© 2021 г. В.А. ТОКАРЕВА (victoria.tokareva@kit.edu)
(Технологический институт Карлсруэ, Германия)

РАСПИСАНИЯ С ПРИОРИТЕТАМИ ДЛЯ ЗАДАЧ ОНЛАЙН-УПРАВЛЕНИЯ РЕСУРСАМИ В СИСТЕМЕ АГРЕГИРОВАННОГО ДОСТУПА К ДАННЫМ[1]

Исследуется задача управления ресурсами в системе агрегированного доступа к данным на примере облачного хранилища, предназначенного для управления данными научных экспериментов астрофизики частиц средних и малых размеров. На основе подходов теории массового обслуживания и теории расписаний сформулирована и исследована математическая модель рассматриваемых процессов. Выполнено имитационное моделирование для нахождения эвристики, которая минимизирует сумму критериев расписания. Показано, что применение данной вычислительной дисциплины устойчиво показывает лучшие результаты по сравнению с "наивными" подходами по ряду рассматриваемых параметров.

Ключевые слова: онлайн-планирование, управление данными, управление ресурсами, распределенные вычисления, облака данных, открытые исследования, компьютерные расписания, моделирование.

DOI: 10.31857/S000523102111009X

1. Введение

Для современной науки характерен тренд на междисциплинарные исследования и свободный обмен научными данными. Создание инструментов и платформ, агрегирующих доступ исследователей к научным данным, является объектом внимания многочисленных международных проектов, в том числе с участием отечественных специалистов.

В настоящее время хранение и обработка больших объемов разнородных данных вызывает большой интерес со стороны бизнеса и исследователей. Актуальные методы анализа данных, в частности методы машинного и глубокого обучения, зачастую показывают большую точность при обучении на увеличенной выборке. Использование новейших методов анализа позволяет также получать новые результаты по архивным данным, которые не могли быть получены ранее в связи с техническими и методологическими ограничениями. Совместный анализ данных, полученных в различных экспериментах, помогает проверять точность методов, калибровать детекторы, увеличивать точность анализа вновь собранных данных, извлекать принципиально новое

[1] Работа выполнена при финансовой поддержке совместного гранта Российского научного фонда (№ 18-41-06003) и Общества Гельмгольца (№ HRSF-0027).

научное знание, недоступное ранее. Так, например, одним из наиболее молодых и многообещающих направлений на стыке современных астрономии, астрофизики и физики частиц является так называемая многоканальная астрономия (multi-messenger astronomy) — область науки, комплексно изучающая данные об электромагнитном излучении, гравитационных волнах и элементарных частицах, таких как нейтрино и космические лучи высокой энергии, с целью получения сведений о происходящих в космосе процессах [1]. Сходный запрос на междисциплинарные исследования и коллаборацию между научными экспериментами, в частности экспериментами малого и среднего размера, наблюдается и в других областях науки [2–5].

В то же время сводный анализ данных в таких коллаборациях зачастую оказывается технически перегруженным из-за отсутствия гибкой системы доступа к данным, которая бы предлагала пользователю единый интерфейс запросов и инкапсулировала решение технических задач, возникающих в процессе работы с удаленными хранилищами. Для мега-экспериментов (ATLAS [6], CMS [7] и т.д.), в контексте которых чаще всего обсуждается управление данными в науке, можно выделить популярные решения управления данными, такие как GRID [8] и Hadoop [9]. Данные решения стали де-факто стандартами в этой области, но они зачастую оказываются плохо применимыми для агрегации данных экспериментов меньшего размера. На настоящий момент поиском решений в данной области заняты такие проекты, как [2–4, 10–13].

В качестве недостатка перечисленных подходов можно выделить незначительное количество практико-ориентированных математических моделей, подкрепляющих принятые при разработке инженерные решения и предоставляющих возможность аргументированного исследования и оптимизации характеристик рассматриваемой системы.

В данной статье предлагается динамическая модель онлайн обработки пользовательских заявок сервером для случая агрегированных запросов к K хранилищам со стохастическими элементами, являющая развитием предыдущих исследований автора [14, 15]. Описаны вероятностные и функциональные зависимости в системе. Приводятся результаты имитационного моделирования, нацеленного на поиск обслуживающей дисциплины, оптимизирующей общее время пользователя в системе.

Статья имеет следующую структуру: в разделе 2 приводится обзор литературы и вводятся основные понятия; в разделе 3 описываются постановка задачи и используемые методология и методы; в разделе 4 представлен ход исследования; в разделе 5 описываются основные выводы и направления дальнейшей работы.

2. Обзор литературы

Разрабатываемая облачная среда является системой массового обслуживания (СМО). Исследованием характеристик таких систем занимается теория массового обслуживания, в основе которой лежат работы таких за-

служенных математиков, как А.А. Марков [16, 17], А.Н. Колмогоров [18], Е.С. Вентцель [18], Л. Клейнрок [19], Т.Л. Саати [20], А.Я. Хинчин [21] и др.

Классическими моделями, хорошо изученными в литературе, являются модели с одним или несколькими каналами обслуживания [22], имеющими одинаковую производительность и имеющими распределение потоков [22] входящих и исходящих заявок, соответствующее так называемым марковским процессам (обозначаемым символом M). Также относительно хорошо изученными являются сравнительно простые модели с другими (G, GI, E_r) видами распределений потоков событий. Более сложные модели со связанными очередями исследуются в так называемых сетях массового обслуживания [23–25]. Публикации данного направления опираются на классическую статью Дж.Р. Джексона [26], где доказано, что результаты, справедливые для СМО типа $M/M/m/\infty$, можно применить при расчете стационарного (установившегося) режима работы разомкнутой (открытой) сети СМО. Использование указанного математического аппарата в дальнейшем развитии работы, описанной в данной статье, является одной из перспективных задач, которую ставит перед собой автор.

Время пребывания пользователя в системе определяется суммой времени ожидания в очереди и времени непосредственного обслуживания заявки. Оптимизация первого параметра возможна с помощью выбора разумной дисциплины обслуживания заявки и при использовании параллелизации подзадач, выполняемых в рамках обработки заявки [14], что приводит также к сокращению ожидаемого времени обслуживания.

Будем понимать дисциплину обслуживания как некое расписание обработки пользовательских задач. Алгоритмы составления оптимальных расписаний изучаются в рамках теории расписаний и описаны как в классических работах [18, 19, 27], так и в современных работах [28, 29]. Задачи составления расписаний можно разделить по рассматриваемым характеристикам системы: задачи с одним или многими серверами, с распределением ресурсов, задачи online и real time обработки, а также по типу характера распределения операций по обслуживающим серверам (job shop, flow shop, open shop, mixed shop).

Задачи составления онлайн расписаний изучались в [30–32]. Рассматриваемая в данной статье задача построения расписаний для многопроцессорных систем с конфликтами доступа к ресурсу рассматривалась в публикациях Błażewicz и соавт. [33–35], а также в [34, 36]. В [15] автором давались комментарии об условиях применимости результата [33] в достижении цели исследования.

В практических задачах составления онлайн расписаний зачастую используются [29] аппроксимированные алгоритмы, такие как алгоритмы локального поиска, генетические алгоритмы, эвристики декомпозиции, правила приоритетного обслуживания (priority dispatching rules, PDR). Данные подходы позволяют получить решение в заданных временных рамках. Важным свойством для моделирования горизонтально масштабируемых систем является масштабируемость вычислительных алгоритмов. Исследование [37] ука-

зывает на то, что эвристические алгоритмы, такие как эволюционные, некоторые алгоритмы локального поиска (Taboo search, Simulated annealing) могут быть уязвимы с точки зрения масштабируемости, и делает вывод, что наиболее успешно масштабируются эвристики, обладающие линейной вычислительной сложностью, такие как PDR алгоритмы. Такие достоинства этих эвристик, как вычислительная простота и хорошая масштабируемость, приводят к частому применению данных эвристик в практических задачах онлайн обработки событий. Публикации [38, 39] показывают, что в то время как ни одно из правил этой группы не превосходит другую в целом, однако для конкретных моделей исследуемых процессов и критериев оптимальности оказывается возможным выбор искомой дисциплины обслуживания.

3. Общая формулировка проблемы

На рис. 1 показана схема распределенной системы агрегирования и обработки данных GRADLCI. На схеме $S_1, \ldots, S_3$ обозначают удаленные хранилища, которые используются для извлечения агрегированных данных с быстрым поиском событий с использованием базы данных метаданных (MDDB) [12], после чего пользователь может начать обработку выбранных данных в виртуальной среде на сервере приложений.

Далее смоделируем доступ пользователя к серверу приложений и обработку запросов этим сервером. Запросы, отправляемые пользователем на сервер, могут адресоваться одному хранилищу или иметь агрегированный характер и использовать несколько из них. Рассмотрим основные этапы обработки агрегированного пользовательского запроса.

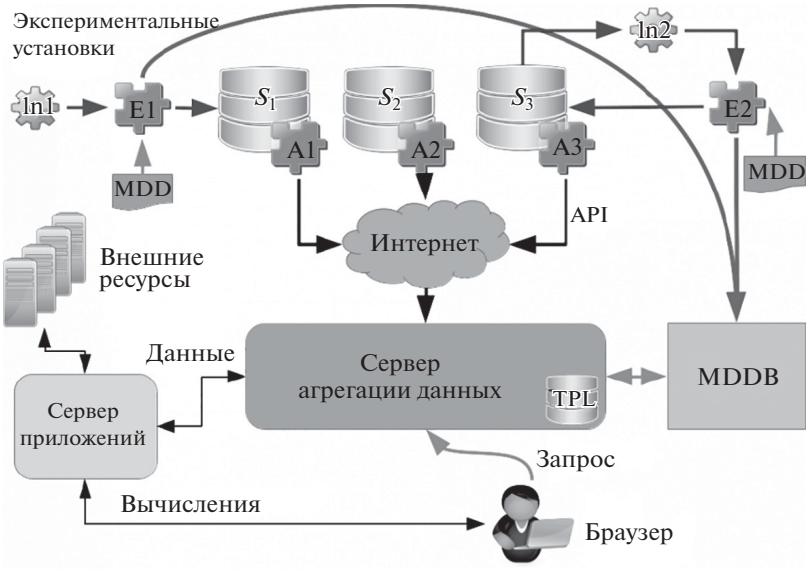

Рис. 1. Архитектура платформы GRADLCI [12].

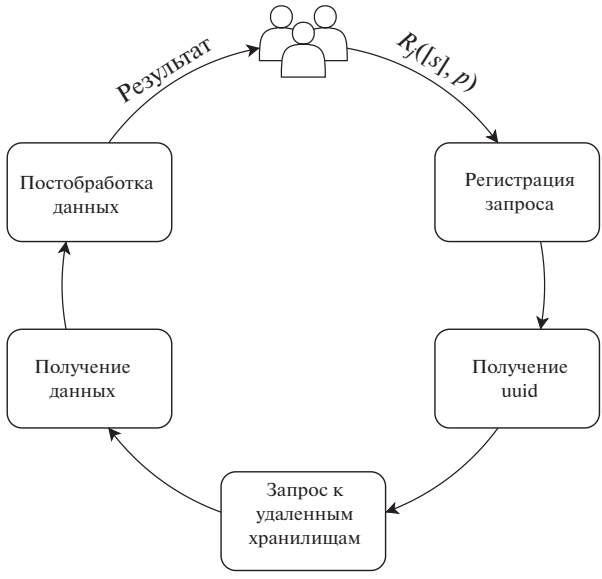

Рис. 2. Цикл обработки агрегированного запроса.

На рис. 2 пользователь отправляет запрос агрегатору и этот запрос ставится в очередь на обработку. Выполнение запроса включает в себя отправку запроса в MDDB для получения уникальных идентификаторов записей (uuids), затем доступ к хранилищам, получение необходимых записей из хранилища, индивидуальную постобработку запроса на стороннем агрегаторе и последующую совместную постобработку. Поскольку сбор данных из разных хранилищ может занимать разное время и влиять на скорость работы системы и эффективность обслуживания клиентов, актуальной проблемой является балансировка запросов в очереди и составление превентивного расписания обработки задач с использованием распределенных ресурсов. Далее построим математическую модель описанных процессов и сформулируем задачу поиска оптимального расписания обработки запросов пользователей.

Пусть $\mathbb{S} = \{S_1, S_2, \ldots, S_K\}$, $K \in \mathbb{N}$ — набор удаленных гетерогенных хранилищ (возобновляемых ресурсов), подключенных к системе агрегации данных. Каждое хранилище представлено одним экземпляром в системе и может быть доступно не более чем одному процессу одновременно. Пусть $[s]$ — подмножество $\mathbb{S}$, а $[k]$ — подмножество соответствующих индексов хранилищ. Пусть p — вектор параметров запроса, принадлежащих множеству $S_1 \bigcup \cdots \bigcup S_K$.

Пусть $R_j([s]_j, p_j)$ — это запрос пользователя к системе, где $j \in [1, \ldots, J]$, $J \in \mathbb{N}$ — порядковый номер запроса к системе, $[s]_j$ — подмножество хранилищ, для которых нужно выполнить агрегированный запрос, а p_j — параметры запроса. Обозначим время поступления запросов через r_j. Для статической модели полагалось $r_j = 0 \ \forall j$.

Предположим, что в системе имеются m идентичных процессоров $P = \{P_1, \ldots, P_m\}$, способных параллельно обрабатывать запросы. Прерыва-

ния в обработке задач запрещены. Предположим, что для каждого запроса $R_j([s]_j, p_j)$ можно назначить вектор $T_j = \{T_{j1}, \ldots, T_{jk}\}$, $k \leqslant m$, приближенных оценок времени T_{jk} пребывания подзапросов к отдельным хранилищам в системе. Для времени пребывания в системе T в литературе также встречается обозначение W_s.

Таким образом, можно записать задачу поиска оптимального расписания обработки запросов в виде

$$\left\langle P_m \mid resK11, T_j \in [\underline{t_j}, \overline{t_j}] \mid \sum_j T_j \right\rangle. \tag{1}$$

Обработка запроса в системе происходит по алгоритму, рассмотренному в [14]. Приведем описание статической модели обработки агрегированного запроса, сформулированной в [15].

В [14] показано, что скорость выполнения отдельных операций линейно зависит от количества записей n_{jk}, извлеченных из удаленных хранилищ, поэтому ключевой момент для оценки общего времени выполнения операций — это процедура оценки количества записей n_{jk}, соответствующих запросу $R_j([s]_j, p_j)$ для каждого k-го хранилища из $[s]_j$. Очевидный подход к этому вычислению — выполнить запрос на подсчет количества записей к MDDB. Однако время выполнения такого запроса может значительно отличаться и иногда замедлять обработку задачи. Для снижения нагрузки на хранилище метаданных и ускорения процедуры оценки предлагается оценивать количество записей по квантилям параметров, хранящихся на сервере агрегации. В этом случае T^c, время оценки n_{jk}, пренебрежимо мало.

Пусть T^w_{jk} будет временем ожидания в очереди j-го подзапроса, использующего ресурс k. Для времени ожидания T^w в литературе также встречается обозначение W_q. Общее время обработки предыдущего подзапроса будет считаться равным $T_{j-1,k}$. Тогда время, необходимое для выполнения этого шага, будет:

$$T^w_{jk} = \begin{cases} T^c, & j = 1, \\ T_{j-1,k}, & j > 1. \end{cases} \tag{2}$$

Эффективное время для инициализации запроса к MDDB будет рассчитываться как:

$$T^i_{jk} = \begin{cases} 0, & t^i_{jk} \leqslant T^p_{j-1,k}, \\ t^i_{jk} - T^p_{j-1,k}, & t^i_{jk} > T^p_{j-1,k}, \end{cases} \tag{3}$$

где $t^i_{jk} \in \mathrm{unif}(0, \Theta_i)$, $\Theta_i \in \mathbb{R}$ — фактическое время инициализации запроса, $T^p_{j-1,k}$ — время обработки $(j-1)$-го запроса.

В [14] приведены следующие зависимости:

$$\begin{cases} t_s(n_{jk}) = \nu \cdot n_{jk}, & \nu \in \mathbb{R}, \\ t_f(n_{jk}) = \mu_k \cdot n_{ik}, & \mu = (\mu_1, \ldots, \mu_k) \in \mathbb{R}^s, \\ t_a(n_{jk}) = \tau \cdot n_{jk}, & \tau \in \mathbb{R}, \end{cases} \tag{4}$$

где ν — скорость обработки MDDB, μ_k — скорость обработки информации для хранилища s_k, τ — скорость постобработки события на стороне агрегатора, а t_s, t_f, t_a являются промежуточными обозначениями для времени операций поиска метаданных, извлечения данных из удаленного хранилища и индивидуальной постобработки соответственно. Эти операции могут выполняться параллельно небольшими частями, и общее время выполнения этих шагов можно рассчитать с помощью уравнения

$$T^p_{jk} = n_{jk} \cdot \max(\nu, \mu_k, \tau). \tag{5}$$

Время совместной постобработки событий агрегатором можно оценить как:

$$T^{pp}_j = \sum_k \xi_k \cdot n_{jk}, \quad \xi \in \mathbb{R}^k. \tag{6}$$

Таким образом, требуемое значение T_j можно рассчитать с помощью уравнения:

$$T_j = \max_k \left(T^w_{jk} + T^i_{jk} + T^p_{jk} \right) + T^{pp}_j. \tag{7}$$

4. Результаты и анализ

Существенным ограничением приведенной модели являлось то, что она является статической и описывает простой случай, когда все заявки считаются поступившими на сервер одновременно в момент времени $t_j = 0 \ \forall j$. Для практических задач характерно динамическое поступление запросов, которое вносит в модель стохастические компоненты. Также вероятностным является выбор пользователем сочетания k из K ресурсов, которое считалось случайным, независимым и равномерно распределенным, т.е. выбиралось сочетание C^k_K с вероятностью использования каждого хранилища q_k. Из каждого хранилища извлекается число событий n_{jk}, удовлетворяющих параметрам запроса p_j. Таким образом, задаче соответствует подмножество используемых хранилищ $[s]_j$ и количеств извлекаемых из каждого хранилища событий $[n]_j$. Ожидаемое число запросов в короткий интервал времени распределено по Пуассону, соответственно интервал времени между запросами определяется экспоненциальным распределением. Однако частота запросов на длинных промежутках времени может изменяться. В данной статье при моделировании был принят вариант, когда частота меняется от времени суток по синусоиде

$$r_j = r_{\min} + \frac{1 - \cos(2\pi t/\tau)}{2}(r_{\max} - r_{\min})$$

с минимумом ночью и максимумом днем по внутреннему времени модели. Здесь τ — период колебаний, $t \in \mathbb{R}$ — текущее время.

Поскольку задача состоит в обеспечении онлайн обработки запросов, вероятностные характеристики были внесены в модель в упрощенном виде. Более

точное исследование вероятностных характеристик системы является одним из направлений дальнейших исследований автора.

Далее, была поставлена задача поиска PDR-эвристики, которая бы минимизировала сумму критериев оптимальности расписания. Были рассмотрены следующие PDR:

- Обслуживание в порядке поступления требований — First In First Out — FIFO;
- Обслуживание в порядке, обратном порядку поступления требований — Last In First Out — LIFO;
- Кратчайшее время обработки — Shortest Processing Time — SPT;
- Наибольшее время обработки — Longest Processing Time — LPT;
- Кратчайшее полное время обработки — Shortest Total Processing Time — STPT;
- Наибольшее полное время обработки — Longest Total Processing Time — LTPT;

и критерии оптимальности расписания:

- Время ожидания (Waiting Time, WT) задачи j в очереди i;
- Время обработки (Processing Time, PT): время, требуемое для обработки задачи j с использованием ресурса i;
- Полное время обработки (Total Processing Time, TPT): полное время пребывания задачи j в системе от момента ее поступления до полного окончания обработки (для всех очередей и ресурсов).

Выбор критериев оптимальности производился из критериев, представленных в [37–39], и обусловлен их применимостью в рассматриваемой задаче.

Было произведено математическое моделирование для каждого из PDR по следующему алгоритму. Число хранилищ при моделировании варьировалось от двух до 10. Задачи, генерируемые при моделировании, обращались к каждому из хранилищ с вероятностью $q_k = 50\,\%$ и затребовали из хранилища число событий, распределенное равновероятно от нуля до максимального количества. Использовавшиеся в моделировании хранилища эмулировали хранилище эксперимента KASCADE [40], способное обработать $r_{\max} = 0{,}75$ запросов в час при условии равномерного распределения числа событий. Было промоделировано поведение системы за месяц для различных средних частот запросов r_{ave} от $0{,}1r_{\max}$ до $0{,}9r_{\max}$. Интервалы между событиями считались случайно распределенными в соответствии с экспоненциальным распределением, параметр которого варьировался в зависимости от времени системы от $0{,}2r_{\text{ave}}$ до $1{,}8r_{\text{ave}}$ в период наименьшей и наибольшей загруженности соответственно.

Моделировалась работа системы в течение условного месяца (30 моделируемых суток). Значения критериев были усреднены по 100 запускам для каждого сочетания моделируемых параметров. Новые задачи генерировались в случайные моменты времени, распределенные экспоненциально с параметром распределения r, варьирующимся в течение модельных суток. Обслуживание заявок из очереди производилось в соответствии с выбранными приори-

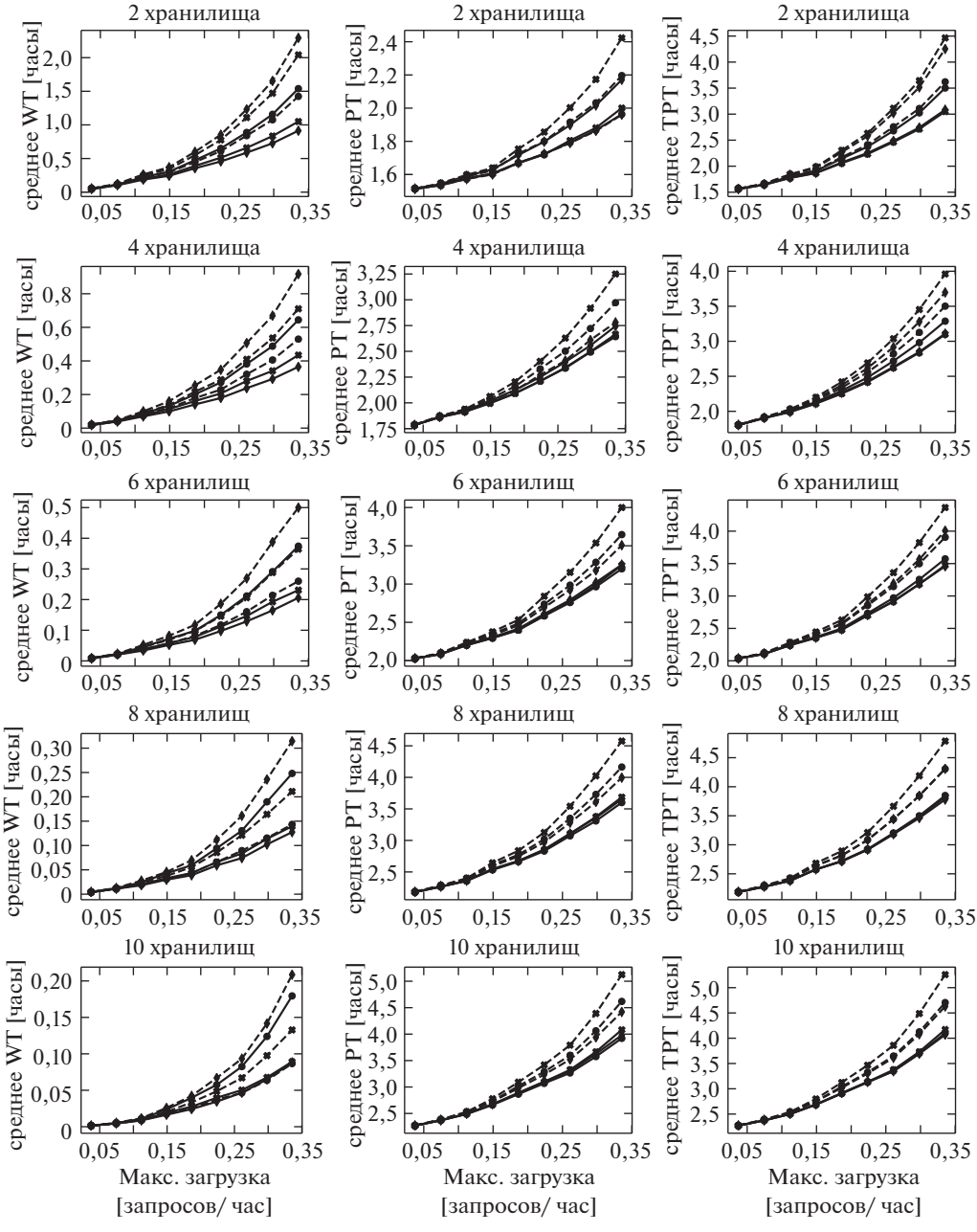

Рис. 3. Результаты моделирования времен-критериев оптимальности расписания для различных правил определения приоритета, количества хранилищ и средних частот запросов. Для маркировки дисциплин обслуживания использованы следующие обозначения. Сплошная линия: круглый маркер — FIFO, крест — SPT, ромб — STPT; штриховая линия: круглый маркер — LIFO, крест — LPT, ромб — LTPT.

тетами. Время обслуживания оценивалось согласно приведенным ранее критериям. Результаты моделирования приведены на рис. 3.

На графиках, приведенных на рис. 3, показаны зависимости значений критериев оптимальности от интенсивности входящего потока заявок для различных PDR. Наименьшее увеличение значения критерия с ростом загруженности системы является показателем удачности выбора PDR.

Можно заметить, что при использовании одного и того же критерия оптимальности увеличение количества хранилищ, подключенных к агрегатору, оказывает влияние на то, какая дисциплина обслуживания оказывается более выгодной. Так, например, в случае двух хранилищ по показателю PT дисциплина LTPT показывает наилучший результат, тогда как с увеличением количества хранилищ данная дисциплина начинает "проигрывать" таким дисциплинам, как STPT и SPT по рассматриваемому критерию.

Выбор критерия оптимальности также влияет на то, какая дисциплина окажется наиболее выгодной для распределения задач в системе. Так, в случае двух хранилищ можно заметить, что дисциплина LTPT оказывается самой выгодной по критерию PT, тогда как по критериям WT и TPT эта стратегия оказывается одной из самых неудачных. Таким образом, конфигурация системы массового обслуживания и выбор критериев оптимизации обслуживания являются существенным фактором, который следует учитывать при выборе используемого PDR.

Вместе с тем для рассматриваемой в рамках данной статьи СМО в случаях, представленных на рис. 3, можно изучить, насколько часто та или иная дисциплина обслуживания оказывается "лучшей" или "худшей" по рассматриваемым критериям. Так, линии, соответствующие дисциплинам LPT и LTPT, оказываются в верхней части графика (среди худших результатов) более чем в 2/3 случаев, что позволяет сделать вывод о том, что применение данных дисциплин является невыгодным в рассмотренных условиях. В то же время дисциплина STPT оказалась среди "лучших" в 0,93% случаев (для некоторых из этих случаев близкий к ней результат показывают FIFO и SPT), что может служить рекомендацией к ее использованию для решения данных классов задач. Качественно данный результат можно объяснить тем, что события, которые стоят в начале очереди, вносят больший вклад в сумму при последующих расчетах.

Результаты данного анализа могут быть использованы при разработке оригинальных планировщиков задач в информационно-вычислительных системах (ИВС) рассмотренной архитектуры и при настройке существующих планировщиков задач в ИВС реального времени.

5. Заключение и выводы

В данной статье была развита динамическая модель онлайн обработки пользовательских заявок сервером для случая агрегированных запросов к K хранилищам за счет добавления стохастической компоненты. Было произведено имитационное моделирование с целью поиска обслуживающей дис-

циплины, позволяющей сократить время обслуживания пользователя в системе.

Развитие данной работы рассматривается в направлении использования математического аппарата сетей массового обслуживания для более точного описания процессов, происходивших в системе. Также автором планируются оценка работы алгоритма на реальных экспериментальных данных и учет неравномерного характера запросов в реальной системе и аппроксимации недостающих измерений для запросов, связанных с определенными сочетаниями запрашиваемых ресурсов.

Автор выражает глубокую признательность коллегам из Научно-исследовательского института ядерной физики им. Д.В. Скобельцына Московского государственного университета им. М.В. Ломоносова (особенно А.П. Крюкову и М.Д. Нгуену), Института астрофизики частиц Технологического института Карлсруэ (в том числе А. Хаунгсу, Д. Вохеле, Ю. Вохеле, Д. Канг и Ф. Полгарту), Иркутского государственного университета и Института динамики систем и теории управления им. В.М. Матросова СО РАН (А.А. Михайлову и А.О. Шигарову) за плодотворную совместную работу в рамках проекта GRADLCI (APPDS) и ценные обсуждения.

СПИСОК ЛИТЕРАТУРЫ

1. *Branchesi M.* Multi-Messenger Astronomy: Gravitational Waves, Neutrinos, Photons, and Cosmic Rays // J. Physics: Conf. Series. 2016. V. 718. P. 022004.
2. *Quix C., Hai R., Vatov I.* GEMMS: A Generic and Extensible Metadata Management System for Data Lakes // CAiSE Forum. 2016. P. 129–136.
3. *Endris K.M., et al.* Ontario: Federated Query Processing Against a Semantic Data Lake / Database and Expert Systems Applications. Eds. Hartmann S., et al. Cham: Springer International Publishing, 2019. P. 379–395.
4. *Villanueva M.J., Valverde F., Levín A.M., Lopez O.P.* Diagen: A Model-Driven Framework for Integrating Bioinformatic Tools // Int. Conf. on Advanced Information Systems Engineering. Heidelberg: Springer, 2011. P. 49–63.
5. *Cohen-Boulakia S., Leser U.* Next Generation Data Integration for Life Sciences // IEEE 27th Int. Conf. on Data Engineering. 2011. P. 1366–1369.
6. *Branco M., et al.* Managing ATLAS Data on a Petabyte-Scale with DQ2 // J. Physics: Conf. Series. 2008. V. 119. P. 062017.
7. *Yzquierdo A.P.-C.* CMS Strategy for HPC Resource Exploitation // EPJ Web of Conferences. 2020. V. 245. P. 09012.
8. *Peters A.J., Janyst L.* Exabyte Scale Storage at CERN // J. Physics: Conf. Series. 2011. V. 331. P. 052015.
9. *Shvachko K., Kuang H., Radia S., Chansler R.* The Hadoop Distributed File System // IEEE 26th Sympos. on Mass Storage Systems and Technologies (MSST). 2010. P. 1–10.
10. *Ayris P., et al.* Realising the European Open Science Cloud. Luxembourg: European Union, 2016.
11. Innovative Digital Technologies for Research on Universe and Matter (ErUM Data IDT). 2020. URL: https://www.erum-data-idt.de/

12. *Bychkov I., et al.* Russian-German Astroparticle Data Life Cycle Initiative // Data. 2018. V. 3. P. 56.
13. *Bolton R., et al.* ESCAPE Prototypes a Data Infrastructure for Open Science // EPJ Web of Conferences. 2020. V. 245. P. 04019.
14. *Tokareva V.* Optimization of Request Processing Times for a Heterogeneous Data Aggregation Platform // J. Physics: Conf. Series. 2021. V. 1740. No. 1. P. 012058.
15. *Tokareva V.* Extended Static Model of User Requests Processing for a Heterogeneous Data Aggregation Platform with K Storages // Proc. of AYSS-2021 Conf. IOP Publishing. 2021. Accepted for publication: Dec. 13 2020.
16. *Чжун К.* Однородные цепи Маркова. М.: Мир, 1964.
17. *Кемени Д., Снелл Д.Л.* Конечные цепи Маркова. М.: Наука, 1970.
18. *Вентцель Е.С.* Исследование операций. М.: Сов. Радио, 1972.
19. *Kleinrock L.* Queuing Systems. N.Y.: Wiley, 1975.
20. *Saaty T.L.* Elements of Queueing Theory, with Applications. N.Y.: McGraw-Hill, 1961.
21. *Хинчин А.Я.* Избранные труды по теории вероятностей. М.: Науч. изд-во ТВП, 1995.
22. *Taha H.A.* Operations Research an Introduction. Pearson Education Limited 2017, 2017.
23. *Gelenbe E., Pujolle G.* Introduction to Queueing Networks. V. 2. N.Y.: Wiley, 1998.
24. *Bramson M.* Stability of Queueing Networks. Heidelberg: Springer, 2008.
25. *Baskett F., Chandy K.M., Muntz R.R., Palacios F.G.* Open, Closed, and Mixed Networks of Queues with Different Classes of Customers // J. the ACM (JACM). 1975. V. 22. No. 2. P. 248–260.
26. *Jackson J.R.* Networks of Waiting Lines // Oper. Res. 1957. V. 5. No. 4. P. 518–521.
27. *Бронштейн О.И., Духовный И.М.* Модели приоритетного обслуживания в информационно-вычислительных системах. М.: Наука, 1976. Т. 2976. С. 221.
28. *Лазарев А.А., Гафаров Е.Р.* Теория расписаний: задачи и алгоритмы. М.: МГУ, 2011.
29. *Błażewicz J., Ecker K., Pesch E., Schmidt G., Sterna M., Weglarz J.* Handbook on Scheduling. Cham: Springer International Publishing, 2019.
30. *Albers S.* Online Scheduling / Introduction to Scheduling. Eds. Robert Y., Vivien F. Boca Raton: Chapman & Hall/CRC Press, 2009. P. 51–78.
31. *Fiat A., Woeginger G.J.* Online Algorithms: The State of the Art. Heidelberg: Springer, 1998. V. 1442.
32. *Pruhs K., Sgall J., Torng E.* Online Scheduling / Hand-Book of Scheduling: Algorithms, Models, and Performance Analysis. Ed. Leung. J.Y.-T. Boca Raton: Chapman & Hall/CRC, 2004. P. 15.1–15.43.
33. *Błażewicz J., Kubiak W., Szwarcfiter J.* Scheduling Independent Fixed-Type Tasks // Advances in Project Scheduling. Elsevier, 1989. P. 225–236.
34. *Błażewicz J., Ecker K.* A Linear Time Algorithm for Restricted Bin Packing and Scheduling Problems // Oper. Res. Lett. 1983. V. 2. No. 2. P. 80–83.
35. *Błażewicz J., Lenstra J.K., Kan A.R.* Scheduling Subject to Resource Constraints: Classification and Complexity // Discrete Appl. Math. 1983. V. 5. No. 1. P. 11–24.
36. *Garey M.R., Johnson D.S.* Complexity Results for Multiprocessor Scheduling under Resource Constraints // SIAM J. on Computing. 1975. V. 4. No. 4. P. 397–411.

37. *Chen B., Matis T.I.* A Flexible Dispatching Rule for Minimizing Tardiness in Job Shop Scheduling // Int. J. Production Economics. 2013. V. 141. No. 1. P. 360–365.

38. *Rajendran C., Holthaus O.* A Comparative Study of Dispatching Rules in Dynamic Flowshops and Jobshops // Eur. J. Oper. Res. 1999. V. 116. No. 1. P. 156–170.

39. *Nguyen S., Zhang M., Johnston M., Tan K.C.* A Computational Study of Representations in Genetic Programming to Evolve Dispatching Rules for the Job Shop Scheduling Problem // IEEE Trans. Evol. Comput. 2013. V. 17. No. 5. P. 621–639.

40. *Wochele J., Wochele D., Haungs A., Kang D.* The KASCADE Cosmic-ray Data Centre KCDC: Releases and Future Perspectives // Proc. 4th Int. Workshop on Data Life Cycle in Physics. 2020. P. 143.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 24.01.2021

После доработки 01.06.2021

Принята к публикации 30.06.2021