Институт прикладных математических исследований КарНЦ РАН

Математические модели управления заданиями и защиты информации в высокопроизводительных вычислительных системах

Диссертация
на соискание ученой степени кандидата технических наук

Специальность 05.13.18 — Математическое моделирование, численные методы и комплексы программ

Никитина Наталия Николаевна

Научный руководитель:
д. ф.-м. н., профессор
Мазалов Владимир Викторович

Санкт-Петербургский институт информатики и автоматизации РАН, 23.10.2014 г.

Цель и задачи

Цель диссертационной работы заключается в построении и исследовании свойств математических моделей управления ресурсами высокопроизводительных вычислительных систем и защиты информации в них с применением методов теории вероятностей, теории оптимизации и теории некооперативных игр.

В работе исследуются следующие основные **задачи**:

1. Задача повышения эффективности алгоритма Backfill для управления заданиями на вычислительном кластере;
2. Задача минимизации нагрузки на сервер в централизованной системе распределенных вычислений;
3. Задача определения момента начала DoS-атаки на вычислительную систему с последующим изменением протокола обслуживания заданий.

Положения, выносимые на защиту

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 присутствие DoS-атаки, позволяющий выбирать из потока заданий регулярные и игнорировать искусственные.

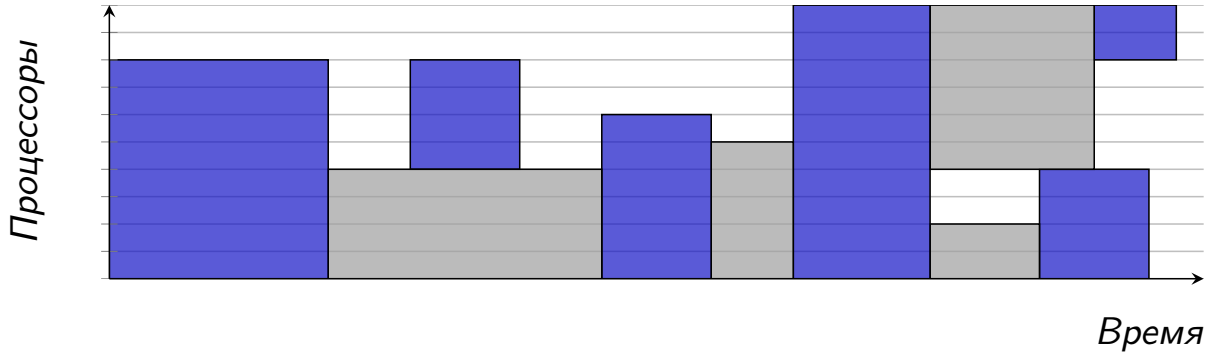
Глава 1. Обзор методов управления заданиями в высокопроизводительных вычислительных системах

Высокопроизводительные вычислительные системы используются для решения задач, требующих выполнения значительных объемов вычислений, а также обработки, хранения, передачи и визуализации больших объемов данных.

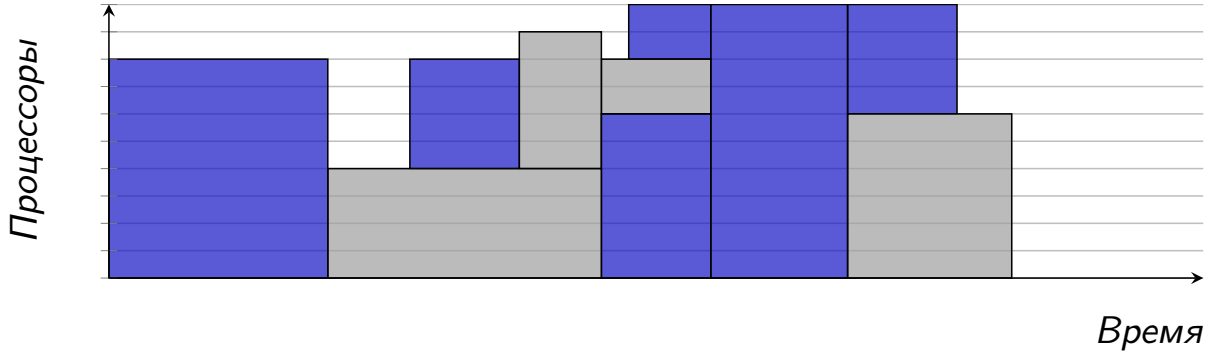
- **Суперкомпьютеры**, архитектура и ПО которых разработаны для решения частного класса научных задач;
- **Вычислительные кластеры**, представляющие собой множество стандартных многопроцессорных вычислительных узлов, объединенных высокоскоростными коммуникационными каналами в единую программно-аппаратную систему;
- **Системы распределенных вычислений**, объединяющие территориально распределенные вычислительные ресурсы, в том числе неспециализированные.

Глава 2. Оценка характеристик алгоритма Backfill при управлении потоком заданий на вычислительном кластере

Управление очередью заданий при помощи алгоритма FCFS (First Come First Served)



Управление очередью заданий при помощи алгоритма FCFS + Backfill



Параграф 2.1. Постановка задачи

Исходный алгоритм Backfill:

1. пока имеется достаточно 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 каждое задание, для которого имеется достаточно свободных процессоров (**процедура Backfill**), при условии, что вероятность задержки момента запуска первого задания из очереди меньше заданного порога τ.

Параграф 2.1. Постановка задачи

M — количество процессоров на вычислительном кластере.
c_0 — количество свободных процессоров ($0 \leq c_0 \leq M$).
$J_1, \ldots, J_N$ — задания в очереди.
c_q — количество процессоров, требуемых заданием J_1 ($c_0 < c_q \leq M$).
J_i — задание, для которого возможно применение Backfill ($1 < i \leq N$).
c_i — количество процессоров, требуемых заданием J_i ($c_i \leq c_0$).
t_{est} — ожидаемое время выполнения задания J_i.
τ — пороговое значение для вероятности ошибки.
P — вероятность ошибки, то есть задержки запуска J_1.

Решение о применении процедуры Backfill

К заданию J_i применяется процедура Backfill, если $P < \tau$.

Параграф 2.2. Аналитическое выражение вероятности ошибки

$S(t_i) = S(t_{i-1}) + \xi_i$ — количество свободных процессоров в момент времени t_i.
$t_i, i \geqslant 0$ — моменты освобождения ресурсов (скачков).
ξ_i — число процессоров, освобожденных в момент i-го скачка.
$S(0) = S_0 = 0$.

$$P = P(\exists t_i : c_q \leq S(t_i) < c, c = c_q + c_i) \tag{1}$$

Усеченное экспоненциальное распределение

$$f_{\xi_1}(x) = \frac{\mu e^{-\mu x}}{e^{-\mu L} - e^{-\mu H}}, \text{ где } L = 1, H = M$$

Аналитическое выражение вероятности ошибки

$$P = \left(1 - e^{-\lambda t_e}\right)\left(e^{-\mu c_q} - e^{-\mu c}\right) \sum_{n=1}^{\infty} \sum_{k=n}^{\infty} \frac{(\mu c_q)^{n-1}}{(n-1)!(e^{-\mu} - e^{-\mu M})^n}$$

Параграф 2.2. Аналитическое выражение вероятности ошибки

Распределение Зипфа

$$P(\xi_1 = i) = \frac{i^{-\mu}}{\sum_{k=1}^{M} k^{-\mu}}$$

Аналитическое выражение вероятности ошибки

$$P = \sum_{k=1}^{\infty} \frac{(\lambda t_e)^k}{k!} e^{-\lambda t_e} \sum_{n=1}^{k} \left(\sigma^n \sum_{i_1=1}^{c_q-1} \sum_{i_2=1}^{c_q-i_1-1} \cdots \sum_{i_n=c_q-i_1-\ldots-i_{n-1}}^{c-i_1-\ldots-i_{n-1}-1} \left(\prod_{j=1}^{n} i_j \right)^{-\mu} \right)$$

Параграф 2.3. Вычислительные эксперименты

Имитационная модель управления потоком заданий на вычислительном кластере

$M = 64$ — общее количество процессоров.
9.53 — среднее запрашиваемое количество процессоров.
105.93 мин. — средний интервал времени между приходами заданий.
208.33 мин. — среднее время выполнения заданий.

$\tau = 0.05$ — пороговое значение для вероятности ошибки.

T_{wait}^{FCFS} — среднее время ожидания задания в очереди при дисциплине обслуживания FCFS.

$T_{wait}^{Backfill}$ — среднее время ожидания задания в очереди с применением процедуры Backfill.

$\Delta T_{wait} = \dfrac{T_{wait}^{FCFS} - T_{wait}^{Backfill}}{T_{wait}^{FCFS}}$ — относительное изменение среднего времени ожидания задания в очереди.

Параграф 2.3. Вычислительные эксперименты

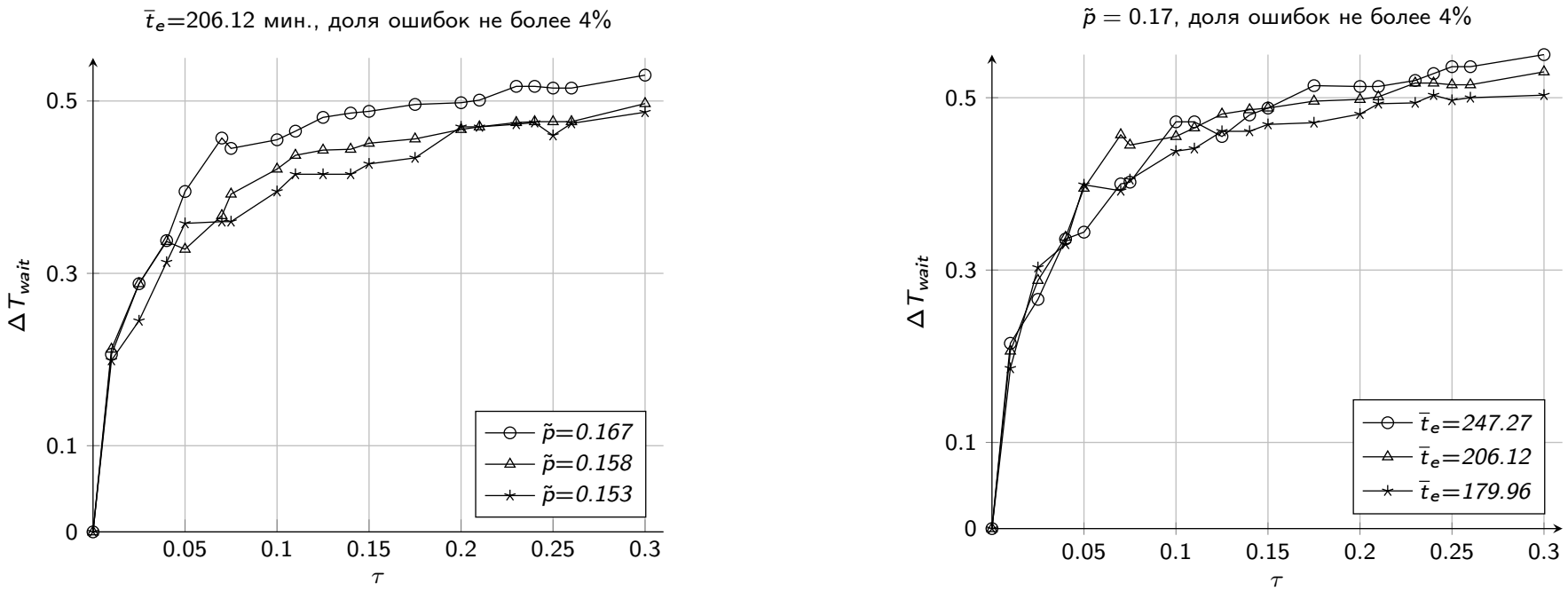


Рис. 1: Изменение среднего времени ожидания в зависимости от порогового значения τ.

Параграф 2.3. Вычислительные эксперименты

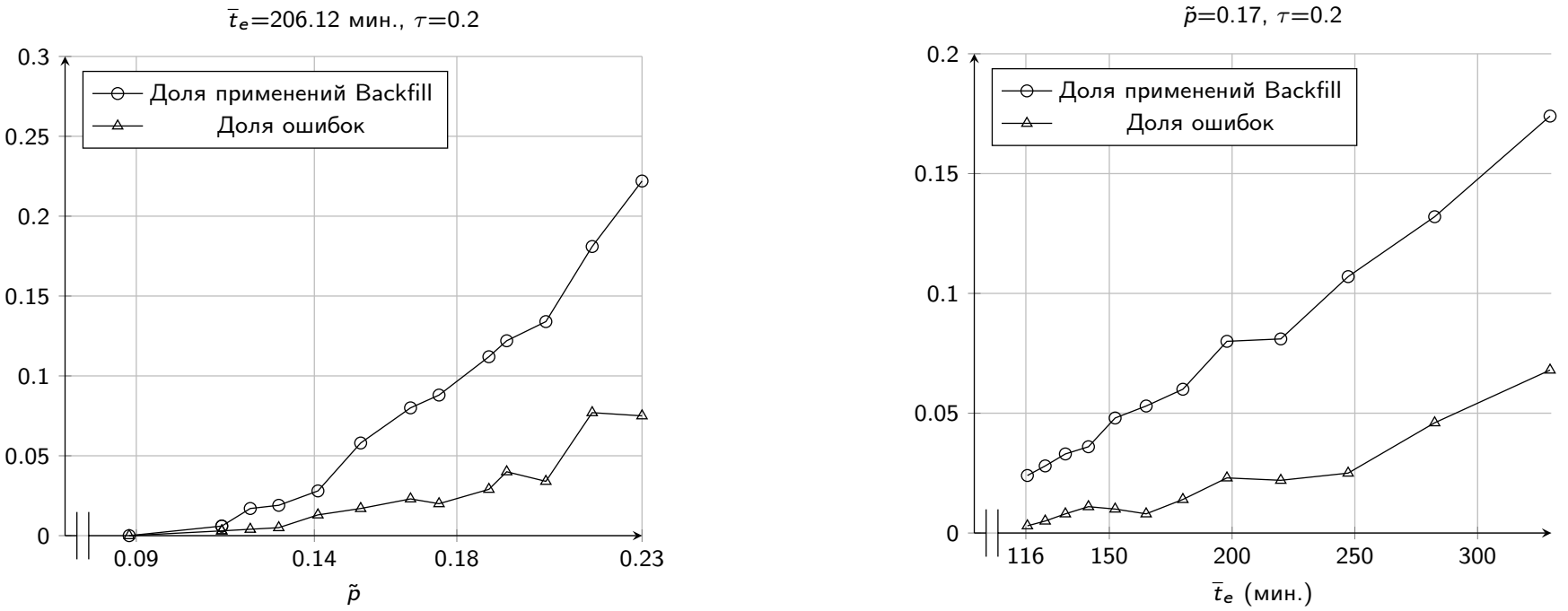


Рис. 2: Доля применений Backfill и доля ошибок в зависимости от среднего числа процессоров и среднего времени выполнения одного задания.

Параграф 2.3. Вычислительные эксперименты

Модель управления потоком заданий, восстановленным из истории функционирования вычислительной системы

KRC — 80-процессорный вычислительный кластер КарНЦ РАН: среднее время выполнения задания — 208.4 мин., максимальное — 3 суток, среднее количество процессоров — 9.57.
DAS2 — 64-процессорный вычислительный кластер университета г. Амстердам: среднее время выполнения задания — 20.07 мин., максимальное — 37.5 суток, среднее количество процессоров — 9.54.
DAS4 — 64-процессорный вычислительный кластер университета г. Утрехт: среднее время выполнения задания — 45.94 мин., максимальное — 25.1 суток, среднее количество процессоров — 3.55.
KTH — 100-процессорный вычислительный кластер Шведского королевского техн. института: среднее время выполнения задания — 147.54 мин., максимальное — 2.6 суток, среднее количество процессоров — 7.67.

Система	ΔT_w	Доля применений Backfill	Доля ошибок
KRC	0.204	0.013	0.001
DAS2	0.540	0.004	0.0004
DAS4	0.525	0.017	0.001
KTH	0.678	0.534	0.144

Параграф 2.3. Вычислительные эксперименты

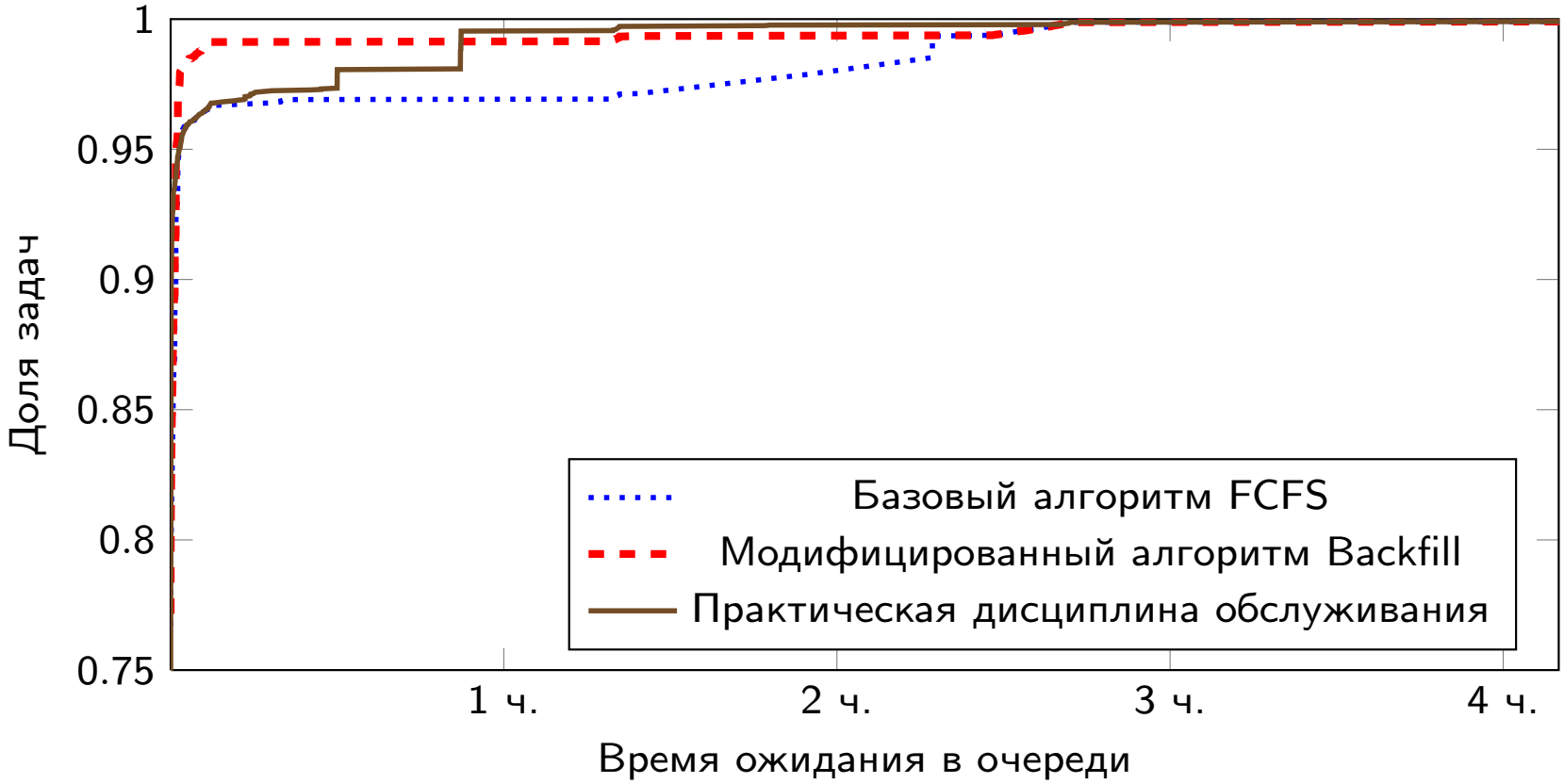


Рис. 3: Эмпирические функции распределения времени ожидания для различных алгоритмов планирования в системе DAS2.

Глава 3. Теоретико-игровая модель управления заданиями в Desktop Grid

Desktop Grid

Desktop Grid — вычислительная система типа «сервер-клиент», в которой клиентами являются разнородные территориально распределенные компьютеры, способные выполнять большое количество независимых заданий.

Виртуальный скрининг лекарств

Виртуальный скрининг лекарств — процесс автоматизированного перебора большого количества независимых моделей химических соединений и выбора среди них потенциально способных влиять на течение заболевания.

Проблема высокой нагрузки на сервер в Desktop Grid

Большое количество независимых, быстро выполняющихся клиентами вычислительных заданий приводит к высокой нагрузке на сервер.

Глава 3. Теоретико-игровая модель управления заданиями в Desktop Grid

M_0 — сервер, $M_1, \ldots, M_m$ — клиенты Desktop Grid. Сервер выбирает способ распределения N заданий между m клиентами. Множество стратегий сервера:

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

Клиент M_i выбирает размер пакета n_i. Множество стратегий клиента:

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

Функция выигрыша для клиента M_i:

$$C_i(N_i, n_i) = -K(n_i) \left\lceil \frac{N_i}{n_i} \right\rceil$$

$K(n_i) = G_1 S(n_i) + G_2 Z(n_i)$ — расходы клиента на расчет одного пакета. Оптимальный размер пакета n_i^* минимизирует расходы клиента:

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

Глава 3. Теоретико-игровая модель управления заданиями в Desktop Grid

Каждый расчет пакета заданий требует одного запроса к серверу и одного отчета с результатом. В случае ошибки вместо результата посылается отчет о ней, и требуется пересчет всего пакета, то есть новый запрос к серверу. Расходы на коммуникацию узла с сервером определяются стоимостью выполнения операции сервером:

- $A_{\mathrm{req}}(n) \geq 0$ — стоимость запроса пакета заданий,
- $a_{\mathrm{req}} = A_{\mathrm{req}}(1)$ — стоимость запроса одного задания,
- $A_{\mathrm{rep}}(n) \geq 0$ — стоимость отправки серверу результата пакета заданий,
- $a_{\mathrm{rep}} = A_{\mathrm{rep}}(1)$ — стоимость отправки серверу результата одного задания,
- $A_{\mathrm{err}} = a_{\mathrm{err}} \geq 0$ — стоимость отправки серверу уведомления об ошибке.

Глава 3. Теоретико-игровая модель управления заданиями в Desktop Grid

Узлы Desktop Grid не являются абсолютно надежными:
p — вероятность успешного выполнения одного задания за одну попытку,
p^n — вероятность успешного выполнения пакета из n заданий за одну попытку.
$G_1 S$ — плата за коммуникацию с сервером; G_1 — стоимость одного обращения к серверу, S — МО количества обращений к серверу для расчета одного пакета.
$G_2 Z$ — плата непосредственно за расчеты; G_2 — стоимость расчета одного задания, $Z = n\sum_{i=0}^{\infty}(1-\bar{p})^i = \frac{n}{\bar{p}}$ — МО количества посчитанных заданий с учетом перезапусков в результате ошибок.
МО количества обращений к серверу для расчета пакета:

$$S(n_i, p) = \frac{A_{\text{req}}(n_i) + A_{\text{err}}(n_i)}{\bar{p}} + A_{\text{rep}}(n_i) - A_{\text{err}}(n_i)$$

Функция выигрыша клиента M_i:

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

Глава 3. Теоретико-игровая модель управления заданиями в Desktop Grid

Функция выигрыша для сервера выражает его расходы на подготовку пакетов, ожидание и обработку результатов:

$$C_0 = C_0(N_1, \ldots, N_m, n_1, \ldots, n_m)$$

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

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

Глава 3. Теоретико-игровая модель управления заданиями в Desktop Grid

Функция выигрыша сервера:

$$C_0(N_1, \ldots, N_m, n_1, \ldots, n_m) = -T(N_1, \ldots, N_m, n_1, \ldots, n_m) -$$

$$- F(N_1, \ldots, N_m, n_1, \ldots, n_m) = -\sum_{i=1}^{m} \left\lceil \frac{N_i}{n_i} \right\rceil (B(n_i) + S(n_i, p_i)) -$$

$$- F(N_1, \ldots, N_m, n_1, \ldots, n_m),$$

- $T(N_1, \ldots, N_m, n_1, \ldots, n_m)$ — функция стоимости обработки сервером всех заданий.
- $B(n_i)$ — стоимость создания одного пакета из n_i заданий и обработки результата его выполнения.
- $S(n_i, p_i)$ — МО количества обращений к серверу для расчета пакета.
- $F(N_1, \ldots, N_m, n_1, \ldots, n_m)$ — функция штрафа за ожидание результатов.

Параграф 3.4. Аналитическое выражение функций выигрыша

Пусть в состав Desktop Grid входят два клиента I и II с надежностями $0 < p < 1$ и $0 < q < 1$. Сервер выбирает способ разделения N заданий на две части: N_1 заданий будет отправлено первому клиенту, $N - N_1$ — второму.

$A_{\text{req}}(n) = kn + b$, $b \neq 0$, $k > 0$ — стоимость запроса пакета заданий.

$A_{\text{rep}}(n) = \alpha n$, $\alpha > 0$ — стоимость ответа.

$A_{\text{err}}(n) \equiv 0$ — стоимость отправки серверу сообщения об ошибке.

$B(n) = xn + y$, $x > 0, y \neq 0$ — стоимость создания и обработки пакета.

$F(N_1, \ldots, N_m) = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{N_i}{p_i^{n_i^*}} \right)^2$ — штраф за ожидание результатов.

Параграф 3.4. Аналитическое выражение функций выигрыша

Пример

$p = 0.999$, $q = 0.97$, $k = 0.0025$, $b = 0.003$, $\alpha = 0.005$, $x = 0.025$, $y = 0.5$, $G_1 = 1$, $G_2 = 0.01$, $N = 1\,000\,000$.
$n_1^* = 15$ — оптимальный размер пакета для клиента I.
$n_2^* = 3$ — оптимальный размер пакета для клиента II.
$N_1^* = 538\,111$ — оптимальное количество заданий для сервера.
$C_I(n_1^*) = -9\,627.9$ — выигрыш клиента I.
$C_{II}(n_2^*) = -9\,013.7$ — выигрыш клиента II.
$C_0 = C_0(N_1^*) = -2.77254 \cdot 10^{11}$ — выигрыш сервера.
$T = T(N_1^*, n_1^*) = 128\,165$ — стоимость обработки сервером всех заданий.

Рассмотрим случай без группировки заданий в пакеты, $n_1 \equiv n_2 \equiv 1$.
$\tilde{N_1}^* = 514\,725$.
$\tilde{C}_I(1) = -10\,559.8$.
$\tilde{C}_{II}(1) = -10\,180.8$.
$\tilde{C}_0 = -2.57879 \cdot 10^{11} \approx 0.93 \times C_0$ — выигрыш сервера в отсутствие затрат на обработку пакетов.
$\tilde{T} = 535\,585 \approx 4.2 \times T$ — стоимость обработки всех заданий.

Параграф 3.5. Вычислительные эксперименты

Desktop Grid для проведения виртуального скрининга

В ходе проведения совместного научно-исследовательского проекта между ИПМИ КарНЦ РАН и Любекским Институтом экспериментальной дерматологии при университете г. Любек (Германия) была реализована Desktop Grid и проведен виртуальный скрининг с использованием открытого программного обеспечения.

- Настольные ПК ($p \approx 0.98$);
- Узлы вычислительного кластера ($p \approx 0.973$);
- Вычислительные серверы ($p \approx 0.993$).

Модель системы с тремя однопроцессорными клиентами

$n_1^* = 3$, $n_2^* = 3$, $n_3^* = 5$ — оптимальные размеры пакетов для клиентов.
$N_1^* = 3\,328$, $N_2^* = 3\,258$, $N_3^* = 3\,414$ — оптимальное распределение заданий.
$C_0 = C_0^* = -226\,486.66$ — выигрыш сервера.
$T = 1\,771.47$ — стоимость обработки сервером всех заданий.

Рассмотрим случай без группировки заданий в пакеты, $n_1 \equiv n_2 \equiv n_3 \equiv 1$.
$\tilde{C}_0 \approx 0.98 \times C_0$ — выигрыш сервера в отсутствие затрат на обработку пакетов.
$\tilde{T} \approx 3.4 \times T$ — стоимость обработки всех заданий.

Параграф 3.5. Вычислительные эксперименты

Модель системы с $m = m_1 + m_2 + m_3$ многопроцессорными клиентами

$m_1 = 25, m_2 = 10, m_3 = 10$ — количество клиентов I, II и III типов.
$Q_1 = 4, Q_2 = 8, Q_3 = 12$ — среднее количество ЦП у клиентов I, II и III типов.
$n_1^* = 3,\ n_2^* = 3,\ n_3^* = 5$ — оптимальные размеры пакетов для клиентов.
$N_1^* = 133,\ N_2^* = 259,\ N_3^* = 409$ — оптимальное распределение заданий.
$C_0 = -3\,963.10$ — выигрыш сервера.

Рассмотрим случай без группировки заданий в пакеты, $n_1 \equiv n_2 \equiv n_3 \equiv 1$.
$\tilde{C}_0 \approx 2.1 \times C_0$ — выигрыш сервера в отсутствие затрат на обработку пакетов.
$\tilde{T} \approx 3.5 \times T$ — стоимость обработки всех заданий.

Глава 4. Метод кумулятивных сумм для обнаружения вторжений в вычислительную систему и борьбы с ними

Метод кумулятивных сумм для обнаружения разладки

Разладка — это изменение в момент времени θ параметра распределения неотрицательных, независимых, одинаково распределенных случайных величин $\{x_n\}_{n=1,2,\ldots}$ с α_0 на $\alpha \neq \alpha_0$.

$$S_0 = s,\ S_n = (S_{n-1} + \log \frac{dF(x_n,\alpha)}{dF(x_n,\alpha_0)})^+,\ n \geq 1 \text{ — кумулятивные суммы.}$$

На шаге наблюдения τ_b делается вывод о наличии разладки, если значение КС впервые превысило заданное пороговое значение b:

$$\tau_b = \inf\{n > 0 : S_n \geq b\}$$

$ARL = j_\infty(s) = \mathbb{E}_s\{\tau_b | \theta = \infty\}$ — среднее количество наблюдений до сигнала об обнаружении разладки при условии, что по факту разладка не произойдет.
$AD = j_0(s) = \mathbb{E}_s\{\tau_b | \theta = 0\}$ — среднее количество наблюдений до сигнала при условии, что по факту разладка произошла в нулевой момент времени.
Е. Пейдж (1954): $j_\infty(s) = 1 + \mathbb{E}_s\{\mathbb{I}(0 < S_1 < b) j_\infty(S_1)\} + \mathbb{P}_s\{S_1 = 0\} j_\infty(0),\ s < b.$

Параграф 4.2. Аналитическое выражение характеристик процесса

Пусть до разладки x_n $(n = 1, 2, \ldots)$ имеют распределение Бернулли с параметром α_0, $0 < \alpha_0 < 1$, после разладки — распределение Бернулли с параметром $\alpha \neq \alpha_0$ (далее предполагаем $\alpha > \alpha_0$).

$S_n = \left(S_{n-1} + \left(\ln \frac{\alpha}{\alpha_0} - \ln \frac{1-\alpha}{1-\alpha_0}\right) x_n + \ln \frac{1-\alpha}{1-\alpha_0}\right)^+$ — выражение для КС.

$$j(s) = \begin{cases} 1 + \alpha_0 j(s + \gamma + \beta) + (1 - \alpha_0) j(s + \beta)^+, & 0 \leq s < b \\ 0, \ s \geq b \end{cases} \tag{2}$$

Здесь $\gamma = \ln \frac{\alpha}{\alpha_0} - \ln \frac{1-\alpha}{1-\alpha_0}$, $\beta = \ln \frac{1-\alpha}{1-\alpha_0}$.

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

$$j(s) = \begin{cases} 1 + \alpha_0 j(s + 1) + (1 - \alpha_0) j(s - 1)^+, 0 \leq s < b^* \\ 0, \ s \geq b^* \end{cases} \tag{3}$$

Решение уравнения, $j(s)$, описывает среднее количество наблюдений до ложной тревоги в зависимости от параметра — порогового значения b^*, при условии $S_0 = s$. В диссертации получено аналитическое выражение решения уравнения (3), а также решения для случаев $[\gamma + \beta] > [-\beta]$ и $[\gamma + \beta] < [-\beta]$.

Параграф 4.2. Аналитическое выражение характеристик процесса

При $S_0 = s = 0$ функция $j(0, b)$ возрастает, поэтому пороговое значение

$$\mathbf{b} = min\{b : j_\infty(b) \geq a\}$$

минимизирует среднюю задержку AD при условии, что среднее количество наблюдений до ложной тревоги $ARL \geq a$.

Пример

Пусть $\alpha_0 = 0.1$, $\alpha = 0.6$. Тогда для целых значений параметра b характеристика ARL принимает следующие значения:

b	3	4	5	6	7	8	9
ARL	62.62	102.56	377.13	763.04	2310.51	5211.67	14459.4

Параграф 4.3. Изменение протокола обслуживания

Пусть z_n $(n = 1, 2, \ldots)$ — некоторая характеристика задания в наблюдаемом входном потоке, и в регулярном потоке заданий z_n распределена по известному нам закону с медианой m.
Пусть x_n — с. в., принимающая значение 0, если $z_n < m$; 1, если $z_n \geq m$.
Тогда при регулярном потоке заданий x_n имеет распределение Бернулли с $\alpha_0 = 0.5$. В результате атаки параметр распределения α меняется в большую или меньшую сторону относительно $\alpha_0 = 0.5$.
Задания, поступившие за некоторый промежуток времени, накапливаются в буфере и затем выборочно обслуживаются.
Для обслуживания моделируется с. в., имеющая такое же распределение, как и z_n в регулярном входном потоке заданий. После каждой реализации случайной величины из буфера выбирается задание, по рассматриваемой характеристике ближайшее к ней слева (при $\alpha > \alpha_0$) или справа (при $\alpha < \alpha_0$).

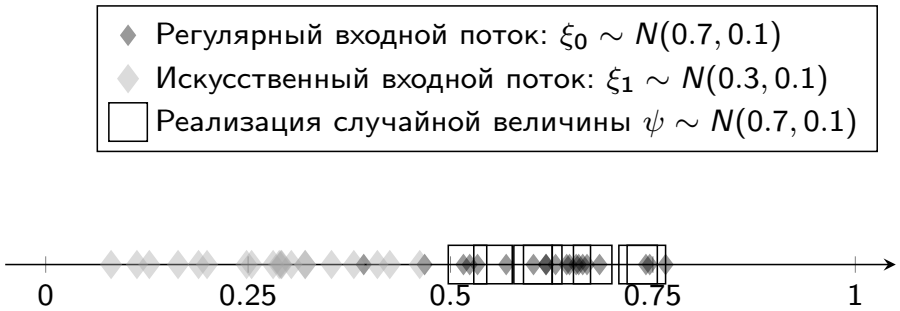


Рис. 4: Реализация случайной величины для выбора заданий из буфера

Параграф 4.4. Вычислительные эксперименты

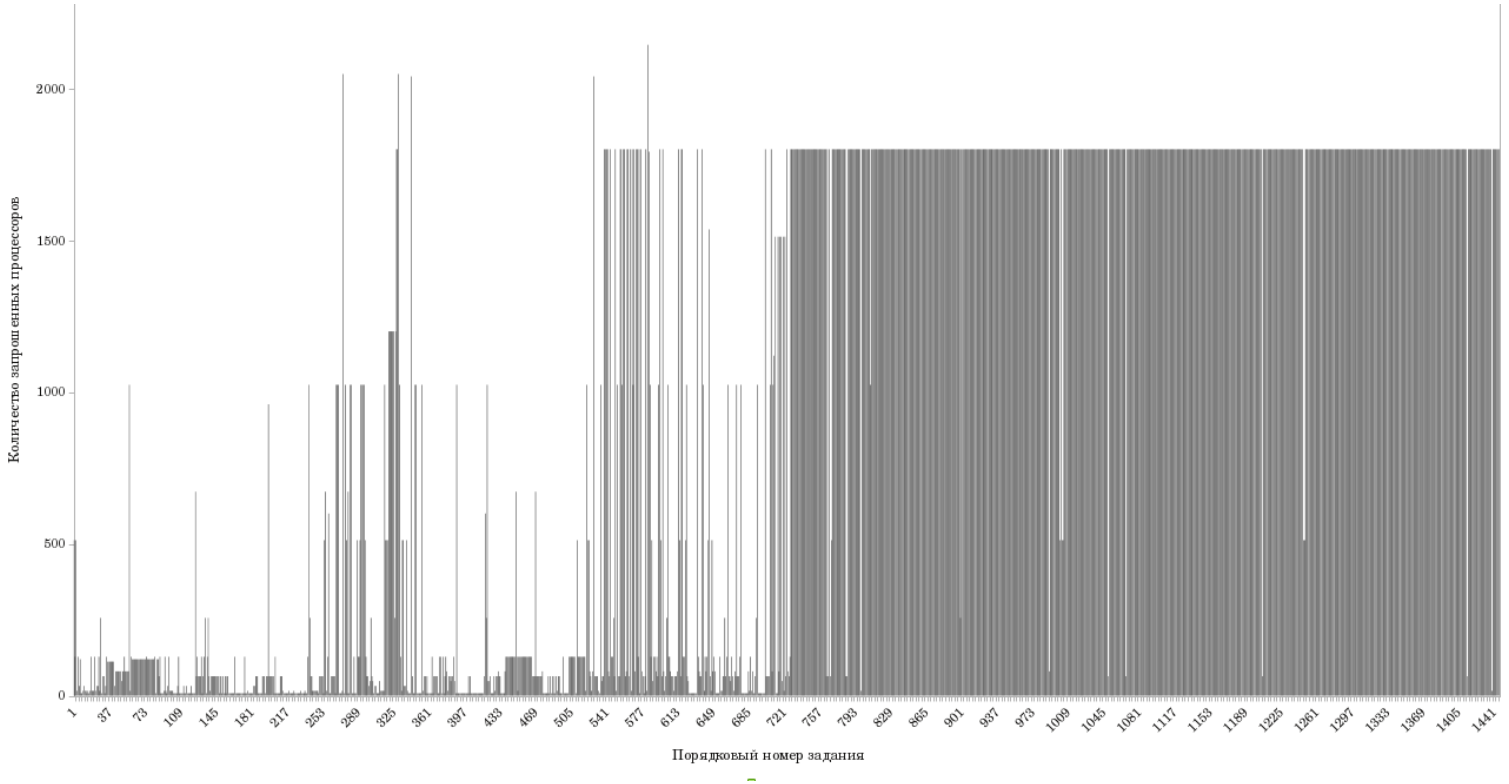


Рис. 5: Зависимость количества запрашиваемых процессоров от порядкового номера задания. Фрагмент истории функционирования вычислительного кластера LLNL Atlas в Ливерморской национальной лаборатории им. Э. Лоуренса (США), иллюстрирующий всплеск активности отдельного пользователя на протяжении 11 часов работы.

Параграф 4.4. Вычислительные эксперименты

Обнаружение момента атаки на вычислительную систему

$x_n = 0$, если количество процессоров, запрашиваемое заданием, меньше $m_1 = 64$;
$x_n = 1$, иначе.
$\alpha_0 = 0.5$ — параметр распределения регулярного потока заданий.
$\alpha = 0.99$ — параметр распределения в присутствие искусственного потока (всплеска активности пользователя).
Значения параметра **b**, характеристик ARL и AD для различных пороговых значений b при наблюдении динамики KC в системе LLNL Atlas:

b	**b**	ARL	AD	b	**b**	ARL	AD
6	187.68	126	6.22	11	344.08	3996	11.6
7	218.96	254	7.3	12	375.36	7932	12.7
8	250.24	508	8.4	13	406.64	15740	13.7
9	281.52	1012	9.4	14	437.92	31228	14.8
10	312.8	2012	10.5	15	469.2	61950	15.9

В предположении, что желаемое среднее количество наблюдений до ложной тревоги $ARL \geq 500$, $b = 312.8$. При имитационном моделировании процесса обнаружения атаки было зафиксировано 27 ложных тревог, среднее количество заданий между которыми составило 526.7.

Параграф 4.4. Вычислительные эксперименты

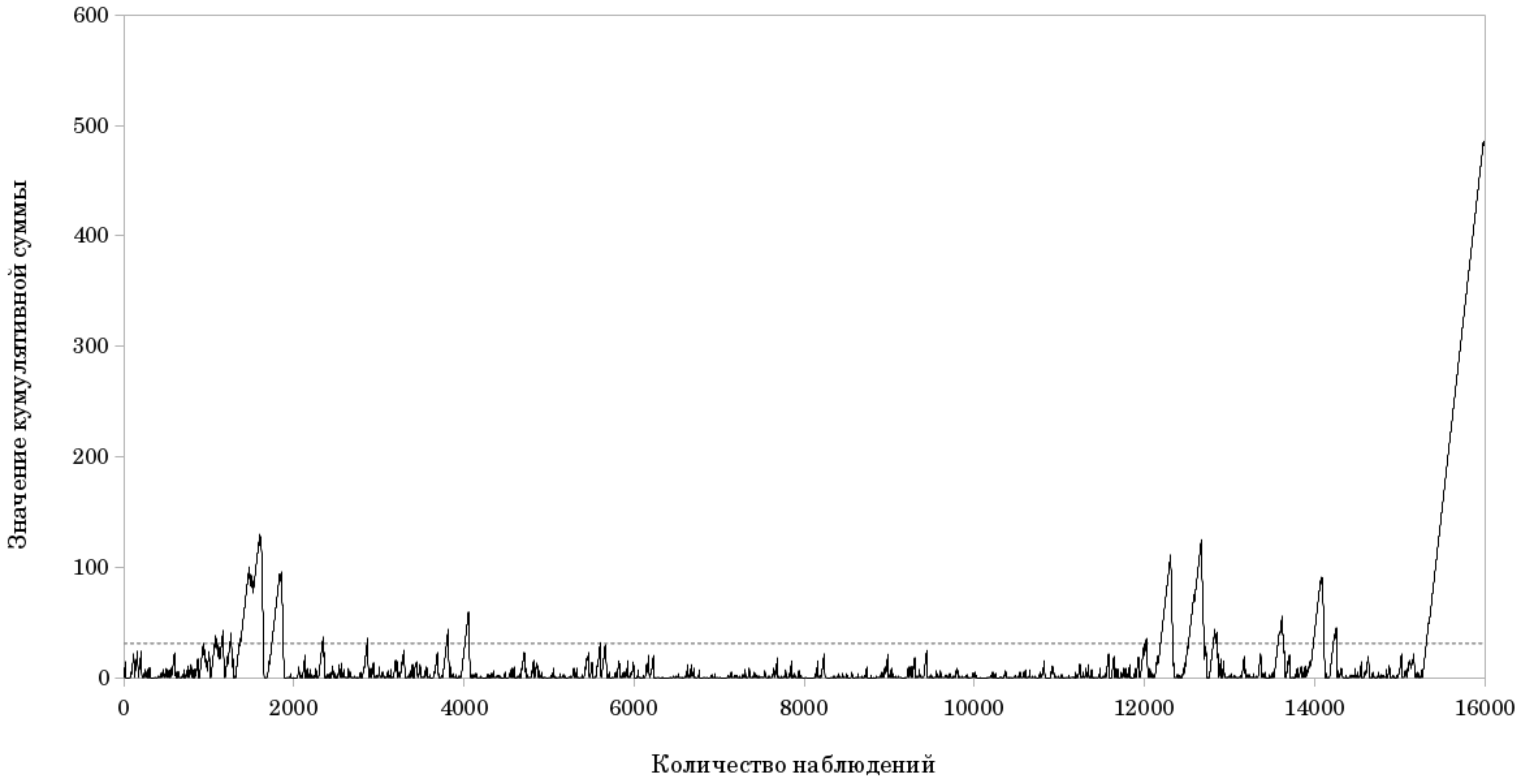
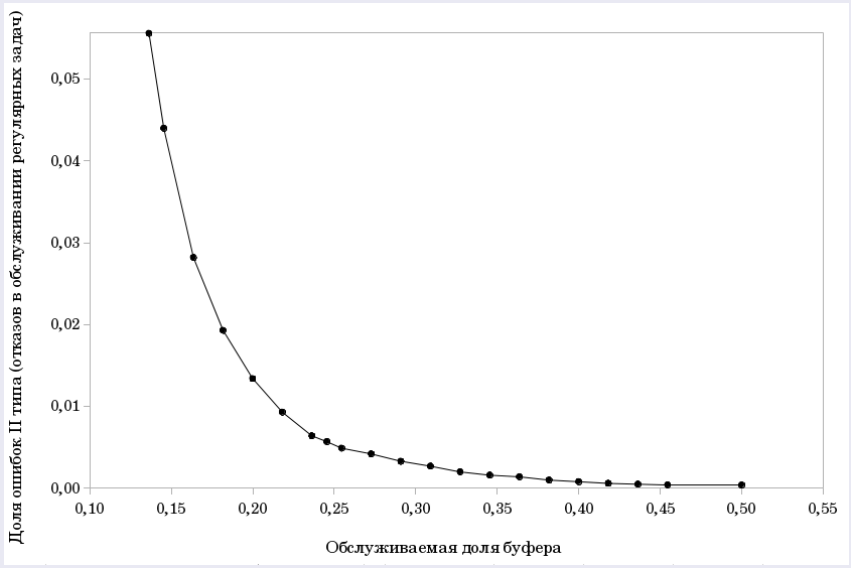


Рис. 6: Динамика кумулятивной суммы на протяжении периода работы с 18 декабря 2006 г. по 9 мая 2007 г. вычислительного кластера LLNL Atlas.

Параграф 4.4. Вычислительные эксперименты

Изменение протокола обслуживания

В предположении, что характеристика регулярного потока заданий имеет нормальное распределение $N(0.7, 0.1)$, а характеристика потока заданий при наличии атаки — нормальное распределение $N(0.3, 0.1)$, были смоделированы потоки из $N_r = 1\,000$ регулярных и $N_a = 10\,000$ искусственных заданий ($N_r + N_a = N$). Для обслуживания из буфера выбиралось **M** заданий. Зависимость доли ошибок II типа (отказов в обслуживании регулярных заданий) от доли буфера M/N:



Апробация результатов

Основные результаты диссертации были представлены на следующих семинарах и конференциях:

1. 13-й 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. 12-я всероссийская конференция «Высокопроизводительные параллельные вычисления на кластерных системах», 26–28 ноября 2012 г., г. Н. 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 BOINC: 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.

Список опубликованных работ по теме диссертации

1. Чернов И. А., Ивашко Е. Е., Никитина Н. Н., Габис И. Е. Численная идентификация модели дегидрирования в грид-системе на базе BOINC // Компьютерные исследования и моделирование. 2013. Т. 5, No 1. С. 37–45.
2. Ивашко Е. Е., Никитина Н. Н. Использование BOINC-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, No 5. С. 71–79.
4. Ивашко Е. Е., Никитина Н. Н. Организация квантовохимических расчетов с использованием пакета Firefly в гетерогенной грид-системе на базе BOINC // 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, No 5. С. 125–129.

Список опубликованных работ по теме диссертации

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. **Мазалов В. В., Никитина Н. Н. Программа изменения дисциплины обслуживания при обнаружении DoS-атаки // Свидетельство о государственной регистрации программы для ЭВМ No. 2013618298 от 05.09.2013.**
8. Никитина Н. Н., Чернов И. А. Разрешимость системы разностных уравнений для динамики кумулятивной суммы // Проблемы анализа. 2013. Т. 2, No 20. С. 68–81.
9. Ивашко Е. Е., Никитина Н. Н. Организация квантовохимических расчетов с использованием пакета Firefly в гетерогенной Грид на базе BOINC // Научный сервис в сети Интернет: экзафлопсное будущее. Труды Международной суперкомпьютерной конференции (Абрау-Дюрсо, 19–24 сентября 2011 г.). 2011. С. 178–181.
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.

Список опубликованных работ по теме диссертации

11. Ивашко Е. Е., Никитина Н. Н. Использование BOINC-грид в вычислительноемких научных исследованиях // XIV Российская конференция с участием иностранных ученых DICR-2012 (Новосибирск, Россия, 26.11 — 30.11.2012): Материалы конференции. Режим доступа: http://conf.nsc.ru/dicr2012/ru/reportview/140625.
12. 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: 2012. P. 44.
13. 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.
14. 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.
15. Чернов И. А., Никитина Н. Н. Математические модели управления очередями заданий в BOINC // Первая российская конференция «Высокопроизводительные вычисления на базе BOINC: фундаментальные исследования и разработки». 2013. Режим доступа: http://boincfast.ru/papers/nikitina.pdf.