*ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2024, № 2, С. 25–42*

## ОБРАБОТКА ИНФОРМАЦИИ И ИДЕНТИФИКАЦИЯ

*УДК 519.252*

# СТОХАСТИЧЕСКИЕ МОДЕЛИ ТРУДОЕМКОСТИ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ. II. ОПИСАНИЕ ВЗАИМОДЕЙСТВИЯ С БАЗАМИ ДАННЫХ[1]

**© 2024 г. А. В. Борисов[a], *, А. В. Иванов[a], ****

[a] *ФИЦ ИУ РАН, Москва, Россия*

**e-mail: ABorisov@frccsc.ru*

***e-mail: AIvanov@frccsc.ru*

Поступила в редакцию 07.06.2023 г.
После доработки 17.07.2023 г.
Принята к публикации 29.01.2024 г.

Работа является второй частью цикла, посвященного формализации описания времени выполнения пользовательских заданий на виртуальных вычислительных узлах. В ней приведен анализ качества предлагаемых моделей, описывающих время обработки информации, хранимой в базах данных. В качестве тестового объекта использован макет системы обезличивания персональных данных пассажиров. Построены стохастические модели трудоемкости решения двух типов заданий: обезличивания персональных данных и вычисления их статистических характеристик. Рассмотрено детальное описание планирования и выполнения нагрузочного тестирования для построения данных моделей. Полученные по реальным данным модели трудоемкости продемонстрировали достаточно высокое качество.

**Ключевые слова:** стохастическая модель, виртуальный вычислительный узел, нагрузочное тестирование, обработка информации в базе данных, обезличивание персональных данных.

**DOI:** 10.31857/S0002338824020032, **EDN:** VOQZTV

# STOCHASTIC MODELS FOR TIME COMPLEXITY OF COMPUTING TASKS: II. DESCRIPTION OF INTERACTION WITH DATABASES

**A. V. Borisov[a], *, A. V. Ivanov[a], ****

[a] *Federal Research Center "Computer Science and Control" of Russian Academy of Sciences*

**e-mail: ABorisov@frccsc.ru,*

***e-mail: AIvanov@frccsc.ru*

The paper contains the second part of an investigation devoted to the design of the mathematical models for the execution time of user tasks carried out on the virtual calculating nodes. We provide the performance of the proposed model for the description of the data processing fulfilled in the databases. As a testbed for stress testing, we choose a prototype of the anonymization system of the passengers' personal data. There are stochastic models describing two types of user tasks: personal data anonymization procedure and calculation of the sample statistical characteristics. The paper contains a detailed description of the stress test planning and fulfillment for both models. The obtained mathematical models developed by the real data demonstrate high performance.

**Keywords:** stochastic model, virtual computational node, stress testing, information processing in databases, anonymization of personal data

**Введение.** Предметом исследования первой части [1] являются методы математического описания времени выполнения пользовательских заданий на виртуальных вычислительных узлах. Подобные математические модели актуальны для оптимизации использования ресурсов в сетях

[1] Работа выполнена с использованием инфраструктуры Центра коллективного пользования «Высокопроизводительные вычисления и большие данные» (ЦКП «Информатика») ФИЦ ИУ РАН (г. Москва).

распределенных/облачных/граничных вычислений, сетях, управляемых вычислениями, и пр. Время выполнения $\tau$, очевидно, зависит от ресурсов узла $x$, характеристик пользовательских заданий $y$, а также текущих значений аппаратно-программной среды узла $z$. Однако основной чертой $\tau$ является наличие статистической неопределенности: время выполнения одного и того же задания на одном и том же узле варьируется от запуска к запуску и может трактоваться как случайная величина: $\tau = \tau(\omega)$. Для решения прикладной задачи оценивания вероятности превышения временем выполнения задания некоторого фиксированного порога достаточно знать моментные характеристики: среднее $\mathcal{M} \triangleq \mathsf{E}\{\tau(\omega)\}$ и дисперсию $\mathcal{D} \triangleq \mathsf{E}\left\{\left(\tau(\omega) - \mathcal{M}\right)^2\right\}$. Очевидно, что $\mathcal{M}$ и $\mathcal{D}$ являются функциями от $(x, y, z)$: $\mathcal{M} = \mathcal{M}_z(x, y)$, $\mathcal{D} = \mathcal{D}_z(x, y)$. Предполагается, что вид функций $\mathcal{M}_z(x, y)$ и $\mathcal{D}_z(x, y)$ известен с точностью до параметров. Задача построения стохастической модели трудоемкости сводится к оцениванию этих параметров по статистической информации, полученной в результате нагрузочного тестирования.

В первой части исследования дано детальное описание векторов $x$, $y$ и $z$, а также свойств функций $\mathcal{M}_z(x, y)$ и $\mathcal{D}_z(x, y)$, предложен конкретный вид функций, пригодных для описания $\mathcal{M}_z(x, y)$ и $\mathcal{D}_z(x, y)$, дана постановка задачи идентификации модели трудоемкости в форме задачи оценивания параметров регрессионной модели и определены требования к статистической информации, необходимой для решения данных регрессионных задач, ее состав, принципы сбора и обработки по результатам проведения нагрузочного тестирования.

Целью настоящей работы является иллюстрация эффективности предложенной модели трудоемкости на примере описания временны́х затрат на обработку информации, хранимой в базах данных.

Статья имеет следующую структуру. В качестве объекта тестирования выступает макет системы обезличивания персональных данных пассажиров [1]. В разд. 1 описаны аппаратные и программные характеристики виртуального узла, на котором установлен макет. Особое внимание уделено задачам, решаемым с помощью специального программного обеспечения (СПО), установленного на узле. Именно они и определяют типы пользовательских заданий, на выполнение которых рассчитан узел. С алгоритмической точки зрения функции макета разделены на две части. Раздел 2 посвящен построению стохастической модели задания 1, реализующего процедуру обезличивания персональных данных. Дано краткое описание результатов выполнения этой процедуры. Изложены результаты нагрузочного тестирования и проанализированы свойства построенной модели. В разд. 3 имеются аналогичные сведения о стохастической модели трудоемкости задания 2 – вычисления выборочных статистических характеристик персональных или обезличенных данных. В Заключении сведены выводы и перспективы дальнейших исследований.

**1. Краткие сведения о тестируемом программном комплексе.** Для демонстрации применимости предложенной модели трудоемкости выполнения пользовательских заданий в качестве примера СПО использован макет системы обезличивания персональных данных пассажиров [1]. Виртуальный узел установлен на основной компьютер со следующими характеристиками: процессор Intel Core i7-3770 3.4 ГГц, ОЗУ 16 Гб, ОС Windows 10 Pro, технология виртуализации Hyper-V. Программное обеспечение (ПО) виртуального узла состоит из:

общесистемного программного обеспечения (ОПО), состоящего из операционной системы Астра Linux CE 2.12 Орел и системы управления базами данных (СУБД) – Postgres 9.6;

СПО, включающего в себя тестовую базу персональных данных, содержащую порядка 26 млн записей и программный компонент, который реализует графический пользовательский интерфейс и основные шаблоны обезличивания, разработанный с помощью средств Microsoft Visual Studio.

Вектор характеристик ресурсов узла $x = (x_1, x_2)$ содержит следующие характеристики:

$x_1$ – количество процессорных ядер (до 4 шт., назначаются узлу поштучно),

$x_2$ – объем оперативной памяти (до 8 Гб, назначается узлу с шагом 0.5 Гб).

СПО виртуального узла разработано для выполнения следующих целевых функций:

1) выбор массива необезличенных данных с помощью системы фильтров,

2) решение задачи обезличивания набора выбранных данных в соответствии с различными шаблонами обезличивания полей (исключение, группировка, случайная перестановка, зашумление, синтез искусственных данных),

3) вычисление уровня анонимности для выбранных комбинаций полей,

4) вычисление корреляции выбранных пар полей,

5) вычисление уровня полезности выбранных комбинаций полей,

6) построение гистограммы значений выбранного поля.

В качестве исследуемых типов заданий выбраны

комбинация функций 1 – 2: выбор массива необезличенных данных с последующим их обезличиванием; задание характеризуется единственным параметром – своим объемом $y_1$,

комбинация функций 1 – 4: выбор массива данных с последующим вычислением выборочной ковариации пары столбцов; задание также характеризуется единственным параметром – своим объемом $y_1$.

Данный выбор продиктован следующими причинами. Во-первых, при эксплуатации комплекса функция 1 будет задействована практически всегда: обезличивание и вычисление статистических характеристик всегда выполняются для некоторого подмножества данных, которые должны быть предварительно выбраны с помощью механизма фильтров. Во-вторых, по своей алгоритмической структуре целевые функции 2 – 6 могут быть разделены на две части. В первую входят функции, реализация которых предполагает выполнение некоторой "линейной" последовательности операций. К таким относятся функции 2 и 5. Во вторую группу входят функции 3, 4 и 6, реализуемые с помощью стандартных оптимизированных процедур СУБД. Целью данной части исследования является не построение исчерпывающей математической модели конкретного макета программного комплекса системы обезличивания, а лишь демонстрация эффективности использования предложенной стохастической модели для описания функционирования СПО обработки информации, хранящейся в базе данных. Поэтому для построения моделей были выбраны по одному представителю из этих двух групп. Можно ожидать, что модели этих задач будут различаться видом зависимости характеристик времени выполнения как от ресурсов узла, так и от объема выполняемых заданий.

**2. Модель трудоемкости задания 1 обезличивания данных.** Задача обезличивания данных состоит в выборе группы записей необезличенных данных из базы данных (БД), применении к ним шаблонов обезличивания и последующем сохранении результатов в БД. Обработка записей производится в оперативной памяти, что обуславливает высокие требования этой задачи к объему памяти виртуального узла. Обработка ведется последовательно, поэтому задача не выигрывает от наличия дополнительных процессорных ядер. Исходные данные содержат целочисленное значение "Возраст", значения даты/времени "Дата начала", "Дата конца", вещественные значения "Результат 1", "Результат 2", "Результат 3", категориальные значения "Категория 1", "Категория 2", "Категория 3", "Категория 4", "Аэропорт", "Компания". Все записи также содержат поле "Порядковый номер". Выбор данных производится по критерию принадлежности номера записи указанному диапазону. В рассматриваемом примере выполняется обезличивание всех полей исходного набора данных.

В зависимости от типа поля для его обезличивания могут применяться преобразования различного типа – так называемые *шаблоны*. Шаблон разбиения применим для числовых полей. В качестве результата этот шаблон выдает некоторый интервал, которому принадлежит исходное значение. Например, шаблон разбиения может быть применим к полю "Возраст", и в качестве результата выдает тот из предопределенных интервалов возрастов, в который попадает исходное значение возраста.

Шаблон зашумления также применяется к числовым полям и их наборам. Он заключается в добавлении к исходному значению поля независимого шума, аддитивного или мультипликативного, имеющего известное распределение. Например, результатом применения шаблона зашумления к полю "Уровень сахара в крови" является сумма исходного значения и независимой центрированной случайной ошибки, содержащей равномерное распределение с параметром, отвечающим за уровень анонимности.

Шаблон синтеза применим к числовым и словарным полям, а также их наборам. По всей совокупности исходных значений строится их эмпирическое распределение. Результатом применения этого шаблона являются сгенерированные значения, распределение которых совпадает с эмпирическим.

В данном комплексе к полю "Возраст" применяется шаблон разбиения, к полям "Дата начала", "Дата конца", "Результат 1", "Результат 2", "Результат 3", "Категория 1", "Категория 2", "Категория 3", "Категория 4" – шаблон зашумления, к полям "Аэропорт", "Компания" – шаблон синтеза.

Для построения модели трудоемкости было выполнено нагрузочное тестирование. Как уже было сказано, виртуальный вычислительный узел содержал от 1 до 4 процессорных ядер и ОЗУ объемом от 1 до 8 Гб с шагом 0.5 Гб. Количество записей, подвергаемых обезличиванию, варьировалось от 1 000 до 350 000. План экспериментов представлен на рис. 1.

На нем точками обозначены пары "объем ОЗУ – объем задания" $(x_2, y_1)$, испытания с заданной парой проводились для всех возможных вариантов количества процессорных ядер: от

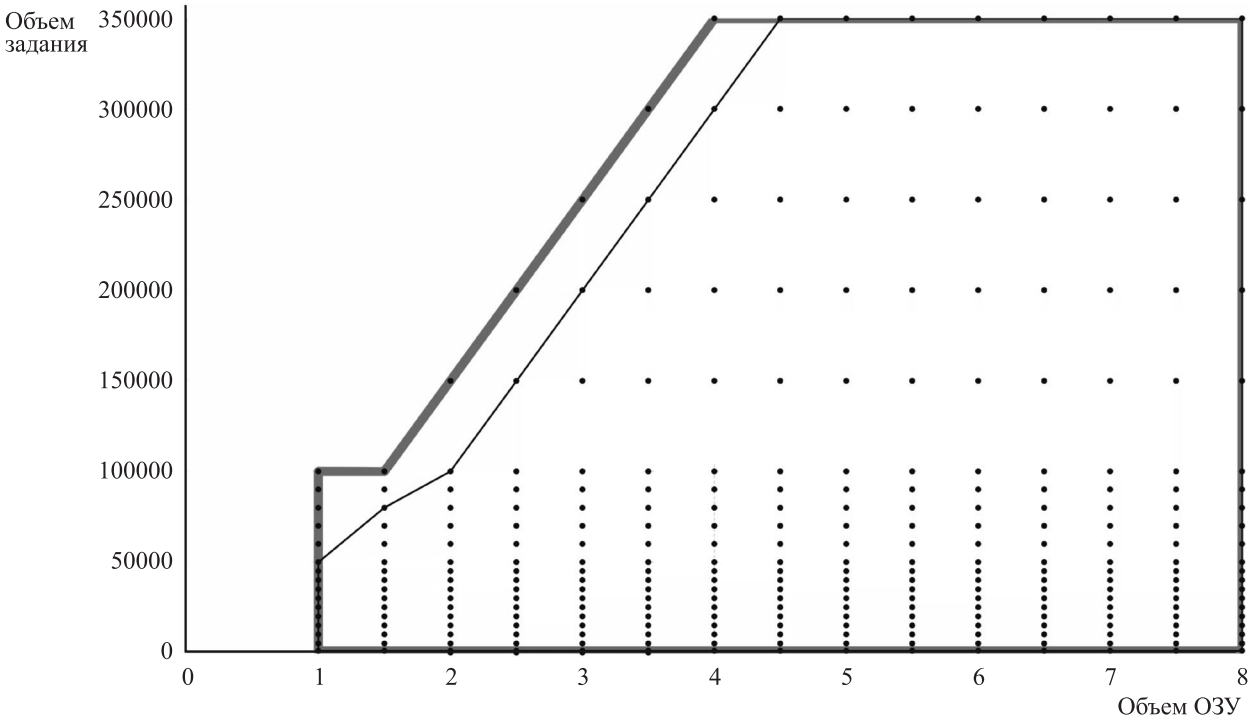


**Рис. 1.** Проекция области определения модели задания 1

1 до 4. Жирной серой линией обведена вся область аргументов $(x_2, y_1)$, для которых проводилось тестирование. Черная линия показывает границу эффективного функционирования узла.

Для вычисления выборочных моментных характеристик – среднего значения времени $\tilde{\mathcal{M}}_r$ и его дисперсии $\tilde{\mathcal{D}}_r$ – каждый опыт с фиксированными параметрами $(x_1, x_2, y_1)$ проводился независимо 10 раз. В качестве возможных вариантов модели среднего времени выполнения заданий и его дисперсии выступали [1]

$$f^1(x_1, x_2, y_1) = \alpha + \exp[\beta + \gamma_1 x_1 + \gamma_2 x_2 + \varepsilon_1 y_1] \tag{2.1}$$

– экспоненциальная модель,

$$f^2(x_1, x_2, y_1) = \alpha + \beta x_1^{\gamma_1} x_2^{\gamma_2} y_1^{\varepsilon_1} \tag{2.2}$$

– степенная модель,

$$\begin{gathered} f^3(x_1, x_2, y_1) = \alpha + \beta x_1^{\gamma_1} x_2^{\gamma_2} \times \\ \times\left(\delta_{11} y_1 + \delta_{12} \max(0, y_1 - \mu_1) + \varepsilon_1\right) \end{gathered} \tag{2.3}$$

– степенная модель с кусочно-линейной зависимостью по $y$. В выбранных функциях $\alpha$, $\beta$, $\{\gamma_i\}$, $\varepsilon_1$, $\{\delta_{ij}\}$ и $\mu_1$ являются параметрами, подлежащими оцениванию.

Нагрузочное тестирование было выполнено в соответствии с планом, графическая иллюстрация которого приведена на рис. 1. Результаты этого тестирования – выборочные средние и дисперсии времени для случая 1 процессорного ядра – приведены на рис. 2. Здесь черными кружками показаны данные, оставленные после 1-го этапа оценивания, звездочками – данные, отсеянные после 1-го этапа, серая линия – граница области тестирования, черная линия – граница эффективного функционирования узла.

Из данного рисунка можно видеть, что при некоторых значениях аргументов $(x_2, y_1)$ среднее время выполнения задания и его дисперсии недопустимо велики, что говорит о неэффективном функционировании узла. В случае ОЗУ недостаточного объема используется механизм свопинга. Следует отметить, что под свопингом подразумевается один из механизмов

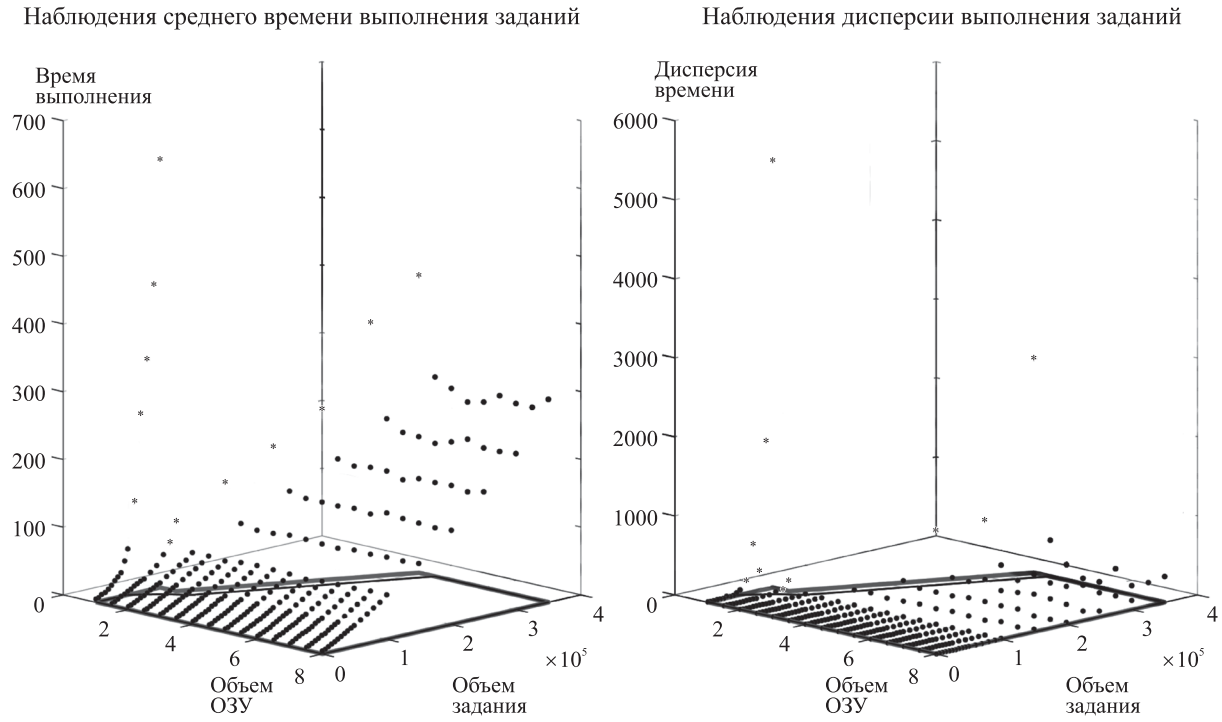


**Рис. 2.** Статистические данные нагрузочного тестирования задания 1 для случая 1 процессорного ядра

функционирования виртуальной памяти, в случае объема ОЗУ, недостаточного для размещения всего объема обрабатываемых данных. При свопинге происходит динамическая перекачка неактивных данных во вторичное хранилище (например, на жесткий диск), обладающее меньшей скоростью обмена по сравнению с ОЗУ. Одновременно с этим происходит загрузка в ОЗУ активных сегментов данных. Процедура свопинга обеспечивает выполнение задания, но существенно замедляет его по сравнению со случаем, когда все данные помещаются в ОЗУ.

Для идентификации параметров модели в качестве функции потерь решено взять абсолютную величину ошибки оценки, и задачу построения модели можно свести к решению следующих оптимизационных задач:

$$\sum_{r=1}^{R}\left|\tilde{\mathcal{M}}_r - f^i\left(x_1^r, x_2^r, y_1^r\right)\right| \to \min_{i,\alpha,\beta,\{\gamma_i\},\varepsilon_1,\{\delta_{ij}\},\mu_1}, \tag{2.4}$$

$$\sum_{r=1}^{R}\left|\tilde{\mathcal{D}}_r - f^i\left(x_1^r, x_2^r, y_1^r\right)\right| \to \min_{i,\alpha,\beta,\{\gamma_i\},\varepsilon_1,\{\delta_{ij}\},\mu_1}. \tag{2.5}$$

Была выбрана двухэтапная процедура идентификации параметров модели. На обоих этапах идентификации задачи оптимизации решались численно с помощью алгоритма Нелдера – Мида [3], являющегося одним из стандартных в пакете прикладных программ Matlab. Его применение было продиктовано, в частности, необходимостью решать задачу негладкой оптимизации. На первом этапе выполнялся разведочный регрессионный анализ: была построена оценка наименьших модулей параметров экспоненциальной модели (2.1). Из него следовали два вывода. Во-первых, на некоторой области значений $(x_2, y_1)$ среднее время выполнения задания недопустимо большое. Это означает, что в указанной области виртуальный узел не может эффективно выполнять свои функции. Во-вторых, точность приближения, обеспечиваемая моделью (2.1), оказалась неприемлемой.

Было решено сузить область нагрузочного тестирования (см. рис. 1, 2: область с черной границей). Статистические данные о выборочных моментах $(\tilde{\mathcal{M}}_r, \tilde{\mathcal{D}}_r)$ в этой области позволяют ожидать приемлемых значений времени выполнения заданий. На этой области были решены задачи оптимизации (2.4) и (2.5), обеспечивающие построение модели трудоемкости задачи обезличивания. И для среднего времени, и дисперсии лучшими оказались модели сте-

пенной зависимости $f^2(\cdot)$. Ниже приведены их явные формулы, а также соответствующие коэффициенты детерминации $R^2$ [4]:

$$\mathcal{M}_Z(x_1, x_2, y_1) = 6.356 \times 10^{-5} \times$$
$$\times x_1^{-0.058} x_2^{-0.030} y_1^{1.190} + 4.952, R^2_{\mathcal{M}} = 0.996,$$

$$\mathcal{D}_Z(x_1, x_2, y_1) = 1.360 \times 10^{-15} \times$$
$$\times x_1^{-0.043} x_2^{-0.280} y_1^{3.141} + 0.808, R^2_{\mathcal{D}} = 0.726.$$

Следует также отметить, что задачи оптимизации (2.4) и (2.5) были решены без ограничений на оптимизируемые параметры. Это было сделано намеренно для проверки свойств функций $\mathcal{M}(\cdot)$ и $\mathcal{D}(\cdot)$ по переменным $X$ и $Y$, представленных в [1].

Визуально качество оценивания можно оценить по рис. 3 и 4: на них представлены измерения, полученные в результате нагрузочного тестирования, и построенные на их основе модели. На рисунках точками показаны результаты нагрузочного тестирования, поверхностью – построенная модель.

Так как $\mathcal{M}_Z(x_1, x_2, y_1)$ и $\mathcal{D}_Z(x_1, x_2, y_1)$ являются функциями трех переменных, то на рисунках изображены $\mathcal{M}_Z(1, x_2, y_1)$ и $\mathcal{D}_Z(1, x_2, y_1)$ – "сечения" моделей, построенные для 1 процессорного ядра; качество оценивания для другого количества ядер аналогично.

Анализируя рис. 3 и 4, можно сделать следующие заключения. Во-первых, невязки модели дисперсии значительно больше невязок модели среднего времени. Во-вторых, в области малого объема ОЗУ можно увидеть резкий рост как среднего, так и дисперсии времени выполнения задания даже при малом его объеме. Именно это обстоятельство вынудило сократить область возможных значений ресурсов узла, обеспечивающих эффективное выполнение пользовательских заданий № 1. Причиной такого замедления и нестабильной работы является недостаток оперативной памяти и активное использование механизма свопинга.

Для более детального анализа исследовались двумерные сечения модели трудоемкости. Графически зависимость среднего выполнения $\mathcal{M}(\cdot, y_1)$ и дисперсии $\mathcal{D}(\cdot, y_1)$ от объема выполняемого задания $y_1$ при некоторых фиксированных значениях ресурсов $(x_1, x_2)$ представлена на рис. 5 и 6. На рисунках ломаными с маркерами показаны результаты нагрузочного тестирования, кривыми – построенная модель. Графики данных зависимостей при большем числе процессорных ядер мало отличимы от случая 1 ядра, о чем будет более подробно сказано ниже.

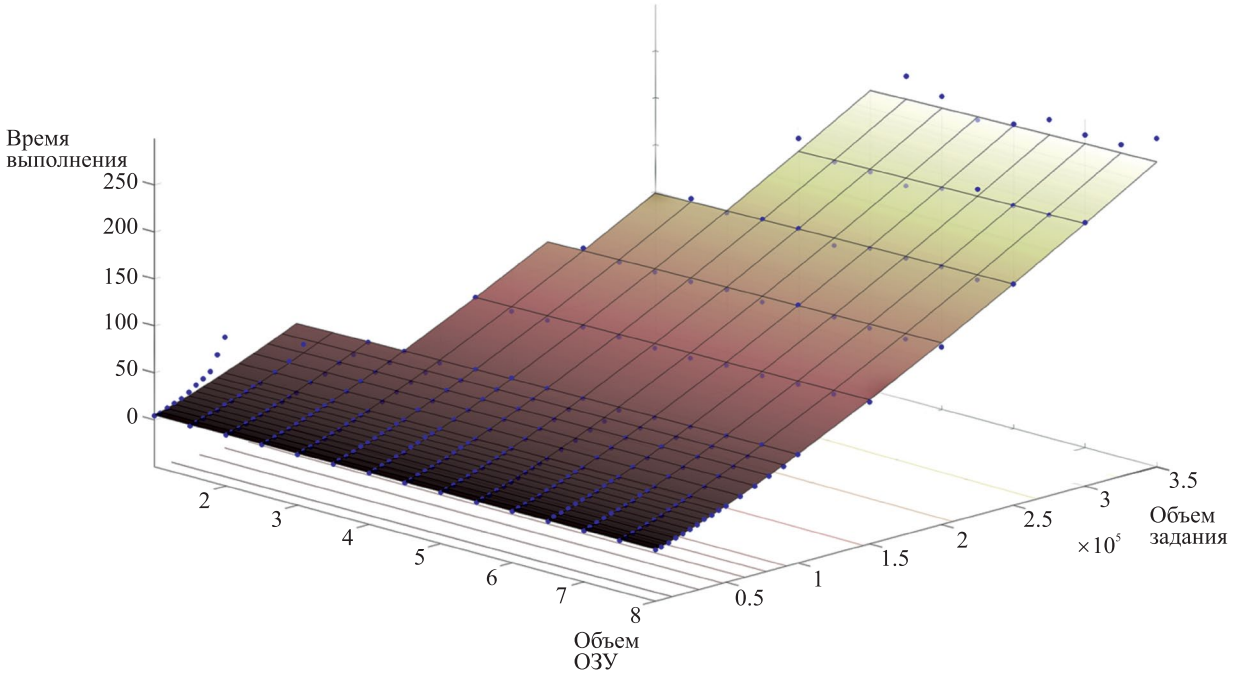


**Рис. 3.** Сечение модели $\mathcal{M}(\cdot)$ для случая 1 ядра

Данные рисунки позволяют сделать следующие заключения. Во-первых, на основании полученных наблюдений среднего и дисперсии времени выполнения задания №1 модели $\mathcal{M}(\cdot)$ и $\mathcal{D}(\cdot)$ на исследуемой области обладают свойством неотрицательности ([1, свойство 1]). Во-вторых, модель $\mathcal{M}(\cdot)$ также удовлетворяет свойствам 3 и 5 из [1]: неубыванию и выпуклости по переменной $Y$.

На рис. 7 и 8 представлены зависимости среднего и дисперсии времени выполнения задания в зависимости от объема ОЗУ для различных фиксированных объемов задания при использовании 1 процессорного ядра. На рисунках ломаными с маркерами показаны результа-

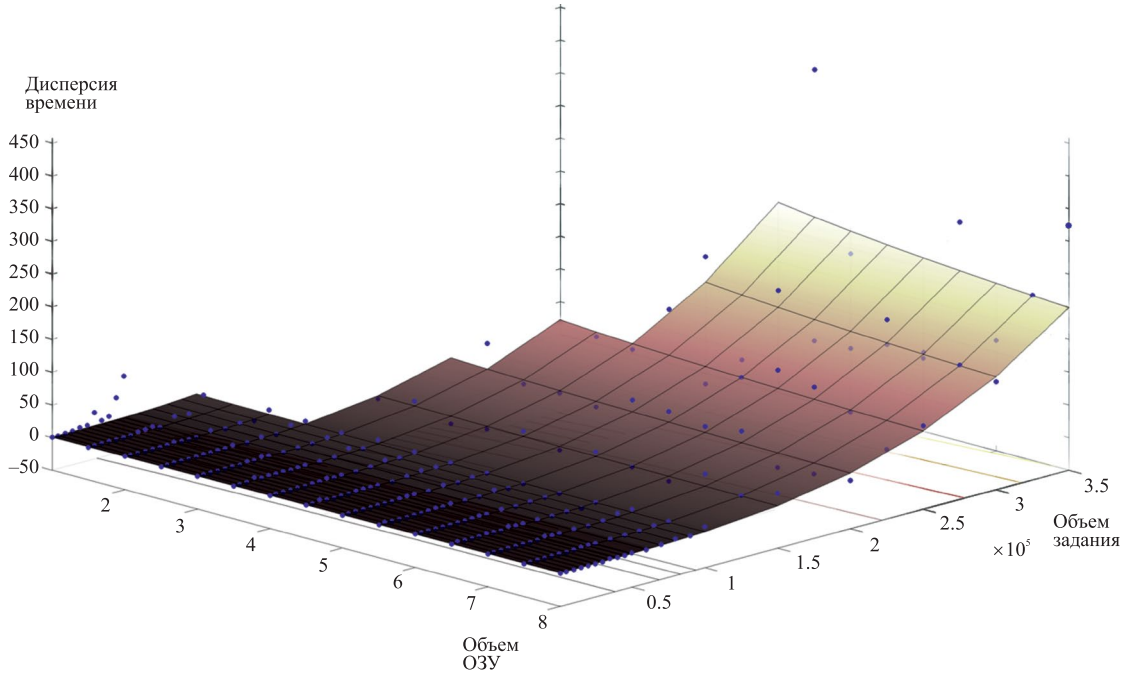


**Рис. 4.** Сечение модели $\mathcal{D}(\cdot)$ для случая 1 ядра

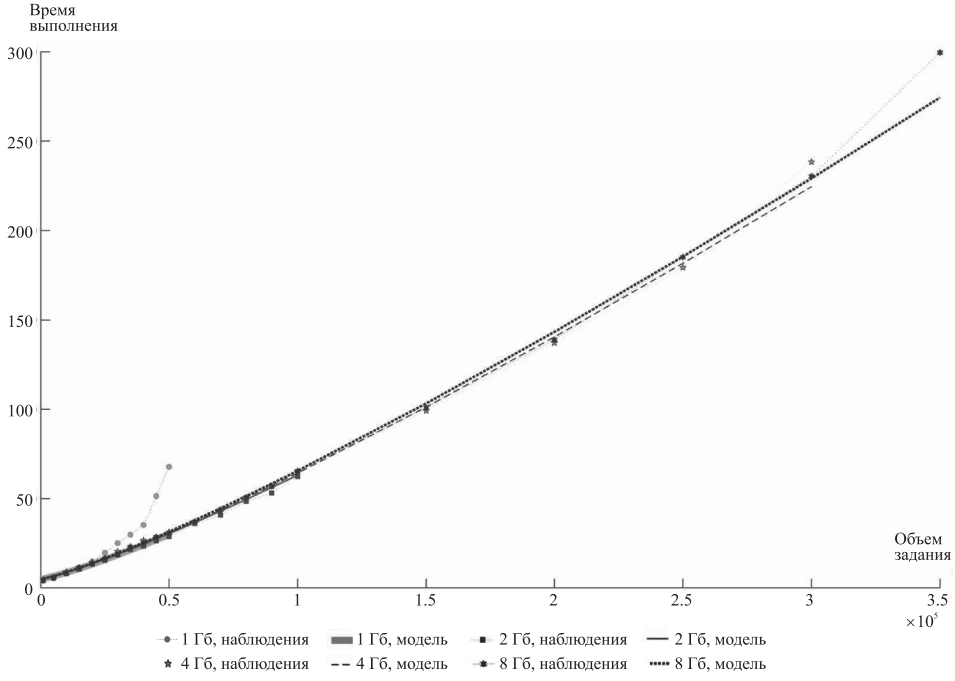


**Рис. 5.** Зависимость $\mathcal{M}(\cdot)$ от объема задания для различных фиксированных объемов ОЗУ (1 ядро)

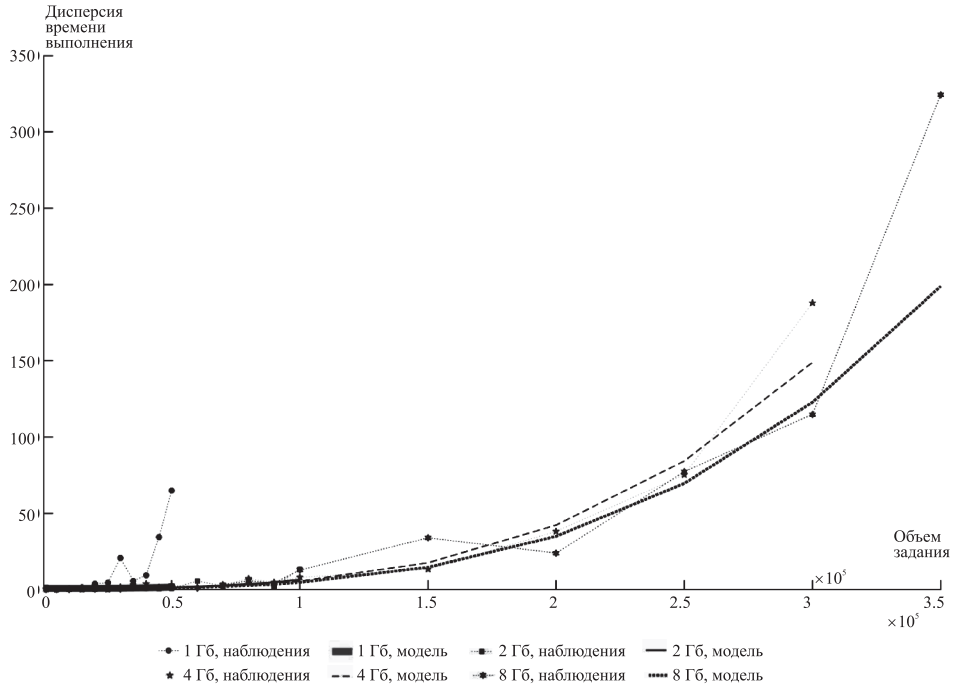


**Рис. 6.** Зависимость $\mathcal{D}(\cdot)$ от объема задания для различных фиксированных объемов ОЗУ (1 ядро)

ты нагрузочного тестирования, кривыми – построенная модель. Графики зависимостей при большем числе ядер мало отличимы от случая 1 ядра.

Анализ данных рисунков ведет к следующим заключениям. Начиная с объема ОЗУ 4,5 Гб все задание умещается в него, и механизм свопинга, существенно задерживающий процесс выполнения заданий, в этом случае не используется. По этой причине на рис. 3 и 7 можно заметить практически постоянное среднее время, не зависящее от объема ОЗУ, для заданий объемом до 250 000. В то же время для объемов 300 000 и 350 000 наблюдения демонстрируют

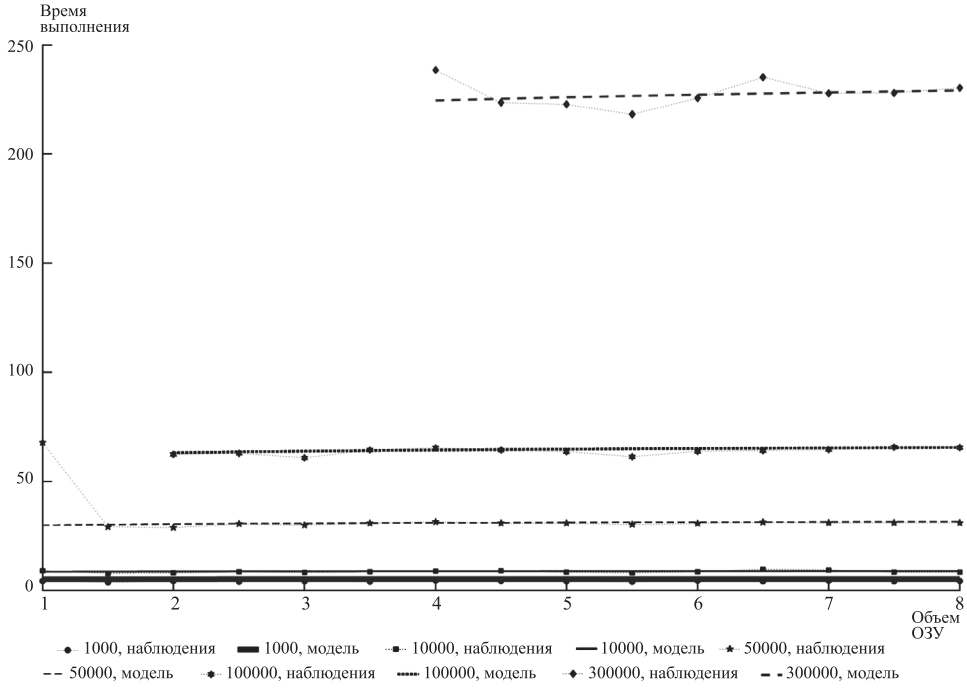


**Рис. 7.** Зависимость $\mathcal{M}(\cdot)$ от объема ОЗУ для различных фиксированных объемов задания (1 ядро)

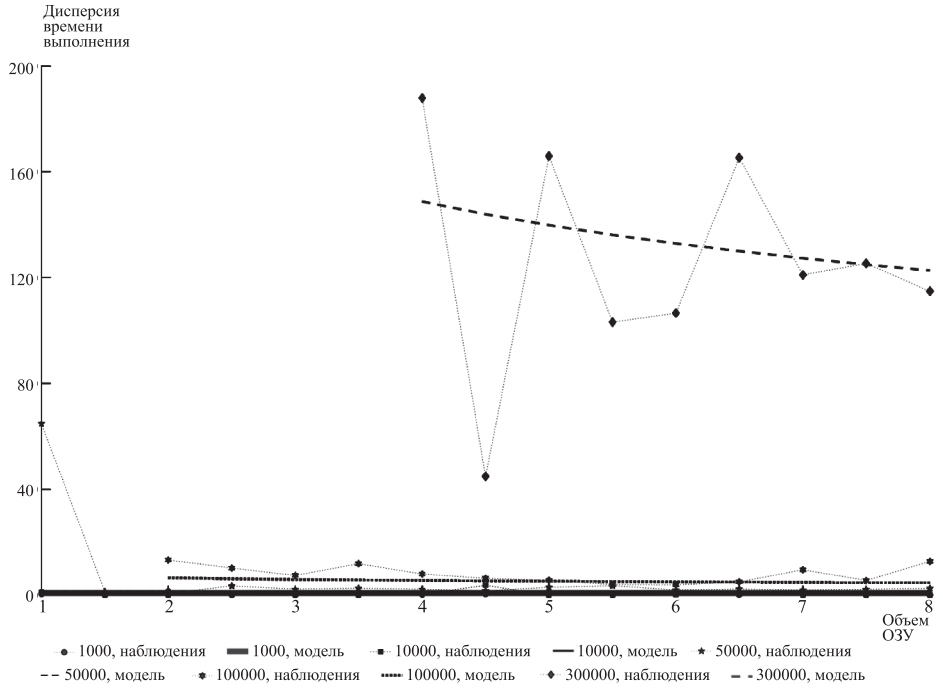


**Рис. 8.** Зависимость $\mathcal{D}(\cdot)$ от объема ОЗУ для различных фиксированных объемов задания (1 ядро)

некоторый рост среднего времени выполнения заданий с ростом объема ОЗУ. Заметим, что оцененный показатель степени при $x_2$ мал по модулю (равен 0.03), но имеет положительный знак. Таким образом, построенная модель среднего времени не обладает свойством локального невозрастания по переменным $X$, описывающим ресурсы узла, т.е. имеет место нарушение свойства 2 из [1]: выбранная модель $f^2(\cdot)$ среднего времени, идентифицированная по полученным наблюдениям, строго возрастает с ростом ресурса $x_2$. Для этого имеются две причины. Во-первых, задачи оптимизации (2.4) и (2.5), решаемые для идентификации параметров моделей $\mathcal{M}$ и $\mathcal{D}$, не содержали ограничений на переменные $\gamma_1$, $\gamma_2$: введение дополнительных ограничений $\gamma_1, \gamma_2 \leq 0$ обеспечило бы выполнение свойства 2 из [1]. Во-вторых, класс функций $f^1(\cdot)$, $f^2(\cdot)$ и $f^3(\cdot)$, выбранный для описания модели, недостаточно гибок для описания свойства 2 из [1]. Для наблюдений выборочной дисперсии и ее модели, построенной по данным нагрузочных испытаний, ситуация иная: для заданий большого объема (300 000 и более) наблюдается тенденция к снижению дисперсии с ростом ОЗУ. Построенная модель отражает эту тенденцию: показатель степени при $x_2$ хотя и относительно мал по абсолютной величине, но имеет отрицательный знак: $\gamma_2 = -0.28$. По данным рисункам можно сделать следующий качественный вывод. На некоторых областях рост какого-либо ресурса (в данном случае объема ОЗУ) может нисколько не снижать среднее время выполнения задания, но в некоторых случаях даже увеличивает его.

На рис. 9 представлена зависимость среднего времени выполнения задания от числа процессорных ядер при фиксированном объеме ОЗУ 4 Гб для разных фиксированных объемов заданий. На рисунке ломаными с маркерами показаны результаты нагрузочного тестирования, кривыми – построенная модель. Зависимость дисперсии имеет идентичный характер и на отдельный рисунок не вынесена.

Анализируя рисунок, можно заметить, что среднее время практически постоянно и не зависит от числа процессорных ядер.

На практике пользователи узла точно знают целевые функции СПО узла и имеют общие представления об алгоритмах, реализующих эти функции. Проанализируем полученную модель обезличивания с этих позиций, используя лишь здравый смысл и не привлекая анализ конкретного кода.

Все действия по обезличиванию производятся одновременно со всеми записями данных, но последовательно для каждого типа полей. На первом этапе выполняется выборка данных, подлежащих обезличиванию. На каждом последующем этапе к соответствующей группе полей применяется свой шаблон обезличивания: группировка, зашумление и синтез. Все эти действия последовательны и не допускают какого-либо распараллеливания. Следует ожидать,

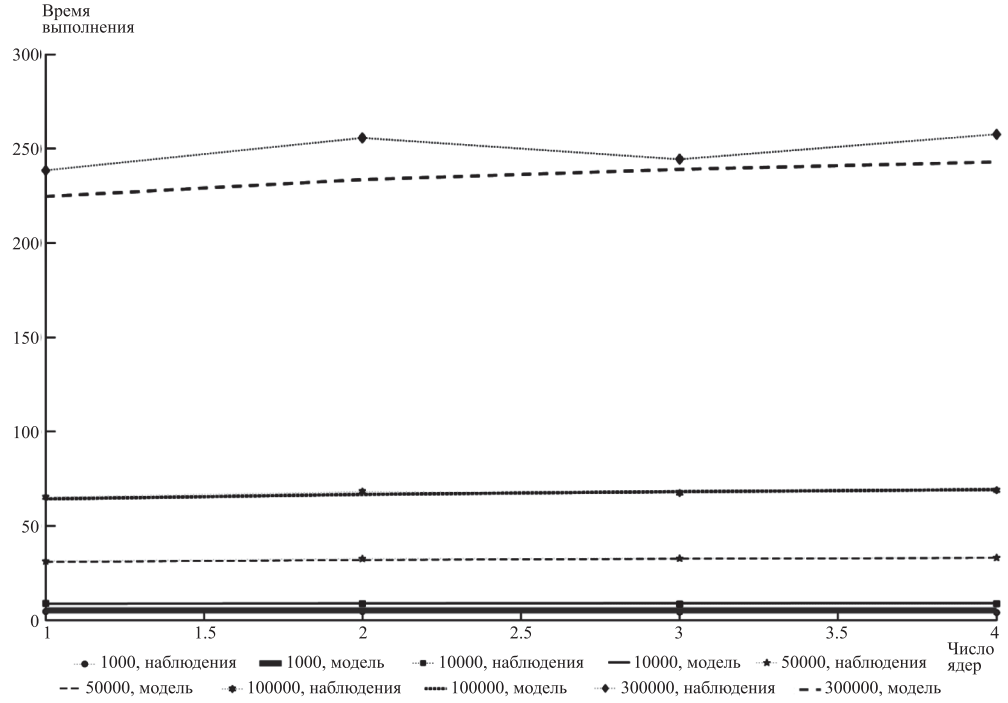


**Рис. 9.** Зависимость $\mathcal{M}(\cdot)$ от числа процессорных ядер для различных фиксированных объемов задания (ОЗУ 4 Гб)

что увеличение числа используемых процессорных ядер не приведет к значимому снижению времени выполнения задания, и даже может незначительно его увеличить из-за наличия "накладных расходов" на механизм распараллеливания. В то же время с ростом объема ОЗУ, после того, как его стало хватать для размещения всех данных задания, дальнейший его рост практически не влияет на среднее время выполнения, однако может несколько уменьшить дисперсию времени выполнения. Результаты нагрузочного тестирования и построенная модель подтверждают данные логические выводы.

**3. Модель трудоемкости задания 2 обработки статистических запросов.** Задача выполнения статистических запросов состоит в выборе группы записей из БД и вычислении выборочной ковариации определенной пары полей. При вычислении ковариации используются возможности СУБД, поэтому достигается высокая степень параллелизма выполнения запросов, и наличие дополнительных процессорных ядер дает значимый выигрыш во времени выполнения. Минимальное время выполнения достигается при условии нахождения всех обрабатываемых данных в файловом кэше операционной системы, поэтому эта задача предъявляет высокие требования к объему памяти виртуального узла. В выбранном запросе производится расчет выборочной ковариации полей "Дата начала" и "Дата конца" необезличенного набора данных. Поскольку эти поля имеют тип "дата/время", то предварительно они приводятся к вещественному числовому типу. Выбор данных выполняется аналогично задаче обезличивания – по критерию принадлежности номера записи указанному диапазону.

Для построения стохастической модели трудоемкости было выполнено нагрузочное тестирование. В нем виртуальный вычислительный узел содержал от 1 до 4 процессорных ядер и ОЗУ объемом от 3.5 до 8 Гб с шагом 0.5 Гб. Задание полностью описывалось числом записей $y_1$, используемых для вычисления выборочной ковариации; оно варьировалось от 2 до 20 млн. План экспериментов представлен на рис. 10.

На нем точками обозначены пары "объем ОЗУ – объем задания" $(x_2, y_1)$, испытания с заданной парой проводились для всех возможных вариантов количества процессорных ядер: от 1 до 4. Жирной серой линией обведена вся область аргументов $(x_2, y_1)$, для которых проводилось тестирование. Черная линия показывает границу эффективного функционирования узла.

Для вычисления выборочных моментных характеристик – среднего значения времени $\tilde{\mathcal{M}}_r$ и его дисперсии $\tilde{\mathcal{D}}_r$ – каждый опыт с фиксированными параметрами $(x_1, x_2, y_1)$ проводился независимо 40 раз. В качестве возможных вариантов модели среднего времени выполне-

ния заданий и его дисперсии выступали функции $f^1(\cdot), f^2(\cdot), f^3(\cdot)$, описываемые формулами (2.1) – (2.3) соответственно.

Статистические результаты тестирования – выборочные средние и дисперсии при использовании 1 ядра приведены на рис. 11. Здесь черными кружками показаны данные, оставлен-

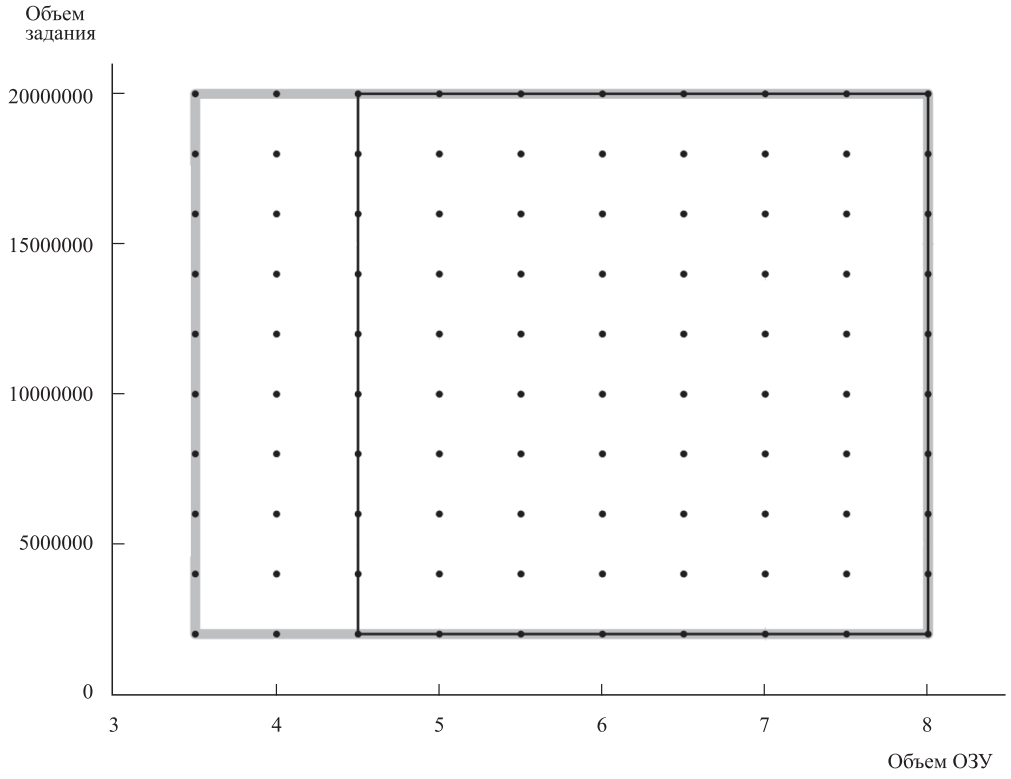


**Рис. 10.** Проекция области определения модели задания 2

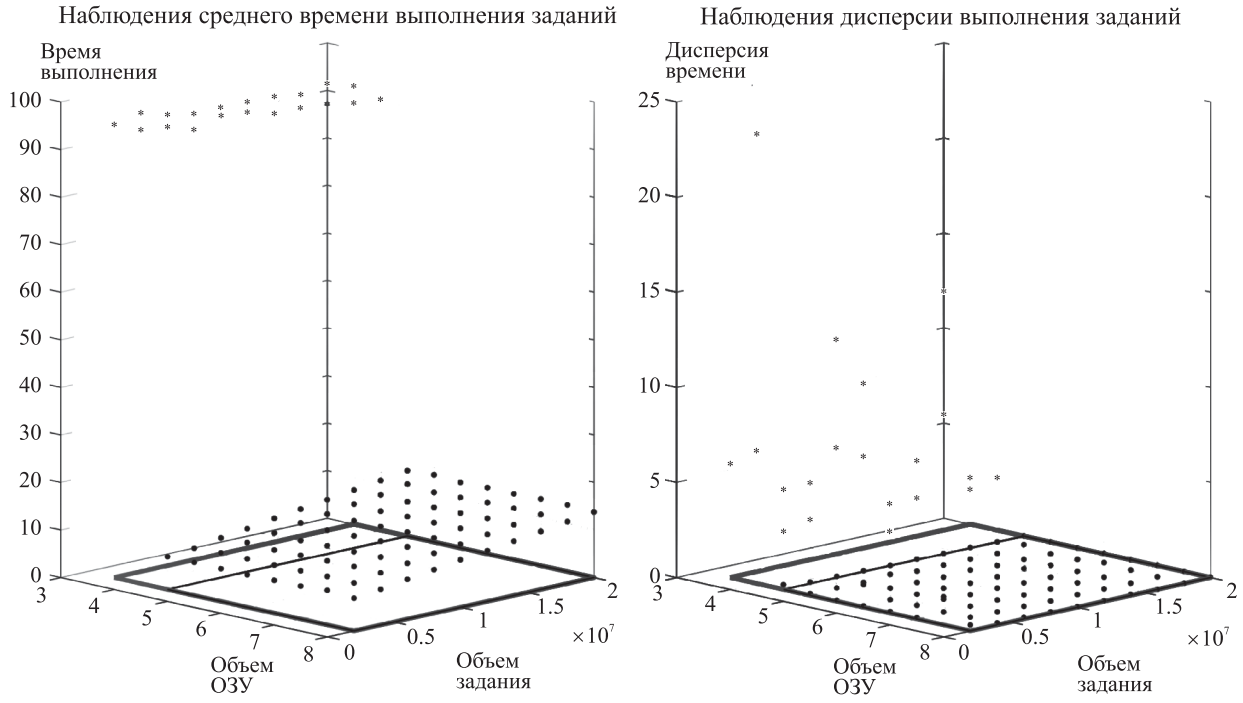


**Рис. 11.** Статистические данные нагрузочного тестирования задания 2 для случая 1 процессорного ядра

ные после 1-го этапа оценивания, звездочками – данные, отсеянные после 1-го этапа, серая линия – граница области тестирования, черная линия – граница эффективного функционирования узла. Случаи 2 – 4 ядер выглядят аналогично.

Для идентификации параметров модели необходимо решить оптимизационные задачи (2.4) и (2.5). Вновь использовалась двухэтапная процедура, как и в предыдущем разделе.

На первом этапе выполнен разведочный регрессионный анализ: построена оценка наименьших модулей параметров экспоненциальной модели (2.1). Из него следовали два вывода. Во-первых, на некоторой области значений $(x_2, y_1)$ среднее время выполнения задания недопустимо большое. Это означает, что на указанной области виртуальный узел не может эффективно выполнять свои функции. Причиной неэффективности является объем ОЗУ, недостаточный для того, чтобы все обрабатываемые данные поместились в кэш. Во-вторых, точность приближения, обеспечиваемая моделью (2.1), оказалась неприемлемой.

На втором этапе область нагрузочного тестирования была сужена (см. рис. 10, 11: область с черной границей). Статистические данные о выборочных моментах $(\tilde{\mathcal{M}}_z^r, \tilde{\mathcal{D}}_z^r)$ в этой области позволяют ожидать приемлемых значений времени выполнения заданий. На области были решены задачи оптимизации (2.4) и (2.5), обеспечивающие построение модели трудоемкости задания 2. Для среднего времени это оказалась функция $f^3(\cdot)$ со степенной зависимостью по переменным $x$ и кусочно-линейной зависимостью по $y$. Более того, зависимость от $y$ оказалась просто линейной. Для дисперсии лучшей моделью оказалась функция $f^2(\cdot)$, степенная по $x$ и $y$. Ниже приведены их явные формулы и коэффициенты детерминации $R^2$:

$$\mathcal{M}_z(x_1, x_2, y_1) = 4.254 \times 10^{-7} \times x_1^{-0.790} x_2^{-0.058} \left(y_1 + 9.106 \times 10^6\right) + 2.436, R^2_{\mathcal{M}} = 0.998,$$

$$\mathcal{D}_z(x_1, x_2, y_1) = 40.792 x_1^{-1.234} x_2^{-0.211} y_1^{-0.408} + 0.002, R^2_{\mathcal{D}} = 0.227.$$

Как и для идентификации параметров модели трудоемкости задания 1, оптимизационные задачи (2.4) и (2.5) были решены без ограничений на оптимизируемые параметры. Это было сделано намеренно для проверки свойств функций $\mathcal{M}_z(\cdot)$ и $\mathcal{D}_z(\cdot)$ по переменным $x$ и $y$.

Визуально качество оценивания можно оценить по рис. 12 и 13: на них точками представлены измерения, полученные в результате нагрузочного тестирования, а поверхностями – построенные на их основе модели.

Анализируя рис. 12 и 13, а также числовые характеристики идентифицированных моделей, можно сделать следующие заключения. Во-первых, коэффициент детерминации модели сред-

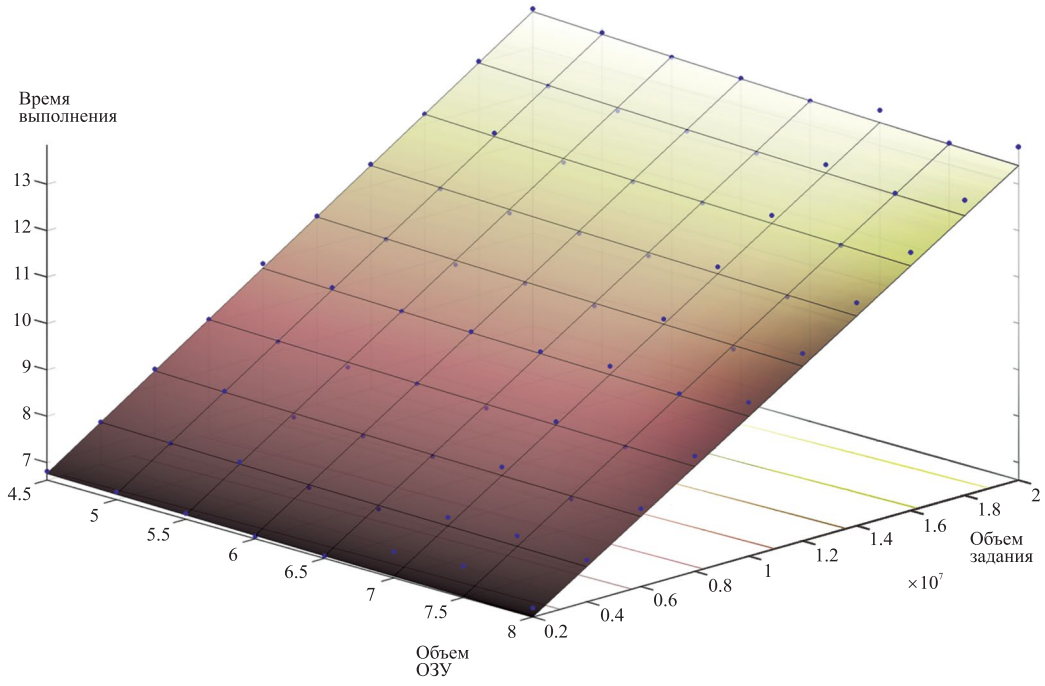


**Рис. 12.** Сечение модели $\mathcal{M}_z(\cdot)$ для случая 1 ядра

него времени очень высок: $R^2_{\mathcal{M}} = 0.998$, в то время как для дисперсии $R^2_{\mathcal{D}} = 0.227$. Это связано с тем, что модель $\mathcal{D}_z(\cdot)$ достаточно близка к константе. Во-вторых, рассмотрим в качестве показателя неопределенности среднее

$$\frac{1}{R}\sum_{r=1}^{R}\frac{\sqrt{\tilde{D}_z^r}}{\tilde{\mathcal{M}}_z^r}\times 100\,\%.$$

Эта величина показывает, какой процент составляет среднее квадратическое отклонение по отношению к среднему значению. Для задания 1 эта величина составляет $4.8\,\%$, в то время как для задания 2 она равна $2.4\,\%$, т.е. вдвое ниже. Коэффициент детерминации – лишь одна из возможных количественных характеристик качества модели, показывающая, насколько точно модель отображает изменчивость данных. Этот коэффициент может быть мал и в том случае, когда сами данные изменяются незначительно, что имеет место в данном случае.

Дальнейший анализ модели трудоемкости выполнялся по двумерным сечениям модели. Графически зависимость среднего выполнения $\mathcal{M}_z(\cdot, y_1)$ и дисперсии $\mathcal{D}_z(\cdot, y_1)$ от объема выполняемого задания $y_1$ при некоторых фиксированных значениях ресурсов $(1, x_2)$ представлена на рис. 14 и 15, т.е. для случая 1 процессорного ядра. На рисунках ломаными с маркерами показаны результаты нагрузочного тестирования, кривыми – построенная модель.

Во-первых, как и в случае задания 1 функции $\mathcal{M}_z(\cdot)$ и $\mathcal{D}_z(\cdot)$, задания № 2 на исследуемой области обладают свойством неотрицательности. Во-вторых, модель $\mathcal{M}_z(\cdot)$ также удовлетворяет свойствам 3 и 5 из [1]: неубыванию и выпуклости по переменной $Y$: для данного типа задания зависимость по $y$ оказывается линейной. В-третьих, получен неожиданный эмпирический результат: для задания 2 дисперсия времени выполнения с ростом объема задания несколько убывает.

На рис. 16 и 17 представлены зависимости среднего и дисперсии времени выполнения задания в зависимости от объема ОЗУ для различных фиксированных объемов задания при использовании 1 или 4 процессорных ядер. На рисунках ломаными с маркерами показаны результаты нагрузочного тестирования, кривыми – построенная модель.

Анализ данных рисунков ведет к следующим заключениям. Начиная с объема ОЗУ 4.5 Гб среднее время выполнения задания и его дисперсия не изменяются с ростом объема ОЗУ. При

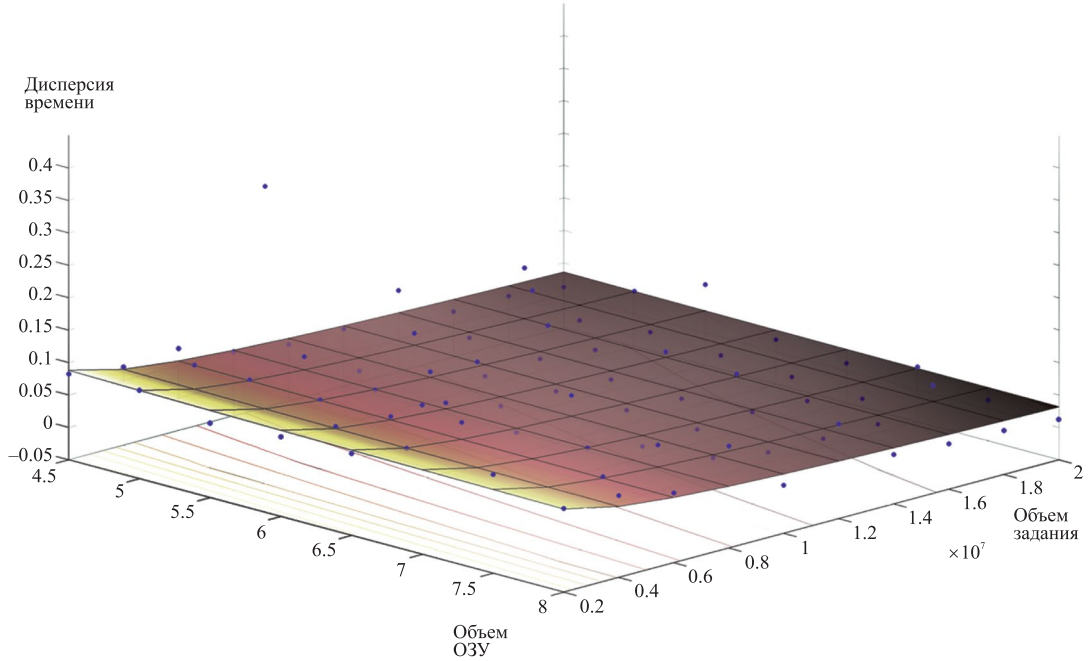


**Рис. 13.** Сечение модели $\mathcal{D}_z(\cdot)$ для случая 1 ядра

этом рост числа используемых процессорных ядер уменьшает как среднее времени выполнения задания, так и его дисперсию. Объяснение подобному явлению вполне очевидно: для эффективного вычисления выборочной ковариации требуется такой объем ОЗУ, чтобы все обрабатываемые данные помещались в кэш и не возникало необходимости чтения данных с диска;

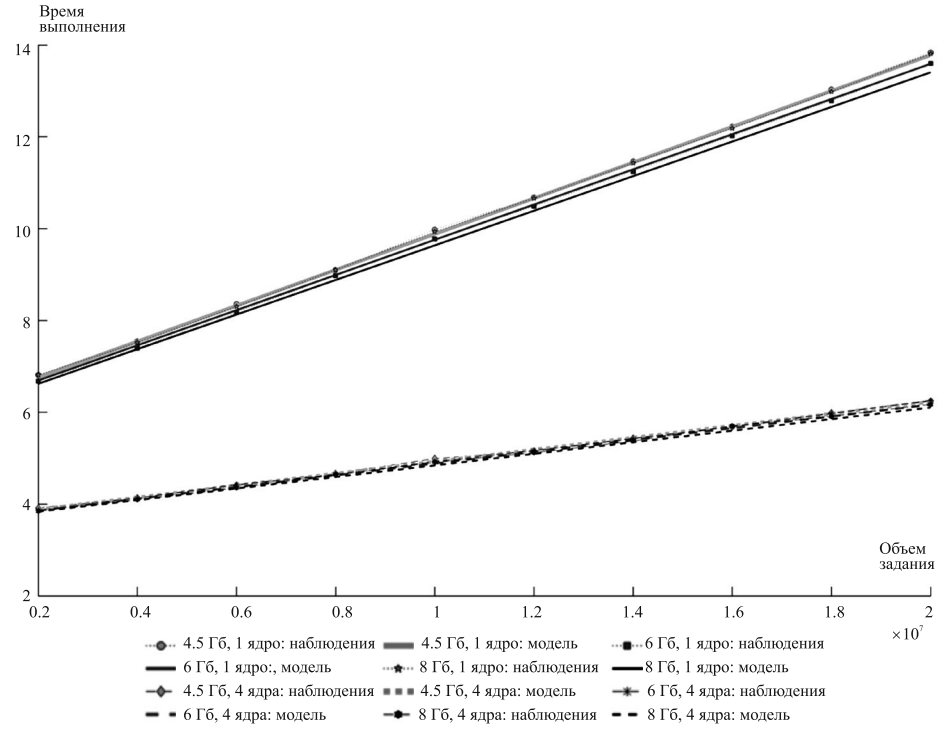


**Рис. 14.** Зависимость $\mathcal{M}_z(\cdot)$ от объема задания для различных фиксированных объемов ОЗУ и числа ядер

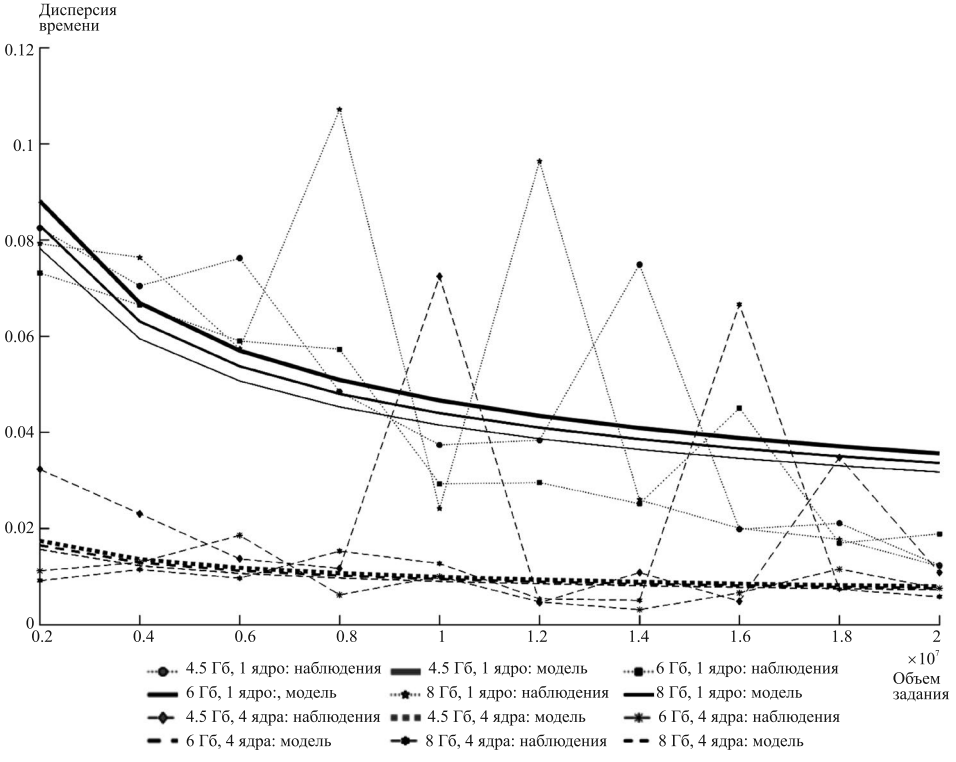


**Рис. 15.** Зависимость $\mathcal{D}_z(\cdot)$ от объема задания для различных фиксированных объемов ОЗУ и числа ядер

дальнейшее наращивание объема ОЗУ не ведет ни к уменьшению времени выполнения этого задания, ни к повышению устойчивости, выражающемуся в снижении дисперсии времени.

На рис. 18 и 19 представлена зависимость среднего времени и дисперсии выполнения задания от числа ядер для различных фиксированных объемов ОЗУ и размеров задания. На рисунках ломаными с маркерами показаны результаты нагрузочного тестирования, кривыми – построенная модель.

Примечательно, что число процессорных ядер не влияло на характеристики трудоемкости выполнения задания 1. Это означает, что задание 1 не допускает распараллеливания. Для задания 2, напротив, увеличение количества используемых ядер значительно снижает как среднее

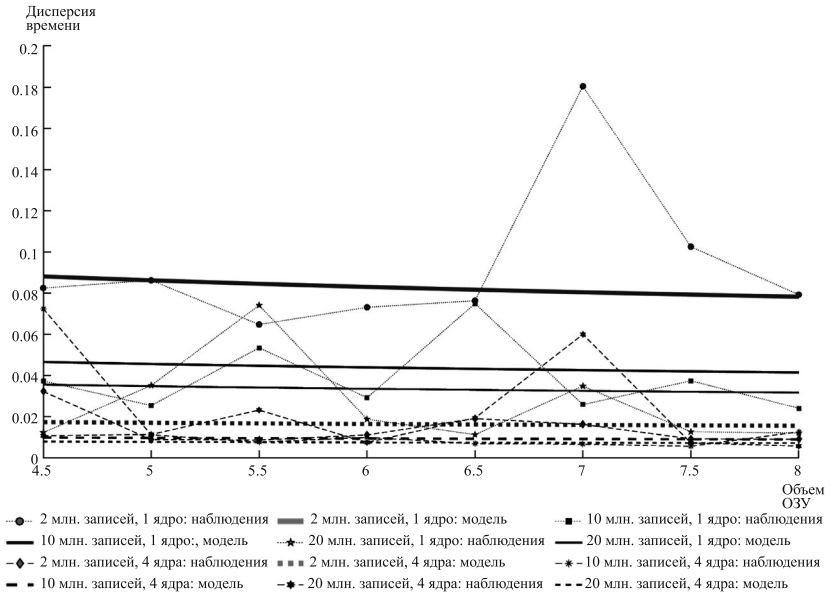


**Рис. 16.** Зависимость $\mathcal{M}_z(\cdot)$ от объема ОЗУ для различных фиксированных объемов задания и числа ядер

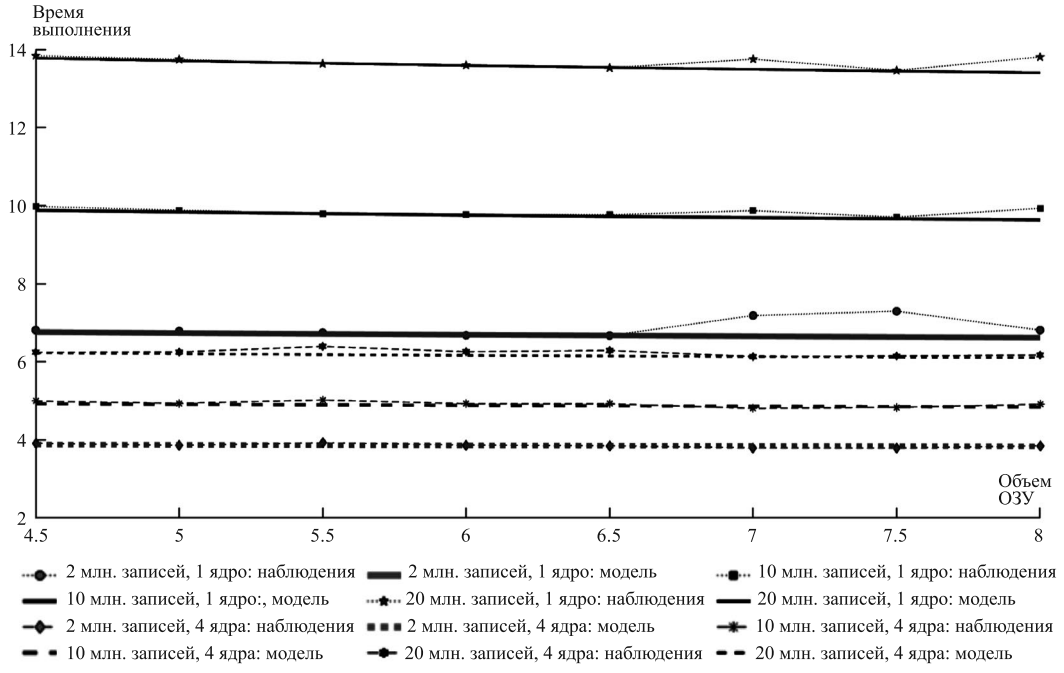


**Рис. 17.** Зависимость $\mathcal{D}_z(\cdot)$ от объема ОЗУ для различных фиксированных объемов задания и числа ядер

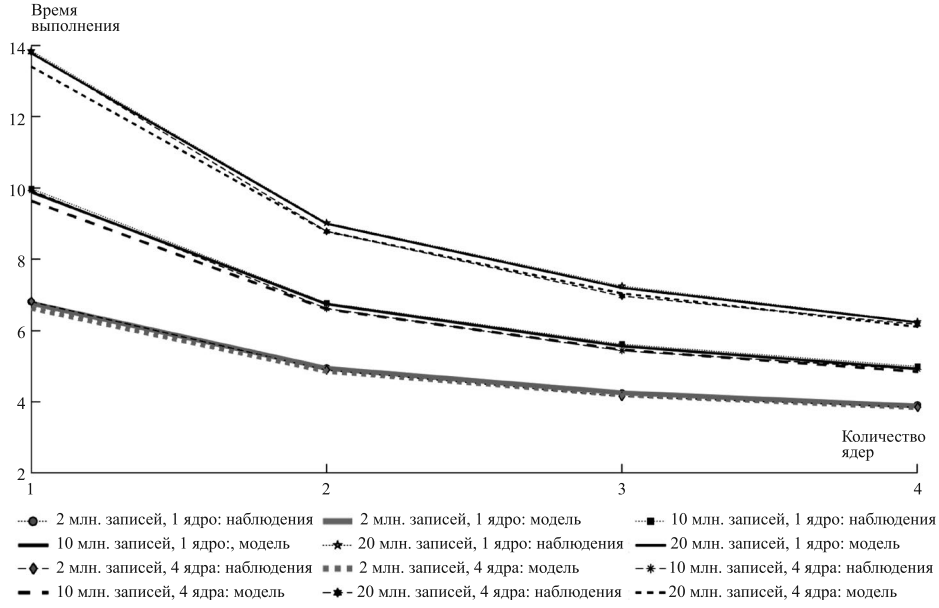


**Рис. 18.** Зависимость $\mathcal{M}_z(\cdot)$ от числа ядер для различных фиксированных объемов задания и ОЗУ

время выполнения задания, так и его дисперсию. Данный факт вполне объясним, потому что процедура вычисления выборочной ковариации содержит в себе ряд операций, допускающих практически полное распараллеливание. Не зная кода данной процедуры, можно предположить, что на первом шаге выполняется выбор данных, подлежащих последующему осреднению. Эти операция, по-видимому, не может быть распараллелена эффективно. Второй шаг, собственно, заключается в вычислении выборочной ковариации, и эта операция допускает практически “полное” распараллеливание: для вычисления выборочной ковариации по выборке объемом $N$ можно задействовать от 1 до 16 процессорных ядер. На третьем шаге происходит окончательное вычисление ответа, когда процесс – сборщик собирает и совместно обрабатывает результаты рабочих процессов, задействованных в распараллеливании. Бóльшее число ядер позволяет запустить бóльшее число рабочих процессов одновременно. Так как не все операции доступны распараллеливанию, среднее время выполнения задания уменьшается не кратно числу используемых ядер, а медленнее.

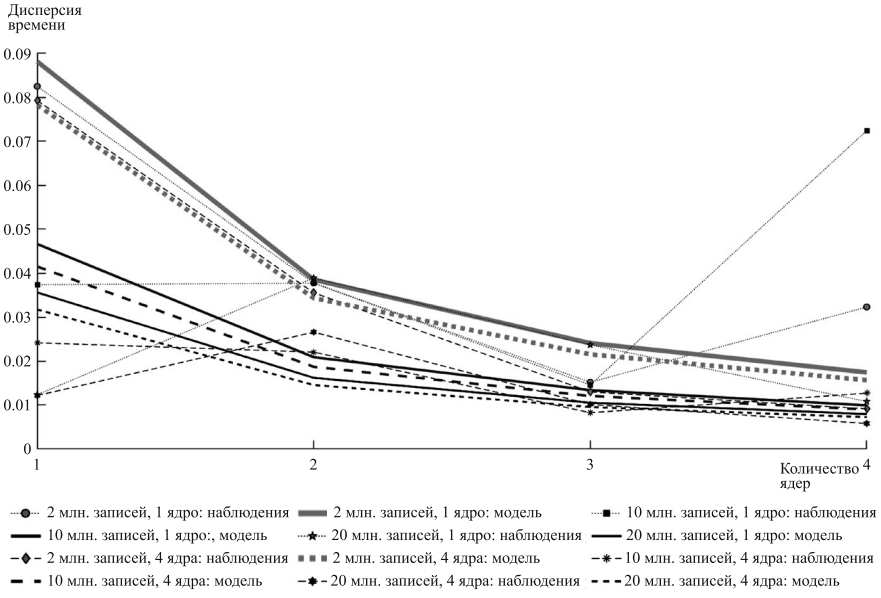


**Рис. 19.** Зависимость $\mathcal{D}_z(\cdot)$ от числа ядер для различных фиксированных объемов задания и ОЗУ

**Заключение.** Вторая часть предложенного исследования носит прикладной характер и иллюстрирует возможность построения стохастической модели трудоемкости пользовательских задач на виртуальном узле на примере двух различных процедур обработки информации, хранимой в некоторой базе данных. В качестве тестируемой была выбрана макетная БД персональных данных пассажиров с инструментарием их обезличивания и статистического анализа. Целью построения моделей двух из пяти возможных типов пользовательских заданий является проверка адекватности математического описания трудоемкости заданий, реализуемых этим макетом, а также подтверждение

возможности постановки данной задачи именно в том смысле, в котором она была сформулирована в [1],

возможности проведения нагрузочного тестирования, необходимого для последующей идентификации параметров предложенных моделей,

основных теоретических предположений о виде зависимости моментных характеристик $\mathcal{M}$ и $\mathcal{D}$ от ресурсов узла и характеристик выполняемых заданий,

приемлемого качества полученных оценок.

Проведенная проверка всех вышеперечисленных пунктов, касающихся моделей трудоемкости пользовательских заданий, связанных с обработкой информации, хранящейся в БД, дала положительный результат.

На следующих этапах исследования предполагается продемонстрировать возможности использования предложенной модели для описания других типов пользовательских заданий: обработки изображений, сжатия данных, выполнения научных вычислений, обучения нейронных сетей и пр.

Детальный анализ моделей трудоемкости заданий 1 и 2 представлен в двух предыдущих разделах. В заключение необходимо отметить следующие факты. Во-первых, минимальные требования к ресурсам узла, обеспечивающим его эффективную работу, различаются для заданий. Для задания 1 достаточным является объем ОЗУ 4 Гб, в то время как для задания 2 – 4.5 Гб.

Во-вторых, результаты нагрузочного тестирования подтверждают тот факт, что минимальные и рекомендуемые аппаратные требования [5] не вполне соответствуют реальным условиям, обеспечивающим эффективное выполнение заданий на вычислительном узле. Так, минимальный объем ОЗУ, требуемый для нормального функционирования операционной системы Астра Linux Орел, составляет 1.5 Гб, а рекомендованный – 4 Гб [5]. В то же время для обработки заданий 1 объемом до 50 000 записей за приемлемое время достаточно объема ОЗУ 1 Гб, а для обработки заданий 2 объем ОЗУ должен составлять не менее 4.5 Гб. Это означает, что характеристики минимальной и рекомендованной конфигураций узла, представляемые разработчиками ОПО, являются весьма приблизительными и должны уточняться в процессе нагрузочного тестирования.

В-третьих, в процессе проведения нагрузочного тестирования задания 2 получен неожиданный результат: с ростом объема задания среднее время его выполнения растет, а дисперсия – слабо убывает.

В-четвертых, согласно рис. 11, зависимости среднего времени выполнения задания от некоторых компонентов вектора $X$, определяющих используемые ресурсы узла, могут быть разрывными. Для компоненты $x_1$ – числа процессорных ядер – этот вывод вполне естественен, т.к. эта переменная по своему смыслу принадлежит некоторому подмножеству натуральных чисел. Но компонента $x_2$ – объем используемой ОЗУ – на виртуальном узле может настраиваться с достаточно малой дискретностью (1 Мб). В то же время на правой части рис. 11 области эффективного и неэффективного функционирования образуют две “полки”, разность высот между которыми визуально напоминает разрыв.

В-пятых, можно указать два признака неэффективной работы узла при выполнении того или иного типа задания. Первым признаком является значительный рост среднего времени. Второй признак – значительный рост дисперсии. В данной части исследования были получены только результаты, когда рост дисперсии в областях неэффективного функционирования узла всегда сопровождался ростом среднего времени. Тем не менее, можно предположить гипотетическую ситуацию, когда среднее время выполнения задания будет относительно невысоким, в то время как дисперсия возрастет значимо. Это означает нестабильное функционирование узла: различные части ПО начинают конкурировать между собой за недостающие ресурсы. Более того, неэффективное функционирование одного виртуального узла может влиять на функционирование другого, размещенного на тех же аппаратных ресурсах. Например, если недостаточный объем ОЗУ первого виртуального узла приводит к активному сво-

пингу, то эта ситуация будет негативно влиять на производительность второго узла, делящего с первым жесткий диск.

Заметим, что перспектива исследований не исчерпывается построением моделей трудоемкости заданий различных типов. Во-первых, в данной работе для описания $\mathcal{M}(\cdot)$ и $\mathcal{D}(\cdot)$ были предложены функции $f^1(\cdot) - f^3(\cdot)$, обладающие свойством монотонного неубывания по переменным $X$. Как показали результаты нагрузочного тестирования, это убывание лишь локальное, и в некоторых областях $\mathcal{M}(\cdot)$ может расти с ростом объема используемых ресурсов $X$. Такую ситуацию можно наблюдать, например, на рис. 3 при обработке задания 1 объемом 350 000: рост объема ОЗУ сначала ведет к падению среднего времени обработки задания, но затем это время начинает слабо расти. Необходимо расширить класс функций $f^1(\cdot) - f^3(\cdot)$ так, чтобы иметь возможность описывать обнаруженное явление. Во-вторых, относительно функции дисперсии $\mathcal{D}(\cdot)$ в [1] сделано лишь одно очевидное предположение неотрицательности. Для описания $\mathcal{D}(\cdot)$ применяются те же функции $f^1(\cdot) - f^3(\cdot)$, что и для описания функции среднего времени $\mathcal{M}(\cdot)$. Этот выбор обоснован лишь тем, что класс $f^1(\cdot) - f^3(\cdot)$ достаточно широк. Вообще, относительно ожидаемых свойств функции дисперсии $\mathcal{D}(\cdot)$ имеется мало информации. Для повышения качества моделей необходимо определить какие-то дополнительные свойства, присущие $\mathcal{D}(\cdot)$ помимо неотрицательности, а также расширить соответствующим образом класс функций $f^1(\cdot) - f^3(\cdot)$, чтобы более точно описывать дисперсию. В-третьих, при построении моделей трудоемкости заданий 1 и 2 при обработке никак не использовались дополнительные статистические данные, планируемые для оценивания параметров $Z$ аппаратно-программной среды узла. Этот факт обусловлен тем, что $Z$ предполагалось применять для построения моделей, описываемых функциями $f^3(\cdot)$ с кусочно-линейной зависимостью по переменным $Y$. Параметры $Z$ отвечали за локализацию точек “склейки” линейных областей. В практических примерах, представленных в данной части исследования, обработка дополнительной статистической информации не потребовалась: полученные измерения $\tilde{\mathcal{M}}_r$ и $\tilde{\mathcal{D}}_r$ хорошо приближались гладкими функциями “без изломов”. Обработка дополнительной информации может потребоваться, например, для построения моделей трудоемкости заданий другого типа (перечень возможных вариантов приведен в этом разделе выше), или для моделей трудоемкости заданий 1 и 2, но для всех возможных значений параметров $(X,Y)$, включая и области неэффективного функционирования узла (см. рис. 2 и 11). В этом случае использование кусочно-гладких или даже разрывных функций для сглаживания наблюдений $\tilde{\mathcal{M}}_r$ и $\tilde{\mathcal{D}}_r$ может дать значительный выигрыш.

Перечисленные задачи определяют перспективу дальнейших исследований.

## СПИСОК ЛИТЕРАТУРЫ

1. *Борисов А., Иванов А.* Стохастическое модели трудоемкости вычислительных задач. I. Принципы формирования, сбор статистических данных, задачи идентификации // Изв. РАН. ТиСУ. № 1. С. 22–34.
2. *Борисов А., Босов А., Иванов А.* Применение имитационного компьютерного моделирования к задаче обезличивания персональных данных. Модель и алгоритм обезличивания методом синтеза // Программирование. 2023. № 5. С. 19–34.
3. *Lagarias, J., Reeds J., Wright M., Wright P.* Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions // SIAM Journal of Optimization. 1998. V. 9. Iss. 1. P. 112–147.
4. *Себер Дж.* Линейный регрессионный анализ. М.: Мир, 1980. 456 с.
5. https://wiki.astralinux.ru/kb/minimal-nye-i-rekomenduemye-sistemnye-trebovaniya-dlya-os-187796554.html.