментальной дерматологии при университете г. Любек (Германия).
4. Реализована программа обслуживания заданий в присутствии DoS-атаки, допускающая использование различных видов и параметров распределений регулярного и искусственного потоков заданий. Программа прошла государственную регистрацию в Федеральной службе по интеллектуальной собственности, патентам и товарным знакам.

На защиту выносятся следующие **положения**:

1. Разработана модификация алгоритма Backfill для управления заданиями на вычислительном кластере.
2. Предложена и исследована теоретико-игровая модель управления заданиями в централизованной системе распределенных вычислений, предназначенной для проведения поиска лекарств.
3. Разработан метод кумулятивных сумм обнаружения момента начала DoS-атаки в потоке заданий, характеризуемом распределением Бернулли.
4. Предложен и реализован протокол обслуживания заданий в присутствии DoS-атаки, позволяющий выбирать из потока заданий регулярные и игнорировать искусственные.

**Связь работы с научными программами, темами:** основные результаты диссертации были получены в рамках выполнения исследований при финансовой поддержке РФФИ (проекты 11-06-00165а, 12-07-31147 мол\_а), Германской службы академических обменов DAAD (долгосрочная научно-исследовательская стипендия с октября 2013 г. по июль 2014 г.) и Программы стратегического развития ПетрГУ.

**Апробация работы.** Основные результаты работы были представлены и обсуждены на следующих семинарах и конференциях:

1. Тринадцатый всероссийский симпозиум по прикладной и промышленной математике, 2–9 июня 2012 г., г. Петрозаводск;
2. Международный семинар «Networking Games and Management», 30 июня–2 июля 2012 г., г. Петрозаводск;
3. Двенадцатая всероссийская конференция «Высокопроизводительные параллельные вычисления на кластерных системах», 26–28 ноября 2012 г., г. Нижний Новгород;
4. XIV Российская конференция с участием иностранных ученых «Распределенные информационные и вычислительные ресурсы» 26–30 ноября 2012 г., г. Новосибирск;
5. Международный семинар «Russian-Chinese Seminar on Asymptotic Methods in Probability Theory and Mathematical Statistics», 10–14 июня 2013 г., г. Санкт-Петербург;
6. Первая российская конференция «Высокопроизводительные вычисления на базе VOINC: фундаментальные исследования и разработки», 9–13 сентября 2013 г., г. Петрозаводск;

7. VIII Международная научно-практическая конференция «Научно-образовательная информационная среда XXI века», 15–18 сентября 2014 г., г. Петрозаводск.

**Публикации.** Материалы диссертации опубликованы в 14 печатных работах, из них 5 статей в рецензируемых журналах, рекомендованных ВАК [1–5], одна статья в издании, индексируемом в библиографической базе данных Scopus [6], одна статья в рецензируемом журнале [7], 2 статьи в трудах конференций [8, 9] и 5 тезисов докладов [10–14]. Получено свидетельство о государственной регистрации программы для ЭВМ [15].

**Структура и объем диссертации.** Диссертационная работа состоит из введения, четырех глав, заключения и списка литературы. Общий объем диссертации составляет 112 страниц, включая 19 рисунков. Список литературы включает 109 наименований.

## Содержание работы

**Во введении** обоснована актуальность диссертационной работы, сформулирована цель и аргументирована научная новизна исследований, показана практическая значимость полученных результатов, представлены выносимые на защиту научные положения.

**Первая глава** имеет вводный характер. В ней приведен обзор основных принципов функционирования высокопроизводительных вычислительных систем и задач, возникающих при управлении вычислительным процессом в них, а также обзор источников литературы по темам исследований, которым посвящены следующие три главы.

**Во второй главе** приводится модификация алгоритма Backfill управления заданиями на вычислительном кластере и исследуются ее характеристики. С точки зрения владельца вычислительный кластер представляет собой множество тесно связанных вычислительных узлов, имеющих единый центр управления и планирования заданий, а с точки зрения пользователя — единый вычислительный ресурс. Наиболее распространенными базовыми алгоритмами управления заданиями на вычислительном кластере являются FCFS («первый пришедший обслуживается первым») и др. На их основе разрабатываются более сложные дисциплины обслуживания. Широко распространенной на практике и одной из наиболее эффективных по ряду критериев является комбинация базового алгоритма с процедурой Backfill. Исходный вариант алгоритма Backfill был впервые описан в рамках системы планирования заданий на суперкомпьютере IBM SP1 Аргонской Национальной лаборатории (США)<sup>1</sup>. В ходе работы алгоритма обрабатывается очередь заданий, ожидающих запуска на выполнение. Задания в очереди упорядочены по времени их регистрации в системе.

<sup>1</sup> Lifka D. The ANL/IBM SP scheduling system // Job Scheduling Strategies for Parallel Processing. — 1995. — Vol. 949. — P. 295–303.

При каждой регистрации и каждом завершении задания происходит выполнение следующих шагов алгоритма:

1. пока имеется необходимое количество свободных процессоров для запуска первого задания в очереди, удалять задание из очереди и запускать на выполнение;
2. если для запуска первого задания в очереди недостаточно свободных процессоров, вычислить момент времени, когда процессоров станет достаточно, и зарезервировать их для данного задания;
3. продолжать двигаться по очереди, запуская на выполнение задания, если они не нарушают резервирование первого задания.

Таким образом, задания, требующие небольшое количество процессоров или процессорного времени, могут запускаться на выполнение вне очереди при наличии необходимого количества доступных ресурсов (Рис. 1). В результате удается задействовать вычислительные ресурсы, простаивающие во время выполнения первоочередных заданий.

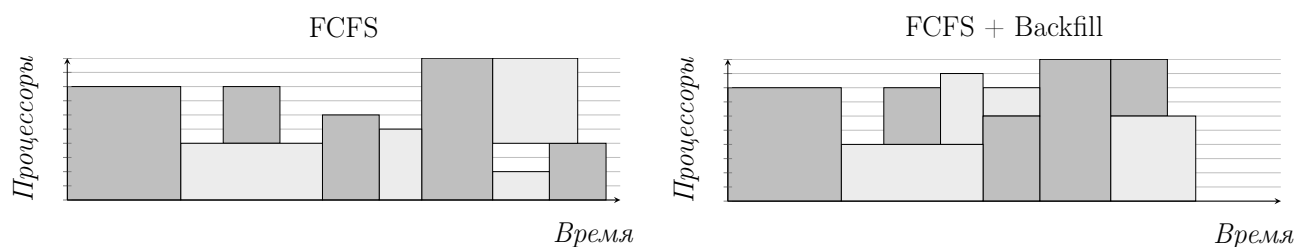


Рис. 1. Управление очередью заданий с использованием алгоритма Backfill.

Ожидаемое время выполнения задания в исходном алгоритме Backfill выражено точечной оценкой, что позволяет повысить производительность базового алгоритма управления заданиями при условии, что оценка достаточно точна<sup>2</sup>. В зависимости от политики регистрации заданий на кластере и поведения пользователей, оценки длительности могут отсутствовать или быть неточными, вследствие чего при применении алгоритма возникают ошибки — задержки запуска или принудительные прерывания заданий. В работе А. Ниссимова и Д. Фейтельсона<sup>3</sup> условие принятия решения о запуске задания, осуществляемого на шаге 2 и 3, заменяется условием «вероятность задержки момента запуска первого задания из очереди (вероятность ошибки) меньше заданного порога  $\tau$ ». При этом вместо точечной оценки времени выполнения заданий используется его эмпирическая плотность распределения.

<sup>2</sup> Tsafir D., Etsion Y., Feitelson D. G. Backfilling using system-generated predictions rather than user runtime estimates // IEEE Trans. Parallel and Distributed systems. — 2007. — Vol. 18, no. 6. — P. 789–803.

<sup>3</sup> Nissimov A., Feitelson D. G. Probabilistic Backfilling // Job Scheduling Strategies for Parallel Processing. — 2007. — Vol. 4942. — P. 102–115.

В диссертационной работе предлагается модификация алгоритма, в которой принятие решения осуществляется с использованием вероятности ошибки, выраженной в аналитическом виде. В **параграфе 2.2** представлена математическая модель задачи и найдены аналитические выражения для оценки вероятности ошибки в предположении, что распределение моментов завершения заданий имеет экспоненциальный вид, а количество освобождаемых в эти моменты процессоров подчиняется закону усеченного экспоненциального распределения или дискретного распределения Зипфа.

Рассматривается математическая модель вычислительного кластера с  $M$  процессорами. Пусть в настоящий момент времени свободно  $c_0$  процессоров ( $0 \leq c_0 \leq M$ ), и пусть в очереди находится  $N$  заданий  $J_1, J_2, \dots, J_N$ , упорядоченных по времени их регистрации в системе. Предположим, что для запуска первого задания в очереди  $J_1$  требуется  $c_q$  свободных процессоров и  $c_q > c_0$ , то есть свободных процессоров для запуска  $J_1$  в настоящий момент недостаточно. Согласно алгоритму Backfill, необходимо принять решение о запуске некоторого задания в очереди  $J_i, 1 < i \leq N$ , для которого имеется достаточно свободных процессоров:  $c_i \leq c_0$ . Пусть  $t_{\text{est}}$  — ожидаемое время выполнения задания  $J_i$ . Если в течение интервала времени длиной  $t_{\text{est}}$  освободится число процессоров, достаточное для запуска задания  $J_1$ , то немедленный запуск  $J_i$  может задержать запуск  $J_1$ . Решение запустить задание  $J_i$  принимается, если вероятность этого события меньше заданного порогового значения  $\tau$ .

Будем считать, что как только некоторое задание завершает работу, освобождаются занимаемые им ресурсы. Освобождение ресурсов будем называть скачком. Число свободных процессоров в дискретный момент времени  $t_i, i \geq 0$ , описывается случайным процессом

$$S(t_i) = S(t_{i-1}) + \xi_i,$$

где  $\xi_i$  — число процессоров, освобожденных в момент  $i$ -го скачка,  $S(0) = S_0 = 0$ . Тогда вероятность задержки запуска задания  $J_1$  в случае принятия решения о немедленном запуске задания  $J_i$  — это вероятность существования момента скачка, в который число свободных процессоров окажется достаточным для запуска первого задания из очереди  $J_1$ , но недостаточным для одновременной работы  $J_1$  и рассматриваемого задания  $J_i$ :

$$P = P(\exists t_i : c_q \leq S(t_i) < c, c = c_q + c_i). \quad (1)$$

Предположим, что величины  $\xi_1, \xi_2, \dots$  имеют усеченное экспоненциальное распределение с параметром  $\mu > 0$ , принимая значения от единицы до  $M$ , а моменты скачков (завершения заданий) распределены экспоненциально с пара-



метром  $\lambda > 0$ . Тогда аналитическое выражение вероятности (1) имеет вид

$$P = (1 - e^{-\lambda t_{\text{est}}}) (e^{-\mu c_q} - e^{-\mu c}) \sum_{k=1}^{\infty} \sum_{n=1}^k \frac{(\mu c_q)^{n-1}}{(n-1)!(e^{-\mu} - e^{-\mu M})^n}.$$

В предположении, что величины  $\xi_1, \xi_2, \dots$  имеют дискретное распределение Зипфа с параметром  $\mu > 0$ , принимая значения от единицы до  $M$ , аналитическое выражение вероятности (1) принимает вид

$$P = e^{-\lambda t_{\text{est}}} \sum_{k=1}^{\infty} \frac{(\lambda t_{\text{est}})^k}{k!} \sum_{n=1}^k \left( \left( \sum_{m=1}^M m^{-\mu} \right)^{-n} \times \right. \\ \left. \times \sum_{i_1=1}^{c_q-1} \sum_{i_2=1}^{c_q-i_1-1} \dots \sum_{i_{n-1}=1}^{c_q-i_1-\dots-i_{n-2}-1} \sum_{i_n=c_q-i_1-\dots-i_{n-1}}^{c-i_1-\dots-i_{n-1}-1} \left( \prod_{j=1}^n i_j \right)^{-\mu} \right).$$

Практическая эффективность предложенного алгоритма была оценена при помощи вычислительных экспериментов на модели потока заданий с различными видами и параметрами распределений моделируемых случайных величин (интервалов времени между приходами заданий, длительности их выполнения, количества требуемых процессоров). Результаты имитационного моделирования показали, что в целом предлагаемая модификация алгоритма Backfill позволяет сократить среднее время ожидания задания в очереди при количестве ошибок, не превышающем приемлемых на практике значений. При этом вид распределений моделируемых случайных величин даже при их равных средних значениях существенно влияет на эффективность работы алгоритма. На модели потока заданий при распределениях случайных величин, хорошо приближающих реальные характеристики вычислительного процесса<sup>4</sup>, алгоритм показывает наилучшую эффективность и по критерию снижения времени ожидания, и по количеству ошибок.

Выигрыш в среднем времени ожидания имеет прямую зависимость как от среднего количества требуемых процессоров, так и от среднего времени выполнения задания. С увеличением среднего количества процессоров или среднего времени выполнения (а следовательно, с увеличением доли относительно «больших» заданий) возрастает и количество ошибок. В Таблице 1 приведены результаты имитационного моделирования работы алгоритма на входных потоках заданий, воспроизводящих фрагменты истории функционирования следующих четырех вычислительных кластеров:

<sup>4</sup> Морозов Е. В., Румянцев А. С. Модели многосерверных систем для анализа вычислительного кластера // Труды Карельского научного центра РАН. — 2011. — No 5. — С. 75–85.

I. 80-процессорный вычислительный кластер Карельского научного центра РАН .....	8 282 заданий
II. 64-процессорный вычислительный кластер университета г. Амстердам, Нидерланды .....	66 429 заданий
III. 64-процессорный вычислительный кластер университета г. Утрехт, Нидерланды .....	33 695 заданий
IV. 100-процессорный вычислительный кластер шведского Королевского технологического института .....	28 490 заданий

Выигрыш в среднем времени ожидания выражен величиной

$$\Delta T_w = \frac{T_{wait}^{FCFS} - T_{wait}^{Backfill}}{T_{wait}^{FCFS}},$$

относительным изменением среднего времени ожидания задания в очереди (здесь  $T_{wait}^{FCFS}$  — среднее время ожидания задания в очереди при алгоритме обслуживания FCFS,  $T_{wait}^{Backfill}$  — при алгоритме Backfill). В качестве порогового значения было принято  $\tau = 0.05$ . Количество успешных применений процедуры Backfill и количество ошибок соответствуют полученным в данной главе выводам об их зависимости от параметров потока заданий.

Вычислительная система	$\Delta T_w$	Среднее время выполнения	Среднее количество процессоров	Доля применений Backfill	Доля ошибок
I	208.4	9.57	0.204	0.013	0.001
II	20.07	9.54	0.540	0.004	0.0004
III	45.94	3.55	0.525	0.017	0.001
IV	147.54	7.67	0.678	0.534	0.144

Таблица 1. Результаты работы алгоритма на множествах заданий, смоделированных по истории функционирования четырех вычислительных систем.

Полученные в главе результаты опубликованы в работе [3].

**В третьей главе** представлена теоретико-игровая модель процесса управления потоком заданий в системе распределенных вычислений типа Desktop Grid. Подобные системы применяются для решения крупных вычислительно-емких задач, которые могут выполняться на большом количестве слабосвязанных вычислительных узлов. Примером такой задачи является виртуальный скрининг лекарств — автоматизированный перебор большого количества моделей химических соединений и выбор среди них потенциально способных влиять на течение заболевания.

В системе Desktop Grid в качестве вычислительных узлов используются разнородные вычислительные ресурсы, принадлежащие организации или груп-

пе организаций: персональные компьютеры, веб-серверы, узлы вычислительного кластера и т.д. Вычислительные узлы (клиенты) связаны с управляющим узлом (сервером) сетью Интернет или локальной сетью передачи данных. Имеется иерархия вычислительных узлов с двумя уровнями: сервер создает рабочие задания, распределяет их между клиентами и обрабатывает полученные от клиентов результаты. Клиенты, получив задания, занимаются непосредственно их выполнением.

При управлении заданиями в системе Desktop Grid учитываются особенности принципов ее построения — в первую очередь, ненадежность<sup>5,6</sup> и разнородность<sup>7,8</sup> узлов, их большое количество при едином центре распределения работы<sup>9</sup>. Однако на выбор критериев эффективности алгоритма управления заданиями также может влиять специфика решаемой задачи.

В процессе виртуального скрининга лекарств задания выполняются на клиентах за относительно небольшое время — от нескольких секунд до нескольких минут. Быстро выполняя задания, отсылая их результаты серверу и запрашивая новые, клиенты генерируют интенсивный поток запросов, накладывающий высокие требования на программно-аппаратные характеристики сервера и сети передачи данных. Для снижения нагрузки на сервер в диссертационной работе предлагается группировать вычислительные задания в «пакеты». Интересы клиента и сервера являются не противоположными друг другу, поэтому процесс управления заданиями в Desktop Grid может быть смоделирован в виде двухуровневой иерархической игры  $m + 1$  игроков —  $m$  клиентов и одного сервера. Выбор стратегий игроков заключается в определении оптимального размера пакета и количества назначаемых заданий, и в равновесной ситуации достигается баланс между объемом вычислений на клиентах и нагрузкой на сервер.

Сервер  $M_0$  имеет  $N$  вычислительных заданий и может задействовать для их выполнения множество клиентов  $M_1, \dots, M_m$ . Сервер выбирает способ распределения заданий между клиентами  $u = (N_1, \dots, N_m)$ , где  $N_i$  — количество заданий, назначенное клиенту  $M_i$ . Множество возможных способов распределе-

---

<sup>5</sup> A Bi-objective Scheduling Algorithm for Desktop Grids with Uncertain Resource Availabilities / L.-C. Canon, A. Essafi, G. Mounié, D. Trystram // Proceedings of the 17th International Conference on Parallel Processing — Volume Part II. — Euro-Par'11. — Berlin, Heidelberg : Springer-Verlag, 2011. — P. 238–249. — URL: <http://dl.acm.org/citation.cfm?id=2033408.2033435>.

<sup>6</sup> Hwang H.-C., Lee K., Chang S. Y. The Effect of Machine Availability on the Worst-case Performance of LPT // Discrete Applied Mathematics. — 2005. — Vol. 148, no. 1. — P. 49–61. — URL: <http://dx.doi.org/10.1016/j.dam.2004.12.002>.

<sup>7</sup> Ibarra O. H., Kim C. E. Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors // J. ACM. — 1977. — Vol. 24, no. 2. — P. 280–289. — URL: <http://doi.acm.org/10.1145/322003.322011>.

<sup>8</sup> Fault-aware scheduling for bag-of-tasks applications on desktop grids / C. Anglano, J. Brevik, M. Canonico et al. // Proceedings of the 7th IEEE/ACM International Conference on Grid Computing. — 2006. — P. 56–63.

<sup>9</sup> Han J., Park D. Scheduling proxy: enabling adaptive-grained scheduling for global computing system // Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on. — 2004. — P. 415–420.

ния заданий  $u$  составляет множество стратегий сервера,

$$U = \{u = (u_1, \dots, u_m) : u_i \in \mathbb{Z}, 0 \leq u_i \leq N, \sum_{i=1}^m u_i = N\}$$

Зная назначенное сервером  $N_i$ , клиент  $M_i$  выбирает размер пакета  $n_i$ , минимизирующий его расходы. Множество возможных размеров пакета  $n_i$  для данного  $N_i$  составляет множество стратегий  $i$ -го клиента,

$$V_i(N_i) = \left\{ v_i \in \mathbb{Z} : \begin{cases} 1 \leq v_i \leq N_i & \text{при } N_i \geq 1, \\ v_i = 0 & \text{при } N_i = 0 \end{cases} \right\}$$

Таким образом, в сумме клиент должен выполнить  $\lceil \frac{N_i}{n_i} \rceil$  пакетов заданий. При этом необходимо учесть, что узлы Desktop Grid являются ненадежными ( $p_i \leq 1$  — вероятность успешного выполнения одного задания за одну попытку), и если хотя бы одно задание из пакета завершилось с ошибкой, то клиенту придется повторно запрашивать пакет у сервера и запускать его на выполнение. Функция выигрыша для клиента  $M_i$  выражает его расходы на вычисления:

$$C_i(n_i) = - \left\lceil \frac{N_i}{n_i} \right\rceil \left( G_1 \left( \frac{A_{\text{req}}(n_i) + A_{\text{err}}(n_i)}{p_i^n} + A_{\text{rep}}(n_i) - A_{\text{err}}(n_i) \right) + G_2 \frac{n_i}{p_i^n} \right). \quad (2)$$

Здесь  $A_{\text{req}}(n) \geq 0$  — стоимость запроса пакета заданий,  $A_{\text{rep}}(n) \geq 0$  — стоимость отправки серверу результата пакета заданий,  $A_{\text{err}} = a_{\text{err}} \geq 0$  — стоимость отправки серверу уведомления об ошибке;  $G_1$  — стоимость одного обращения к серверу,  $G_2$  — стоимость расчета одного задания;  $p_i^n$  — вероятность успешного выполнения пакета размером  $n$  заданий за одну попытку. Оптимальный размер пакета  $n_i^*$  минимизирует расходы клиента, то есть максимизирует его выигрыш:

$$C_i^* = C_i(N_i, n_i^*) = \max_{n_i \in V_i(N_i)} C_i(N_i, n_i), \quad 1 \leq i \leq m$$

Зная размеры пакетов, которые выберут клиенты, сервер стремится минимизировать свои расходы. Функция выигрыша для сервера выражает его расходы на подготовку пакетов, ожидание и обработку результатов:

$$\begin{aligned} C_0(N_1, \dots, N_m, n_1, \dots, n_m) &= \\ &= -T(N_1, \dots, N_m, n_1, \dots, n_m) - F(N_1, \dots, N_m, n_1, \dots, n_m) = \\ &= - \sum_{i=1}^m \left\lceil \frac{N_i}{n_i} \right\rceil (B(n_i) + S(n_i, p_i)) - F(N_1, \dots, N_m, n_1, \dots, n_m), \quad (3) \end{aligned}$$

где  $T(N_1, \dots, N_m, n_1, \dots, n_m)$  — функция стоимости обработки сервером всех заданий,  $B(n_i)$  — стоимость создания одного пакета из  $n_i$  заданий и обработки

результата его выполнения,  $S(n_i, p_i)$  — математическое ожидание количества обращений к серверу для расчета пакета,  $F(N_1, \dots, N_m, n_1, \dots, n_m)$  — функция штрафа за ожидание результатов.

Оптимальное распределение заданий  $u^* = (N_1^*, \dots, N_m^*)$  минимизирует расходы сервера, то есть максимизирует его выигрыш:

$$C_0^* = C_0(N_1^*, \dots, N_m^*, n_1^*, \dots, n_m^*) = \max_{u \in U} C_0(N_1, \dots, N_m, n_1^*, \dots, n_m^*).$$

Таким образом, имеется игра

$$\Gamma = \langle \{M_0, M_1, \dots, M_m\}, \{U, V_1, \dots, V_m\}, \{C_0, C_1, \dots, C_m\} \rangle.$$

В **параграфе 3.4** получены аналитические выражения функций выигрыша, оптимальных размеров пакетов для клиентов и оптимального распределения заданий в предположениях, что стоимость запроса клиентом пакета заданий, как и стоимость создания и обработки пакета на сервере, имеет линейную зависимость от размера пакета; стоимость отправки серверу результата прямо пропорциональна размеру пакета, а стоимостью отправки серверу сообщения об ошибке можно пренебречь. Приводятся численные примеры, иллюстрирующие, что группировка заданий в пакеты, снижая общий выигрыш сервера (за счет затрат на создание и обработку пакетов), позволяет, тем не менее, увеличить выигрыши клиентов и в 2.2–4.2 раза снизить общую нагрузку на сервер.

С октября 2013 г. по июль 2014 г. был реализован совместный научно-исследовательский проект между Институтом прикладных математических исследований КарНЦ РАН и Любекским Институтом экспериментальной дерматологии при университете г. Любек, Германия. В ходе выполнения проекта была реализована система Desktop Grid на основе открытого программного обеспечения с применением компьютеров, имеющихся в распоряжении сотрудников институтов, и проведен виртуальный скрининг лекарств. В **параграфе 3.5** сформулирована игровая модель системы.

Вычислительные узлы, вошедшие в состав Desktop Grid в ходе выполнения проекта, можно условно разделить на три типа: настольные компьютеры, узлы вычислительного кластера и веб-серверы. Узлы одного и того же типа обладают схожим временем доступности ресурсов и программно-аппаратными характеристиками, а следовательно, и надежностью. Надежность узлов различного типа была оценена в ходе выполнения проекта и составила  $p = 0.98$  для настольных компьютеров,  $p = 0.973$  для узлов вычислительного кластера и  $p = 0.993$  для веб-серверов.

Приближения функций стоимости запросов к серверу были получены из статистики обработки запросов к серверу при помощи полиномиальной и линейной регрессии. Штраф за ожидание сервером результатов выражал длительность расчетов, а общие расходы сервера — общее время выполнения проекта. Для нахождения ситуации равновесия по Нэшу была реализована программа

на языке C, осуществляющая поиск решения прямым перебором допустимых вариантов.

Результаты вычислительных экспериментов показали, что в равновесной ситуации при увеличении общего времени выполнения проекта на 2% достигается снижение общей нагрузки на сервер в 3.4 раза по сравнению с дисциплиной управления заданиями, принятой по умолчанию. С учетом многопроцессорности вычислительных узлов, в равновесной ситуации достигается не только снижение нагрузки на сервер в 3.5 раза, но и снижение общего времени выполнения проекта в 2.1 раза. Таким образом, предложенная модель позволяет решить поставленную задачу снижения нагрузки на сервер в системе Desktop Grid.

Полученные в главе результаты опубликованы в работе [6].

**В четвертой главе** рассматривается задача обнаружения момента DoS-атаки, или атаки типа «отказ в обслуживании», на вычислительную систему и последующего изменения протокола обработки входного потока заданий. Суть атаки заключается в том, что ко входному потоку «регулярных» заданий, генерируемому обычными пользователями, добавляется поток «искусственных» заданий, поступающих с высокой интенсивностью и, возможно, требующих большого количества ресурсов. При отсутствии атак ресурсы системы позволяют своевременно обслуживать все регулярные задания. Но при наличии атаки снижается эффективность алгоритма обслуживания, поскольку помимо регулярных заданий приходится обслуживать и искусственные, поступающие с высокой интенсивностью и/или требующие значительного количества ресурсов. Вычислительная система не способна обрабатывать все поступающие задания. В результате возрастает количество отказов в обслуживании, что и является целью DoS-атаки. Таким образом, возникает две задачи: зарегистрировать момент вторжения и изменить протокол обслуживания заданий так, чтобы продолжать обслуживать регулярные задания в присутствии искусственного потока.

Для обнаружения момента атаки в диссертационной работе используется метод кумулятивных сумм, который чувствителен к небольшим изменениям в наблюдаемом процессе, а значит, потенциально эффективен для быстрого выявления DoS-атак. Метод был впервые предложен в работе Е. С. Пейджа<sup>10</sup> для обнаружения момента разладки, то есть изменения среднего значения неотрицательных, независимых, одинаково распределенных случайных величин  $\{x_n\}$ ,  $n = 1, 2, \dots$ . Предполагается, что до момента разладки  $\theta \geq 0$  случайные величины подчиняются распределению  $F(x, \alpha_0)$ , а начиная с него — распределению того же вида, но с измененным параметром  $x_n \sim F(x, \alpha)$ , где  $\alpha \neq \alpha_0$ .

Если сигнал о наличии атаки подан в ее фактическое отсутствие, имеет место ложная тревога; характеристика процесса обнаружения атак ARL (Average Run Length) выражает среднее количество наблюдений до таковой. Характеристика алгоритма AD (Average Delay) выражает среднее количество наблюдений до подачи сигнала о фактически произошедшей атаке.

<sup>10</sup> Page E. S. Continuous Inspection Schemes // *Biometrika*. — 1954. — Vol. 41. — P. 100–114.

В ряде работ были получены приближенные и аналитические выражения среднего количества наблюдений до ложной тревоги и до обнаружения атаки: в частности, численное решение для случая нормального распределения<sup>11</sup>, численные и аналитические решения для случая экспоненциального распределения<sup>12,13</sup>, для распределений с «тяжелыми хвостами»<sup>14</sup>.

В диссертационной работе исследуется случай распределения Бернулли с параметром  $\alpha_0$ ,  $0 < \alpha_0 < 1$ . Подобная характеристика может описывать различные ситуации, возникающие на практике в вычислительной системе. Например, с вероятностью  $\alpha_0$  пришедшее задание сразу запускается на вычисление, а с вероятностью  $1 - \alpha_0$  становится в очередь; вследствие приходов искусственных заданий, занимающих значительный объем ресурсов системы, увеличится доля заданий, вынужденных становиться в очередь.

Для обнаружения момента разладки строится последовательность кумулятивных сумм

$$S_n = (S_{n-1} + q(x_n))^+, \quad (4)$$

где  $z^+ = \max(0, z)$ ,  $q(x) = \log \frac{dF(x, \alpha)}{dF(x, \alpha_0)}$ ,  $S_0 = s \geq 0$ .

Вывод о наличии разладки делается на шаге  $\tau_b$ , когда значение кумулятивной суммы впервые превышает заданное пороговое значение  $b$ .

При начальном условии  $S_0 = s$  выражение для характеристики ARL можно записать в следующем виде:

$$ARL = j_\infty(s) = \mathbb{E}_s\{\tau_b | \theta = \infty\}.$$

В работе Е. С. Пейджа показано, что функция  $j_\infty(s) < \infty$  является решением интегрального уравнения типа Фредгольма.

В случае распределения Бернулли выражение для вычисления кумулятивных сумм, приведенное к линейной форме, принимает вид

$$j(s) = \begin{cases} 1 + \alpha_0 j(s + \gamma + \beta) + (1 - \alpha_0) j(s + \beta)^+, & 0 \leq s < b \\ 0, & s \geq b \end{cases}$$

При  $[\gamma + \beta] = [-\beta]$  преобразованием переменных  $b^* = \left[ \frac{b}{-\beta} \right]$ ,  $-\beta = 1$ ,  $z = \frac{\gamma + \beta}{-\beta}$  получаем линейное разностное уравнение

$$j(s) = \begin{cases} 1 + \alpha_0 j(s + 1) + (1 - \alpha_0) j(s - 1)^+, & 0 \leq s < b^* \\ 0, & s \geq b^* \end{cases} \quad (5)$$

<sup>11</sup> Lucas J. M., Crosier R. B. Fast Initial Response for CUSUM Quality Control Schemes: Give Your CUSUM a Head Start // Technometrics. — 1982. — Vol. 24, No. 3. — P. 199–205.

<sup>12</sup> Gan F. F. Exact Run Length Distributions for One-Sided Exponential CUSUM Schemes // Statistica Sinica. — 1992. — Vol. 2, No. 1. — P. 297–312.

<sup>13</sup> Vardeman S., Ray D. Average Run Lengths for CUSUM Schemes When Observations Are Exponentially Distributed // Technometrics. — 1985. — Vol. 27, No. 2. — P. 145–150.

<sup>14</sup> Мазалов В. В., Журавлев Д. Н. О методе кумулятивных сумм в задаче обнаружения изменения трафика компьютерных сетей // Программирование. — 2002. — Т. 6. — С. 156–162.

Решение уравнения,  $j(s)$ , описывает среднее количество наблюдений до ложной тревоги в зависимости от параметра — порогового значения  $b$ , при условии  $S_0 = s$ . В **параграфе 4.2** получено аналитическое выражение решения уравнения (5), а также решения для случаев  $[\gamma + \beta] > [-\beta]$  и  $[\gamma + \beta] < [-\beta]$ . Приводится пример вычисления характеристик процесса обнаружения атаки с использованием полученных решений.

После обнаружения атаки целесообразно разделять входной поток заданий на регулярный и искусственный и отказывать в обслуживании искусственным заданиям. В **параграфе 4.3** предлагается протокол обслуживания на основе предположения, что атакующему неизвестен закон распределения регулярного входного потока и его параметры, что является отличием искусственных задач от регулярных.

Пусть задания, поступившие за некоторый промежуток времени, накапливаются в буфере и затем выборочно обслуживаются. Для обслуживания задания выбираются из буфера по одному в определенном порядке. Для обеспечения высокого качества обслуживания регулярных заданий после начала атаки необходимо выбирать из буфера регулярные задания, игнорируя искусственные. Для этого предлагается моделировать случайную величину, имеющую такое же распределение, как и регулярный входной поток заданий. После каждой реализации случайной величины из буфера выбирается задание, по рассматриваемой характеристике ближайшее к ней слева или справа.

Обозначим как  $z_n$  некоторую характеристику задания в наблюдаемом входном потоке,  $n = 1, 2, \dots$ . Пусть в регулярном потоке заданий  $z_n$  распределена по известному нам закону с медианой  $m$ . Введем случайную величину  $x_n$ , принимающую значение 0, если  $z_n < m$ , и 1, если  $z_n \geq m$ . Тогда в отсутствие атаки частота появления 0 и 1 примерно одинакова, и  $x_n$  имеет распределение Бернулли с параметром  $\alpha_0 = 0.5$ . После начала атаки параметр распределения  $\alpha$  меняется в большую или меньшую сторону относительно  $\alpha_0 = 0.5$ . Если после атаки характеристики заданий в среднем становятся больше, то и единиц в новой последовательности становится больше, что соответствует случаю  $\alpha > 0.5$ , и после обнаружения атаки следует обслуживать задания, лежащие на числовой оси в среднем «левее» генерируемой случайной величины. Верно и обратное.

При реализации данного алгоритма обслуживания возможно возникновение ошибок двух типов: обслуживание искусственного задания и отказ в обслуживании регулярного. На практике количество ошибок второго типа должно быть сведено к минимуму, поскольку в противном случае цель источника атакующего будет частично или полностью достигнута. В **параграфе 4.4** приводятся результаты вычислительных экспериментов, проведенных для исследования влияния значений параметров на эффективность алгоритма обслуживания. В качестве примера входного потока заданий в вычислительную систему был рассмотрен фрагмент истории функционирования реального вычислительного кластера, на котором был зафиксирован аномальный всплеск активности



$b$	ARL	AD	$b$	ARL	AD
187.68	126	6.22	344.08	3996	11.6
218.96	254	7.3	375.36	7932	12.7
250.24	508	8.4	406.64	15740	13.7
281.52	1012	9.4	437.92	31228	14.8
312.8	2012	10.5	469.2	61950	15.9

Таблица 2. Значения характеристик ARL и AD для различных пороговых значений  $b$ .

отдельного пользователя. При этом имело место существенное изменение характеристик входного потока заданий, которое может быть рассмотрено как DoS-атака. На основе аналитических выражений, полученных в параграфе 4.2, были вычислены значения характеристик ARL и AD для ряда пороговых значений  $b$ , которые приведены в Таблице 2.

Используя регистрационный файл вычислительного кластера, была построена последовательность кумулятивных сумм вида (4) для заданий, поступающих на рассматриваемом интервале времени. В качестве величины  $x_n$  было принято количество запрашиваемых процессоров. В предположении, что желаемое среднее количество наблюдений до подачи ложной тревоги  $ARL \geq 500$ , по Таблице 2 было выбрано значение порога  $b = 312.8$ . При имитационном моделировании процесса обнаружения атаки было зафиксировано 27 ложных тревог, среднее количество заданий между которыми составило 526.7, что согласуется с полученным аналитически результатом.

Работоспособность предложенного алгоритма обслуживания заданий была проверена при помощи имитационного моделирования. При фиксированном виде и значениях параметров распределений регулярного и искусственного потоков были смоделированы потоки из  $N = 11\,000$  заданий, среди которых было  $N_r = 1\,000$  регулярных и  $N_a = 10\,000$  искусственных. Для обслуживания из буфера выбиралось  $M$  заданий. На Рис. 2 приведена зависимость доли ошибок второго типа от обслуживаемой доли буфера в предположении, что интересующая нас характеристика регулярного потока заданий имеет нормальное распределение  $N(0.7, 0.1)$ , а характеристика потока заданий при наличии атаки —  $N(0.3, 0.1)$ . Результаты экспериментов показали, что при наличии точной или близкой к точной информации о виде и параметрах распределения характеристик заданий предложенный алгоритм обслуживания позволяет обрабатывать входящий поток заданий с низкой долей ошибок и низкой средней задержкой, приемлемыми на практике.

Результаты четвертой главы опубликованы в работе [7]. Программная реализация предложенного алгоритма обслуживания заданий прошла государственную регистрацию [15].

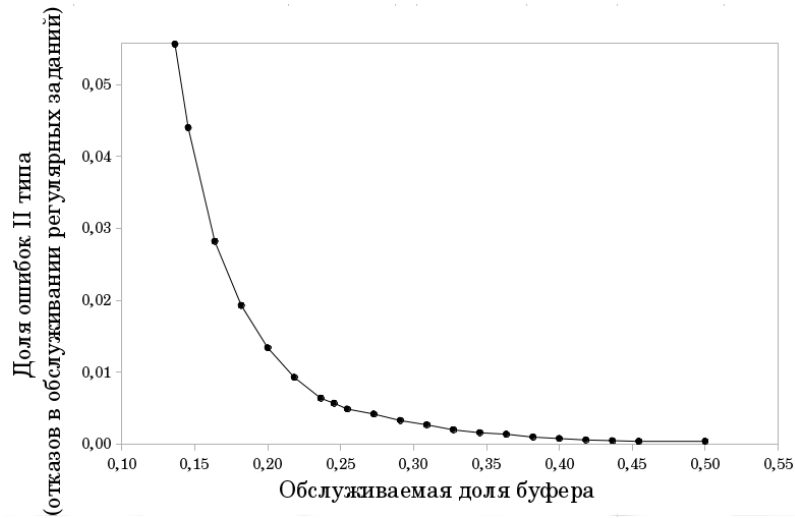


Рис. 2. Зависимость доли ошибок II типа от обслуживаемой доли буфера  $M/N$ .

## Заключение

В диссертации предлагается ряд математических моделей управления ресурсами высокопроизводительных вычислительных систем и защиты информации в них, исследуются и обсуждаются свойства данных моделей.

Предложена модификация алгоритма планирования заданий на вычислительном кластере с применением процедуры Backfill и выполнена оценка ее эффективности. В исходном варианте алгоритма решение о применении Backfill принимается на основе точечной оценки длительности выполнения задания. В описанной модификации алгоритма решение принимается на основе оценки вероятности задержки первоочередного задания. Процедура Backfill применяется, если вероятность задержки не превышает заданное пороговое значение. Приведены результаты имитационного моделирования вычислительного процесса с различными видами и параметрами распределений характеристик заданий, включая потоки заданий, воспроизводящие фрагменты истории функционирования реальных вычислительных кластеров. На смоделированных потоках заданий проиллюстрирована работоспособность алгоритма и приведены выводы о ее применимости на практике.

Также в диссертации приводится описание задачи виртуального скрининга и системы Desktop Grid, реализованной для ее решения в рамках совместного научно-исследовательского проекта между Институтом прикладных математических исследований КарНЦ РАН и Любекским Институтом экспериментальной дерматологии при университете г. Любек, Германия. Предлагается теоретико-игровая модель процесса управления потоком заданий в централизованной системе распределенных вычислений типа Desktop Grid. В качестве критерия оптимизации принимается минимизация общей нагрузки на сервер. Используемые в ходе вычислительных экспериментов оценки вида функций и значений параметров модели получены по результатам проведения виртуального скрининга. Приводятся примеры нахождения ситуации равновесия аналитически,

а также результаты вычислительных экспериментов, иллюстрирующие работоспособность модели.

Результатом диссертационной работы являются также аналитические выражения для характеристик процесса обнаружения атаки типа «отказ в обслуживании» на вычислительную систему — в частности, среднего количества наблюдений до ложной тревоги и средней задержки до обнаружения момента атаки. Представлены результаты имитационного моделирования входного потока заданий в вычислительной системе. Для противодействия атаке предложен протокол обслуживания, при котором из входного потока выделяются задания, ассоциированные с нормальным режимом функционирования системы.

## Список публикаций

1. Чернов И. А., Ивашко Е. Е., Никитина Н. Н., Габис И. Е. Численная идентификация модели дегидрирования в грид-системе на базе VOINC // Компьютерные исследования и моделирование. 2013. Т. 5, № 1. С. 37–45.
2. Ивашко Е. Е., Никитина Н. Н. Использование VOINC-грид в вычислительноемких научных исследованиях // Вестник НГУ. Серия: Информационные технологии. 2013. Т. 11. С. 53–57.
3. В. В. Мазалов, Н. Н. Никитина. Оценка характеристик алгоритма Backfill при управлении потоком задач на вычислительном кластере // Вычислительные технологии. 2012. Т. 17, № 5. С. 71–79.
4. Ивашко Е. Е., Никитина Н. Н. Организация квантовохимических расчетов с использованием пакета Firefly в гетерогенной грид-системе на базе VOINC // Вычислительные методы и программирование. 2012. Т. 13. С. 8–12.
5. Вдовицын В. Т., Сорокин А. Д., Ивашко Е. Е. и др. Основные результаты деятельности ЦКП КарНЦ РАН «Центр высокопроизводительной обработки данных» // Труды КарНЦ РАН. 2011. Т. 2, № 5. С. 125–129.
6. Mazalov V. V., Nikitina N. N., Ivashko E. E. Hierarchical Two-Level Game Model for Tasks Scheduling in a Desktop Grid // Applied Problems in Theory of Probabilities and Mathematical Statistics Related to Modeling of Information Systems. Leonia, NJ, USA: IEEE, 2014. P. 641–645.
7. Никитина Н. Н., Чернов И. А. Разрешимость системы разностных уравнений для динамики кумулятивной суммы // Проблемы анализа. 2013. Т. 2, № 20. С. 68–81.

8. Ивашко Е. Е., Никитина Н. Н. Организация квантовохимических расчетов с использованием пакета Firefly в гетерогенной Грид на базе BOINC // Научный сервис в сети Интернет: эксафлопсное будущее. Труды Международной суперкомпьютерной конференции (Абрау-Дюрсо, 19-24 сентября 2011 г.). 2011. С. 178–181.
9. Ивашко Е. Е., Никитина Н. Н. Использование BOINC-грид в вычислительноемких научных исследованиях. 2012. Режим доступа: <http://conf.nsc.ru/dicr2012/ru/reportview/140625>.
10. Никитина Н. Н. Оценка характеристик алгоритма Backfill при управлении потоком задач на вычислительном кластере // Обозрение прикладной и промышленной математики. Т. 19. Москва: «ОПиПМ», 2012. С. 461.
11. Nikitina N. Detection of a change point in Bernoulli distribution and applications in computer networks // Networking Games and Management. Extended abstracts of the International workshop. Petrozavodsk: Karelian Research Centre of the RAS, 2012. P. 44.
12. Mazalov V., Nikitina N. Detection of a change point in Bernoulli distribution and applications in computer networks // Russian-Chinese Seminar on Asymptotic Methods in Probability Theory and Mathematical Statistics: Programme and Abstracts. St. Petersburg: St. Petersburg State University, 2013. P. 26.
13. Румянцев А. С., Никитина Н. Н. Задача оптимизации времени выполнения приложения в Desktop Grid // Высокопроизводительные параллельные вычисления на кластерных системах. Тезисы докладов XII Всероссийской конференции. Нижний Новгород: Издательство Нижегородского госуниверситета, 2012. С. 289–291.
14. Чернов И. А., Никитина Н. Н. Математические модели управления очередями заданий в BOINC. 2013. Режим доступа: <http://boincfast.ru/papers/nikitina.pdf>.
15. Мазалов В. В., Никитина Н. Н. Программа изменения дисциплины обслуживания при обнаружении DoS-атаки. 2013. Свидетельство о государственной регистрации программы для ЭВМ No.2013618298 от 05.09.2013.

*Научное издание*

Подписано в печать 14.10.2014. Формат 60 × 90 1/16. Тираж 100 экз. Заказ 256.

Карельский научный центр РАН  
185003, Петрозаводск, пр. А. Невского, 50  
Редакционно-издательский отдел