

Федеральное государственное бюджетное учреждение
«Национальный исследовательский центр «Курчатовский институт»

На правах рукописи

Ляховец Дмитрий Сергеевич

Методы и средства имитационного моделирования систем управления
заданиями для высокопроизводительных вычислений

2.3.5 – Математическое и программное обеспечение вычислительных систем,
комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертация на соискание ученой степени
кандидата технических наук

Москва – 2025

Работа выполнена в Федеральном государственном бюджетном учреждении «Национальный исследовательский центр «Курчатовский институт».

Научный руководитель: **Баранов Антон Викторович**, кандидат технических наук, доцент.

Официальные оппоненты: **Разумчик Ростислав Валерьевич**, доктор физико-математических наук, доцент, заместитель директора по научной работе Федерального государственного учреждения «Федеральный исследовательский центр «Информатика и управление» Российской академии наук».

Игнатьев Алексей Олегович, кандидат технических наук, начальник лаборатории – заместитель начальника отдела Российского федерального ядерного центра – Всероссийского научно-исследовательского института технической физики им. академика Е. И. Забабахина.

Ведущая организация: Федеральное государственное учреждение «Федеральный исследовательский центр Институт прикладной математики им. М.В. Келдыша Российской академии наук».

Защита состоится 11 декабря 2025 г. в 16:00 на заседании диссертационного совета 24.1.120.01 при Федеральном государственном бюджетном учреждении науки Институте системного программирования им. В. П. Иванникова Российской Академии Наук по адресу: 115035, г. Москва, ул. Садовническая, д. 41, ст. 2.

С диссертацией можно ознакомиться в библиотеке и на сайте Федерального государственного бюджетного учреждения науки Института системного программирования им. В. П. Иванникова Российской академии наук.

Автореферат разослан «___» _____ 2025 г.

Ученый секретарь
диссертационного совета 24.1.120.01,
кандидат физико-математических наук

Турдаков Д.Ю.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Одним из приоритетов научно-технологического развития Российской Федерации определён переход к передовым технологиям проектирования и создания высокотехнологичной продукции, основанным на применении высокопроизводительных вычислительных систем. При проведении фундаментальных и прикладных научных исследований в таких областях, как синтез лекарств, разработка новых материалов с заданными свойствами, прогнозирование месторождений полезных ископаемых, машинное обучение, анализ климатических изменений и других, применение высокопроизводительных вычислительных систем позволяет существенно сократить сроки исследования и ускорить разработку новых опытных образцов.

Современные высокопроизводительные вычислительные системы, часто называемые суперкомпьютерами, строятся на базе большого числа отдельных вычислительных узлов (ВУ), объединённых высокопроизводительной коммуникационной средой. Эксплуатация таких систем осуществляется преимущественно в режиме коллективного пользования. Для получения доступа к суперкомпьютерным ресурсам пользователь оформляет запрос – вычислительное задание, включающее прикладную программу, требования к объёму вычислительных ресурсов, требуемое время её выполнения, входные данные.

Жизненный цикл задания включает в себя постановку в очередь, ожидание в очереди, запуск задания в соответствии с расписанием, инициализацию вычислительных ресурсов для обработки задания, непосредственно обработку задания – выполнение прикладной программы, освобождение вычислительных ресурсов после завершения обработки. Реализацию жизненного цикла заданий осуществляют специальные программные системы управления заданиями (СУЗ), среди которых наибольшее распространение получили такие системы, как Slurm, IBM Spectrum LSF, PBS. В процессе своего развития СУЗ эволюционировали в сложные комплексные системы с множеством конфигурационных настроек, к которым относятся алгоритм планирования заданий и его параметры, схема приоритезации заданий, пользователей и пользовательских групп, различные ограничения в расписании, настройки подсистем квот и резервирования ресурсов и др.

С точки зрения повышения эффективности использования суперкомпьютерных ресурсов, научный интерес представляют исследования влияния параметров СУЗ и

характеристик входного потока заданий, таких как интенсивность, однородность, средний размер задания и др., на показатели качества обработки заданий: загрузку вычислительных ресурсов, время ожидания заданий в очереди, длина очереди и другие. В настоящее время одним из главных инструментов для исследования СУЗ является имитационное моделирование, при проведении которого возникает задача валидации используемой имитационной модели.

Для имитационного моделирования СУЗ применяются такие симуляторы, как Alea, RM-Replay, Slurm Simulator, Batsim, а также программные платформы для построения имитационных моделей, например, GridSim и GPSS. В актуальных научных публикациях отмечаются трудности валидации имитационных моделей, которую авторы либо не проводят, либо рассчитывают интервальные статистические показатели качества, либо применяют методику визуального сравнения графиков изменения показателей качества. В частности, применяемые методы валидации в работах Н.А. Симакова, А. Lucero, М. D'Amico, P.F. Dutot дают только качественное представление о точности имитационной модели СУЗ. Результаты представленного в диссертации анализа выявили отсутствие общепринятых методов оценки точности имитационных моделей при помощи количественного (численного) показателя, что обуславливает актуальность исследований и разработок методов и средств имитационного моделирования с количественной оценкой точности имитационной модели.

Важным этапом жизненного цикла задания является инициализация задания и необходимых вычислительных ресурсов для его обработки. Во время инициализации вычислительные ресурсы не выполняют полезной работы, и с этой точки зрения затраченное на инициализацию время представляет собой накладные расходы. Это время определяется процедурой инициализации, которая для разных заданий может различаться. Задания с одинаковой процедурой инициализации будем называть заданиями одного типа. Определим накладные расходы на обработку задания (НРО) как соотношение времени инициализации и обработки задания. Задания можно разделить на две большие категории – с малыми НРО, когда возможно пренебречь временем инициализации, и с большими НРО, когда время инициализации оказывает существенное влияние на показатель загрузки вычислительных ресурсов, и этим временем пренебречь нельзя.

С развитием гибридных архитектур суперкомпьютеров, расширением сферы применения облачных вычислений число заданий с большими НРО возрастает.

Инициализация таких заданий может включать, помимо выделения подмножества вычислительных узлов и проверки их работоспособности, такие длительные действия, как перепрограммирование ускорителей на базе программируемых логических интегральных схем (ПЛИС), развёртывание виртуальных машин или контейнеров, инициализацию программы обработки задания, обращение в удалённую базу данных для загрузки входных данных и другие.

Одним из способов снижения накладных расходов при обработке заданий с высокими НРО является группировка однотипных заданий в пакеты или мета-задания. Запуск и завершение производится однократно для всего пакета заданий, а обработка пакета осуществляется путём последовательной обработки всех заданий пакета одного за другим. Анализ актуальных исследований по группировке заданий показывает, что рассматриваются, как правило, потоки однотипных заданий. Способы формирования пакетов из однотипных заданий по таймеру и по числу заданий в очереди рассмотрены в работах M. Maheswaran, P. Rosemarry, R. Singh, P. Singhal, D. Sisodia. Метод формирования пакетов из однотипных заданий, основанный на сложности заданий и производительности вычислительных ресурсов, рассматривается в работах N. Muthuvelu, S. Gomathi, D. Manimegalai. Среди работ отечественных учёных в этой области следует отметить исследования В.В. Топоркова, А.И. Костогрызова, А.С. Румянцева.

Проведённый в диссертации анализ научных публикаций и результатов современных исследований в области организации высокопроизводительных вычислений позволяет сделать вывод, что для повышения эффективности использования суперкомпьютерных ресурсов актуальными являются исследования и разработки методов и средств пакетирования однотипных заданий с высокими НРО. В качестве основного инструмента при проведении исследований в этой области целесообразно применить имитационное моделирование.

Цель диссертационной работы состоит в разработке методов и средств моделирования для анализа и улучшения характеристик систем управления заданиями в суперкомпьютерах.

Для достижения цели определены следующие задачи исследования:

- анализ современных средств моделирования систем управления заданиями;
- разработка метода имитационного моделирования систем управления заданиями с возможностью количественной оценки точности имитационной модели;

– разработка и реализация архитектуры программного комплекса имитационного моделирования систем управления заданиями для исследования характеристик системы управления заданиями;

– разработка методики формирования пакетов заданий по типам; экспериментальная оценка её эффективности, определение степени влияния пакетирования на показатели качества обработки заданий с помощью разработанного программного комплекса имитационного моделирования.

Методология исследования. В диссертации использованы методы проектирования программных систем, методы математической статистики, имитационное моделирование. Работа является продолжением исследований и разработок в области организации высокопроизводительных вычислений в режиме коллективного пользования, основы которых заложены в работах таких известных учёных, как В.К. Левин, А.В. Забродин, Г.И. Савин, Вл.В. Воеводин, В.В. Корнеев, Б.М. Шабанов, В.В. Топорков, А.О. Лацис, В.В. Кореньков.

Научная новизна диссертационной работы.

1. Разработан метод имитационного моделирования систем управления заданиями, отличающийся от известных применением количественного показателя близости выходных потоков заданий для сравнительной оценки точности различных имитационных моделей и поддержкой итеративного процесса настройки параметров модели.

2. Разработана архитектура программного комплекса имитационного моделирования систем управления заданиями, обеспечивающая интеграцию с функционирующими СУЗ (включая СУППЗ) без модификации их исходного кода и реализующая механизм обратной связи для адаптивной настройки параметров моделирования в соответствии с характеристиками входного потока заданий.

3. Разработана методика формирования пакетов заданий по типам, отличающаяся от аналогов использованием весов очередей на основе приоритета, времени ожидания и целесообразности формирования пакета, а также обеспечением очереди СУЗ нулевой длины в штатном режиме. Определены границы применимости методики в виде минимальных значений накладных расходов на обработку задания для входных потоков различной интенсивности и однородности.

Теоретическая значимость. Теоретической значимостью для развития имитационных моделей систем управления заданиями обладают предложенный количественный показатель точности моделей систем управления заданиями, метод имитационного моделирования систем управления заданиями, используемый для анализа и улучшения характеристик систем управления заданиями и оценки точности моделей, методика формирования пакетов заданий для обработки потока заданий разных типов.

Практическая значимость. Разработанный метод имитационного моделирования и методика формирования пакетов заданий реализованы в виде программного комплекса [15], который применяется в Отделении суперкомпьютерных систем и параллельных вычислений НИЦ «Курчатовский институт» при выполнении научно-исследовательских работ по программам фундаментальных научных исследований государственных академий наук и в практике отечественного промышленного предприятия. Внедрение комплекса позволило снизить накладные расходы на инициализацию заданий за счет применения методики формирования пакетов. Результаты работы могут быть применены в суперкомпьютерных центрах коллективного пользования для проведения имитационного моделирования систем управления заданиями с целью повышения эффективности использования вычислительных ресурсов.

Положения, выносимые на защиту.

1. Метод имитационного моделирования систем управления заданиями, позволяющий оценивать точность исследуемых моделей при помощи количественного показателя близости выходных потоков заданий и улучшать характеристики систем управления заданиями для заданного входного потока заданий в ходе итеративного процесса настройки параметров модели.

2. Архитектура программного комплекса имитационного моделирования систем управления заданиями, позволяющая проводить динамическую настройку систем управления заданиями в соответствии с характеристиками входного потока заданий за счет реализации механизма обратной связи для адаптивной настройки параметров моделирования и обеспечивающая интеграцию с различными системами управления заданиями без модификации их исходного кода.

3. Методика формирования пакетов заданий по типам, основанная на организации для заданий каждого типа отдельной очереди с определяемыми весами и позволяющая за счет группировки однотипных заданий в пакеты и однократной инициализации заданий пакета повысить полезную загрузку вычислительных ресурсов.

Достоверность и апробация результатов исследования. Достоверность полученных решений подтверждается результатами моделирования, опытом применения в реальных системах, согласованностью с данными, имеющимися в отечественной и зарубежной литературе. Основные результаты работы докладывались и обсуждались на международных, всероссийских и региональных научных конференциях, в том числе:

1. Всероссийская научная конференция «Научный сервис в сети Интернет: поиск новых решений», 17-22 сентября 2012 г., Новороссийск.
2. Всероссийская научная конференция «Научный сервис в сети Интернет: все грани параллелизма», 23-28 сентября 2013 г., Новороссийск.
3. Всероссийская научная конференция «Научный сервис в сети Интернет: многообразие суперкомпьютерных миров», 22-27 сентября 2014 г., Новороссийск.
4. Национальный Суперкомпьютерный Форум (НСКФ-2016), 29 ноября - 02 декабря 2016 г., Переславль-Залесский.
5. 2019 Federated Conference on Computer Science and Information Systems (FedCSIS), Leipzig, Germany, 01-04 сентября 2019.
6. Международная конференция «Суперкомпьютерные дни в России: Труды международной конференции», Москва, Россия, 21–22 сентября 2020 г.
7. 2020 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 6-9 сентября 2020 г.
8. Научный семинар в Межведомственном суперкомпьютерном центре Российской академии наук (МСЦ РАН) – филиале Федерального государственного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», Москва, Россия, 9 марта 2021 г.
9. Международная конференция «Суперкомпьютерные дни в России», Москва, Россия, 26-27 сентября 2022 г.
10. Национальный суперкомпьютерный форум (НСКФ-2024), г. Переславль-Залесский, 26-29 ноября 2024 г.

Публикации. По теме диссертации автором опубликовано 23 печатные работы, из них 11 работ [1-11] изданы в журналах, рекомендованных ВАК, в том числе 4 работы в журналах из перечня ВАК [8-11] и 7 работ в периодических научных журналах, индексируемых Web of Science и Scopus [1-7]. Получено свидетельство о государственной регистрации программы для ЭВМ [15].

В совместных публикациях автору принадлежат аналитический обзор существующих решений по моделированию систем управления заданиями [9], метод имитационного моделирования систем управления и определение показателя близости выходных потоков заданий [5, 12], методика формирования пакетов заданий по типам [1, 2], архитектура программного комплекса имитационного моделирования [6], имитационная модель системы управления заданиями с внешним интерфейсом управления [14] и реализация методики формирования пакетов заданий по типам в имитационной модели Alea [13]. Кроме этого, автору принадлежат результаты экспериментов и имитационного моделирования для исследования эффективности предложенных решений [3, 4, 7, 8, 10, 11].

Личный вклад. Выносимые на защиту результаты получены автором лично.

Структура и объем диссертации. Диссертация состоит из введения, четырёх глав, заключения, списка литературных источников и двух приложений. Каждая глава соответствует отдельному направлению исследования. Общий объем диссертации 125 страниц, в том числе 38 рисунков и 15 таблиц. Список литературы состоит из 122 источников.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении представлена цель работы, обоснована её актуальность, определены понятия и сформулированы основные результаты работы, а также их научная новизна и практическая значимость.

В первой главе на основе результатов анализа предметной области произведена постановка и формулировка основных задач диссертационной работы.

Обзор предметной области начинается с рассмотрения технологического процесса организации высокопроизводительных вычислений в суперкомпьютерных центрах коллективного пользования. Типовая суперкомпьютерная система состоит из вычислительных узлов (ВУ), объединённых высокоскоростной коммуникационной средой. ВУ представляет собой автономный многоядерный сервер, управляемый

отдельным экземпляром операционной системы. ВУ может комплектоваться ускорителями вычислений на базе графических процессоров или программируемых логических интегральных схем.

Для проведения высокопроизводительных вычислений пользователь оформляет запрос в виде задания – информационного объекта, включающего прикладную программу, требования к ресурсам и входные данные. Минимальный состав требований к ресурсам включает число ВУ и время, необходимые для выполнения прикладной программы. Кроме этого, могут предъявляться требования к программной платформе, конфигурации ВУ, наличию ускорителей в составе ВУ и другие. Поток заданий разных пользователей поступает на вход программной системы управления заданиями (СУЗ), которая формирует очередь заданий, составляет расписание (план запуска), выделяет каждому заданию необходимые ВУ на требуемое время, запускает прикладную программу и контролирует её выполнение, освобождает ресурсы после завершения задания.

Выделим в жизненном цикле i -го задания стадию его выполнения длительностью w_i . Эта стадия включает инициализацию ВУ длительностью a_i , последующую инициализацию задания длительностью b_i , обработку задания в виде выполнения прикладной программы длительностью c_i и завершение задания длительностью d_i . Этапы инициализации и завершения задания отнесём к накладным расходам на обработку (НРО), которые для i -го задания определим, как

$$z_i = \frac{a_i + b_i + d_i}{c_i} \times 100\%. \quad (1)$$

Для инициализации и завершения задания СУЗ выполняет специальные программные процедуры. Будем называть задания с одинаковыми процедурами инициализации и завершения заданиями одного типа.

На рисунке 1 представлена типовая модель обработки заданий.

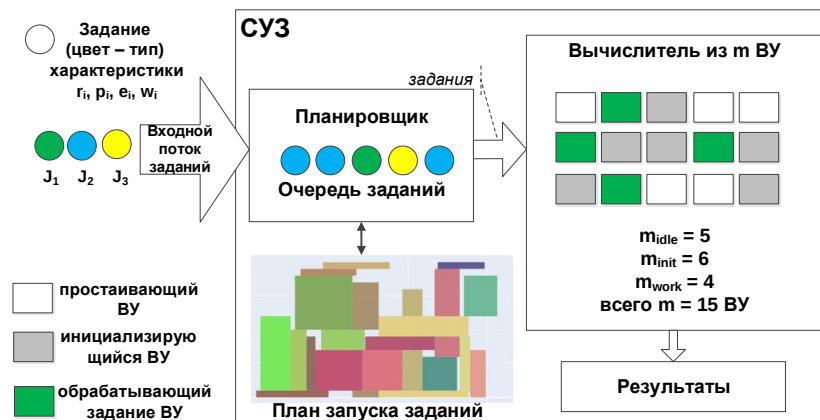


Рисунок 1 – Типовая модель обработки заданий

Поток независимых заданий $J_1, J_2, \dots, J_n$ разных типов поступает на вход суперкомпьютера из m ВУ. Задание $J_i, i \in \overline{1, n}$, поступает в момент времени r_i и требует для своего выполнения p_i ВУ на время w_i .

Важнейшим компонентом СУЗ является планировщик, формирующий план запуска заданий, в котором все задания могут быть представлены прямоугольниками на плоскости, ограниченной осью времени и осью порядковых номеров ВУ. В плане запуска для каждого задания определяется момент его запуска s_i и выделенное ему подмножество из p_i ВУ. План запуска составляется таким образом, чтобы в каждый момент времени выполняющиеся задания были обеспечены ресурсами, то есть должно выполняться условие $\sum_{i=1}^n p_i \varphi_i(t) \leq m$, где $\varphi_i(t) = 1$, если задание i выполняется в момент времени t , и $\varphi_i(t) = 0$ в противном случае.

Каждый исправный ВУ характеризуется состоянием «свободен», «инициализируется для обработки задания типа j », «обрабатывает задание J_i ». Под узло-часом будем понимать единицу измерения объёма вычислительного ресурса, эквивалентную 1 ВУ, находящемуся в определённом состоянии в течение 1 часа. Например, если в течение x часов y ВУ были свободны, то объём свободного вычислительного ресурса будет составлять $x \cdot y$ узло-часов.

Введем показатель полной загрузки L за период времени T :

$$L(T) = \frac{S_{busy}(T)}{S_{all}(T)}, \quad (2)$$

где $S_{busy}(T)$ – суммарный объём вычислительного ресурса в узло-часах, занятого за период T инициализацией или обработкой заданий, $S_{all}(T)$ – суммарный объём вычислительного ресурса в узло-часах, доступного для обработки заданий за период T .

Показатель полной загрузки не учитывает, что во время инициализации не выполняется полезной вычислительной работы, и с этой точки зрения инициализируемые ВУ фактически простаивают. Для учёта этого простоя определим показатель полезной загрузки U за период времени T :

$$U(T) = \frac{S_{work}(T)}{S_{all}(T)}, \quad (3)$$

где $S_{work}(T)$ – суммарный объём вычислительного ресурса в узло-часах, занятого за период T обработкой заданий. Отметим, что всегда выполняется условие $U(T) \leq L(T)$.

Среднее время ожидания задания в очереди с момента поступления задания в систему до начала инициализации для k заданий рассчитывается по формуле:

$$Q = \frac{\sum_{i=0}^k q_i}{k}, \quad (4)$$

где q_i – время ожидания задания в очереди.

Выделим класс заданий с высокими НРО, которыми нельзя пренебречь при оценке качества обработки заданий. Поток заданий с высокими НРО обеспечивает полезную загрузку U , значительно меньшую, чем полная загрузка L . К заданиям с высокими НРО следует отнести задания с длительной инициализацией высокопроизводительного ускорителя; задания, требующие развертывания специализированной программной платформы или загрузки значительного объёма данных; задания с малым временем обработки и другие. Повысить полезную загрузку U возможно за счёт объединения заданий одного типа в пакеты (группы заданий) с однократным выполнением процедуры инициализации для всех заданий пакета.

Приведённый в главе анализ существующих методов и средств формирования пакетов заданий показал, что недостаточно проработаны вопросы обработки заданий разных типов и определения необходимого объёма вычислительных ресурсов для выполнения заданий из сформированного пакета. В известных исследованиях рассматриваются потоки однотипных заданий и решается обратная задача определения числа заданий для включения в пакет и выполнения на выделенных ВУ. Результаты проведённого анализа позволили сделать вывод об актуальности разработки новой методики формирования пакетов заданий, применимой для обработки непрерывно поступающего потока заданий разных типов.

Для исследования влияния методики формирования пакетов заданий и её параметров на показатели качества обработки заданий целесообразно применить имитационное моделирование, для чего могут быть использованы различные симуляторы СУЗ. При этом симулятор СУЗ будет формировать собственный план запуска заданий, отличающийся от плана реальной системы. Оценку точности применяемой в симуляторе СУЗ имитационной модели можно получить путём сравнения плана запуска симулятора с планом запуска реальной системы. Для количественной оценки точности имитационной модели необходима разработка нового метода имитационного моделирования СУЗ. Метод должен быть применим к произвольной имитационной модели СУЗ и поддерживать циклический процесс моделирования, в ходе которого на каждой итерации уточняются параметры используемой модели.

Методика формирования пакетов заданий и метод имитационного моделирования должны быть реализованы в виде программного комплекса имитационного моделирования СУЗ с новой архитектурой, к которой в диссертации предъявлен ряд требований. Среди этих требований следует выделить модульность, возможность адаптации параметров под произвольный входной поток заданий и совместимость с произвольной СУЗ.

Для исследования эффективности и определения границ применимости методики формирования пакетов заданий необходимо при помощи разработанного комплекса имитационного моделирования СУЗ провести экспериментальное исследование на множестве входных потоков заданий с разными характеристиками.

Вторая глава посвящена имитационному моделированию систем управления заданиями. Рассмотрены существующие методы моделирования СУЗ и общая схема имитационного моделирования СУЗ.

На вход имитационной модели СУЗ или реальной системы поступает входной поток заданий с некоторыми характеристиками. Результатом работы модели СУЗ (или реальной системы) является выходной поток заданий, определяемый построенным планом запуска, что показано на рисунке 2.

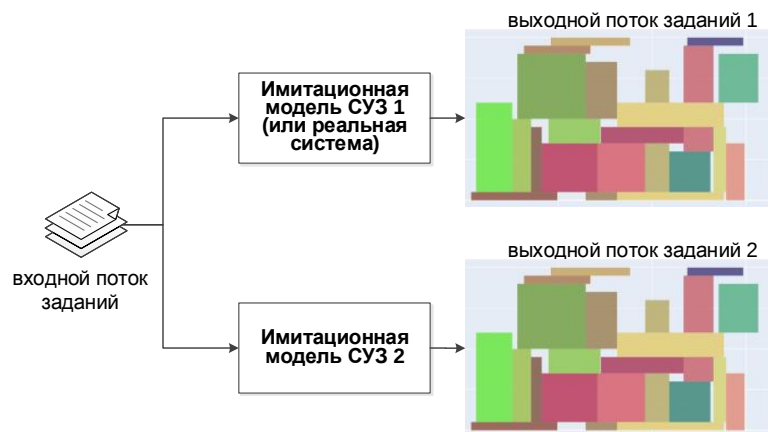


Рисунок 2 – Сравнение точности разных моделей

Под точностью моделирования будем понимать степень схожести выходных потоков заданий имитационной модели и реальной суперкомпьютерной системы одинаковой конфигурации, то есть построенных планов запусков заданий. В рассмотренной литературе можно выделить три способа оценки точности модели СУЗ: сравнение интегральных показателей качества, визуальное сравнение графиков изменения показателей качества, визуальное сравнение планов запуска. Для оценки

разницы между двумя экспериментами при фиксированном входном потоке заданий необходим численный показатель разницы выходных потоков заданий, который позволит сравнить два плана запуска. При совпадении выходных потоков заданий модель и реальная система будут также иметь идентичные интервальные показатели качества: полную и полезную загрузку вычислителя, среднее время ожидания задания в очереди и другие.

Для двух сравниваемых моделей построим два вектора размерности n . Для первой модели определим вектор времен пребывания заданий в системе $v = (v_1, v_2, \dots, v_n)$, $i \in \overline{1, n}$, где каждая компонента соответствует номеру задания в порядке его поступления в систему. Значение компоненты $v_i = s_i - r_i + w_i$ задаётся как время пребывания задания в системе, то есть сумма времени ожидания задания в очереди и времени его обработки. Для второй модели аналогично определим вектор $V = (V_1, V_2, \dots, V_n)$, $V_i = (S_i - R_i + W_i)$, $i \in \overline{1, n}$.

Рассмотрим показатель P близости выходных потоков заданий для двух сравниваемых моделей:

$$P = \sqrt{\frac{\sum_{i=1}^n (V_i - v_i)^2}{n}}. \quad (5)$$

Показатель P представляет собой накопленную суммарную ошибку сдвига времён запуска всех заданий в двух сравниваемых моделях. В диссертации показано, что показатель P не зависит от размера сравниваемых векторов, что позволяет его использовать для любых по длительности модельных экспериментов. Показатель P может быть применён в качестве количественной оценки точности имитационной модели при сравнении её выходного потока заданий с выходным потоком реальной суперкомпьютерной системы.

Далее в главе представлен новый метод имитационного моделирования СУЗ, использующий показатель (5) для оценки точности разных имитационных моделей для заданного входного потока заданий. Предлагаемый метод состоит из следующих этапов.

1. Выбор схемы имитационного моделирования.
2. Выбор цели имитационного моделирования. Целью может быть поиск наилучших настроек СУЗ (таких как алгоритм планирования и его настройки) или исследование точности имитационной модели.

3. Выбор имитационной модели СУЗ для проведения моделирования. В качестве имитационной модели могут быть использованы симуляторы или реальная СУЗ.

4. Получение выходного потока заданий из реальной СУЗ и его преобразование во входной модельный поток заданий, или генерация входного потока заданий на основе выбранных параметров и видов распределения вероятностей характеристик заданий.

5. При необходимости, внесение изменений в настройки модели СУЗ.

6. Проведение имитационного моделирования с входным потоком заданий, получение выходного потока заданий как результата эксперимента.

7. Ручной или автоматизированный анализ выходного потока заданий. В зависимости от выбранной в п. 1 цели рассчитываются интересующие показатели качества обработки потока заданий или количественный показатель близости модели относительно входного потока заданий из реальной системы или из предыдущей итерации эксперимента.

8. Если результаты анализа свидетельствуют о достижении цели (получены подходящие настройки СУЗ с точки зрения показателей качества обработки потока заданий или достигнута требуемая точность моделирования), то процесс моделирования завершается. Если нет, то переход к этапу 2.

На рисунке 3 показан циклический характер исследования при помощи предложенного метода. Многократный эксперимент позволяет исследовать различные вариации входных потоков и параметров СУЗ, т.е. определить влияние множества интересующих факторов на качество обработки потока заданий.

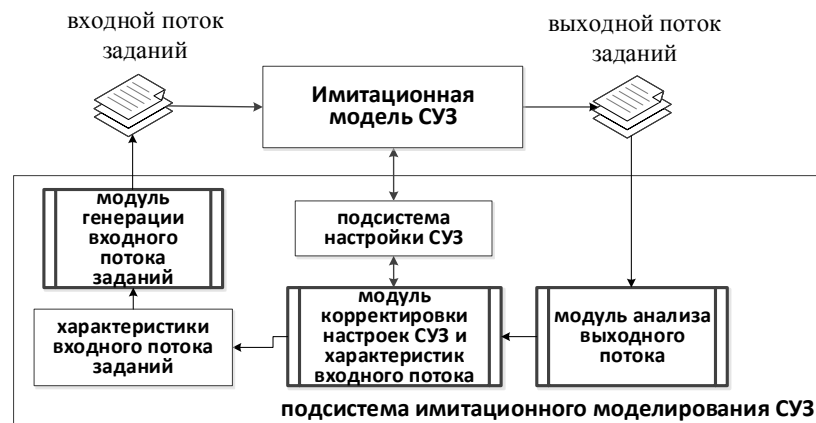


Рисунок 3 – Общая схема имитационного моделирования СУЗ

В современных СУЗ при планировании заданий, как правило, используется стратегия обратного заполнения, позволяющая значительно повысить полную загрузку. В соответствии с этой стратегией менее приоритетные задания могут быть

запущены вне очереди, если это не помешает запуску более приоритетного задания. При этом у каждой СУЗ есть свои особенности применения обратного заполнения, существенно влияющие на итоговый план запуска заданий. В результате один и тот же входной поток заданий на разных СУЗ или моделях СУЗ будет обработан по-разному.

Для подтверждения, что показатель (5) адекватно ранжирует модели по точности, был проведён следующий эксперимент. В качестве входного потока был использован поток заданий за недельный период работы суперкомпьютера МВС-10П ОП (раздел Broadwell), установленного в МСЦ РАН – отделении МСЦ Курчатовского высокопроизводительного вычислительного комплекса НИЦ «Курчатовский институт». На суперкомпьютере применяется отечественная Система управления прохождением параллельных заданий¹ (СУППЗ). Алгоритм планирования СУППЗ основан на принципе обратного заполнения и динамических приоритетах пользователей. Динамический приоритет пользователя влияет на местоположение заданий пользователя в очереди и имеет обратную зависимость от суммарного времени выполнения его заданий за некоторый учетный период. Информация о динамических приоритетах не фиксируется в базе данных подсистемы статистики и журнале запуска заданий, но используется для планирования заданий, эту информацию назовем оперативной информацией (ОИ).

Для экспериментального исследования была сформирована последовательность моделей СУЗ с заведомо разной точностью:

- вариант а) – симулятор СУППЗ с виртуальным вычислителем (ВВ) с учетом оперативной информации, наиболее точная модель и максимально близкий к натурному эксперименту вариант моделирования, при использовании которого вместо реального запуска осуществляется выделение виртуальных ВУ на время обработки задания;
- вариант б) – симулятор СУППЗ с ВВ без учета ОИ, менее точный вариант моделирования, поскольку не учитываются приоритеты пользователей;
- вариант в) – симулятор СУППЗ с ВВ без учета ОИ, в этой модели дополнительно снижена точность за счёт замены во входном потоке заданий заказанного времени выполнения на реальное время выполнения, что оказывает существенное влияние на план запуска заданий;

¹ Savin G.I. et. al. Joint Supercomputer Center of the Russian Academy of Sciences: Present and Future // Lobachevskii J. of Mathematics. 2019. Vol. 40. No. 11. pp. 1853–1862. DOI: 10.1134/S1995080219110271.

– вариант г) – моделирование при помощи симулятора Alea² с алгоритмом обратного заполнения: ещё менее точная модель, поскольку в ней применяется другая реализация алгоритма планирования.

На рисунке 4 и в таблице 1 представлены результаты применения разработанного метода на одном и том же входном потоке заданий для последовательности моделей СУЗ с заведомо разной точностью.

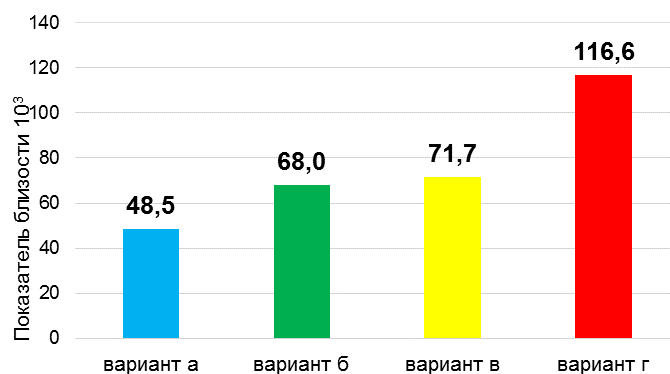


Рисунок 4 – Показатель близости выходных потоков для имитационных моделей с заведомо разной точностью

По некоторым показателям в таблице 1, например, полной загрузке, моделирование без учёта ОИ выглядит наиболее точным. Однако, при этом среднее время ожидания задания в очереди отличается от реальной системы в два раза. Показатель (5) адекватно ранжирует модели по убыванию их точности: априорно менее точная модель демонстрирует большее значение показателя (5).

Таблица 1. Результаты моделирования

Система Показатель	Реальная система	Вариант а)	Вариант б)	Вариант в)	Вариант г)
Полная загрузка в соответствии с (2), %	85,7	87,4	87,7	87,7	84,8
Среднее время ожидания в очереди в соответствии с (4), с	52 178	54 070	40 148	25 017	44 057
Показатель близости в соответствии с (5), 10^3	0	48,5	68,0	71,7	116,6

Для этого же входного потока заданий в диссертации приведены результаты сравнения работы симулятора СУППЗ с ОИ и различных алгоритмов планирования,

² Klusacek D., Soysa M., Suter F. Alea - Complex Job Scheduling Simulator // Parallel Processing and Applied Mathematics. PPAM 2019: Lecture Notes in Computer Science, pp. 217-229 (2020). DOI: 10.1007/978-3-030-43222-5_19.

реализованных в Alea. Показатель (5) позволил выявить существенное различие между алгоритмами планирования СУПЗ и Alea, при этом разные алгоритмы, реализованные в Alea, оказались близки между собой по точности.

Предложенный метод имитационного моделирования систем управления заданиями отличается от известных применением количественного показателя близости выходных потоков заданий и поддержкой итеративного процесса настройки параметров модели, что позволяет как оценивать точность исследуемых моделей, так и улучшать характеристики системы управления заданиями для заданного входного потока. Результаты экспериментов показали, что предложенный показатель (5) адекватно ранжирует априорно разные имитационные модели по точности моделирования.

Третья глава посвящена методике формирования пакетов заданий по типам и архитектуре программного комплекса имитационного моделирования СУЗ.

В главе представлена частная модель системы обработки линейно масштабируемых заданий с поддержкой различных типов заданий. Для каждого типа заданий ведётся отдельная очередь. Для заданий одного типа время инициализации является константой.

В рамках частной модели предложена новая методика формирования пакетов по типам (ФПЗТ), включающая 5 этапов. В диссертации рассмотрены различные варианты реализаций каждого из этапов методики.

Этап 1. Определить момент начала формирования пакета заданий $t_{форм}$.

Этап 2. Среди всех непустых очередей выбрать очередь T_j наибольшего веса $W(T_j)$, где $W(T_j) = \max W(T_i), i \in (1, h)$.

Этап 3. Из выбранной очереди T_j отобрать g заданий для включения в пакет.

Этап 4. Определить число вычислительных узлов $m_{нак}$ для обработки сформированного пакета с учётом показателя масштабирования k , ограничивающего число выделяемых пакету ВУ. Для сформированного пакета из g заданий выделятся ВУ так, чтобы обработка пакета не более, чем в k раз превысила время инициализации. Для этого рассчитывается $m_{цел}$:

$$m_{цел} = \frac{\sum_{i=1}^g e_i}{B_j \times k}, \quad (6)$$

где e_i – время обработки i -го задания на 1 ВУ, B_j – время инициализации заданий типа j . Пакет формируется для выполнения на $m_{нак} = \min(m_{цел}, m_{своб})$ узлах из $m_{своб}$ ВУ, свободных на момент формирования пакета.

Этап 5. Сформировать пакет заданий для обработки на $m_{\text{пак}}$ ВУ. Сформированный пакет подаётся на вход СУЗ.

Методика ФПЗТ основана на идее «жадного» алгоритма, где целью каждого этапа является выбор такого размещения заданий, которое обеспечивает наименьшие накладные расходы на обработку пакетов заданий. Отметим отличительные особенности методики ФПЗТ:

- обеспечение очереди СУЗ нулевой длины в штатном режиме;
- малая вычислительная сложность процедуры выбора очередного пакета $O(h+g)$ для h типов заданий и g заданий в сформированном пакете.

Методика ФПЗТ основана на организации для заданий каждого типа отдельной очереди с определяемыми весами и позволяет за счёт однократной инициализации заданий одного пакета повысить полезную загрузку вычислительных ресурсов. Главным параметром методики является показатель масштабирования k , который определяет объём ресурсов так, чтобы обработка всех заданий пакета не более, чем в k раз превысила время его инициализации.

Для реализации метода имитационного моделирования, рассмотренного в главе 2, и методики ФПЗТ, а также для исследования эффективности и границ применимости методики ФПЗТ разработан программный комплекс имитационного моделирования систем управления заданиями с новой архитектурой, показанной на рисунке 5.

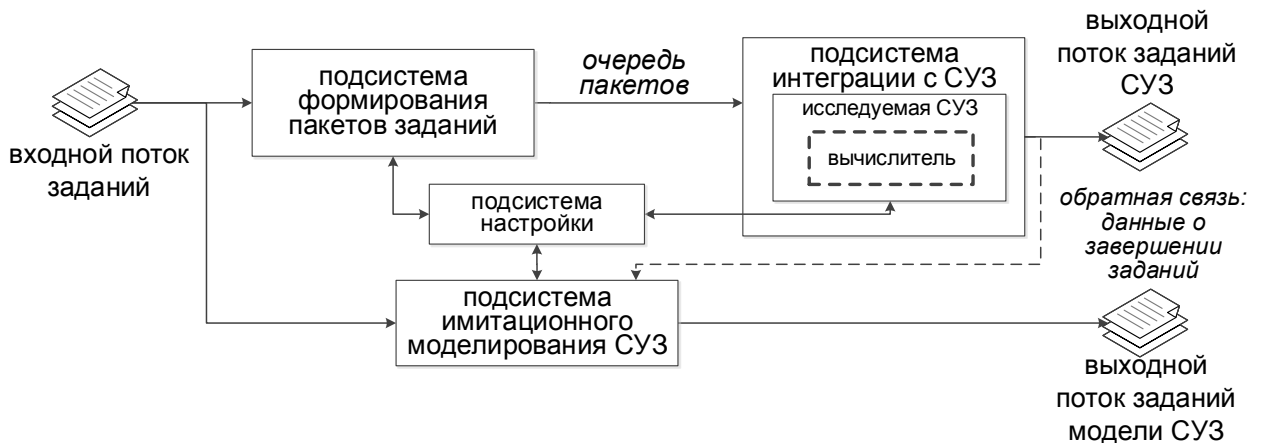


Рисунок 5 – Архитектура программного комплекса имитационного моделирования систем управления заданиями

Архитектура программного комплекса имитационного моделирования систем управления заданиями организована по модульному принципу и включает следующие подсистемы.

1. Подсистема формирования пакетов заданий, реализующая методику ФЗПТ. Каждое задание входного потока распределяется в очередь своего типа, при обработке которой определяется момент $t_{форм}$ формирования пакета, состав пакета и требуемый объём вычислительных ресурсов для его обработки.

2. Подсистема имитационного моделирования реализует исследуемую модель СУЗ. Подсистема отвечает за подбор параметров методики ФПЗТ и других настроек СУЗ, а также за оценку точности исследуемой модели.

3. Подсистема интеграции с СУЗ обеспечивает совместимость с различными СУЗ (Slurm, IBM Spectrum LSF, PBS, СУППЗ и др.) без модификации их исходного кода. Для интеграции требуется реализовать функции постановки в очередь СУЗ очередного пакета заданий и запроса количества свободных вычислительных узлов.

Входной и выходной потоки заданий поступают на вход подсистемы имитационного моделирования, которая может функционировать в двух режимах.

Первый режим используется для оценки точности имитационной модели. В этом режиме производится имитационное моделирование входного потока заданий и получение выходного потока заданий модели, который сравнивается с выходным потоком заданий реальной системы. При снижении точности имитационного моделирования целесообразно ставить вопрос о доработке модели.

Второй режим используется для настройки параметров реальной СУЗ. Системным администратором СУЗ проводится серия экспериментов для фиксированного входного потока заданий с заданным показателем, например, показателем масштабирования в методике ФПЗТ. Если в результате анализа выходных потоков экспериментов будут достигнуты подходящие значения показателей качества обработки потока заданий, настройки переносятся в реальную СУЗ. При необходимости циклически повторяется процедура настройки.

В этом режиме за счёт обратной связи между исследуемой СУЗ и подсистемой имитационного моделирования создается замкнутый итеративный оптимизационный цикл «моделирование → оценка → коррекция», который обеспечивает устойчивость к изменениям характеристик входного потока заданий и реализует свойство адаптивности программного комплекса имитационного моделирования СУЗ.

Таким образом, предложенная архитектура программного комплекса имитационного моделирования СУЗ позволяет проводить динамическую настройку

систем управления заданиями в соответствии с характеристиками входного потока заданий за счёт реализации механизма обратной связи и обеспечивает интеграцию с различными системами управления заданиями без модификации их исходного кода. Архитектура соответствует требованиям, сформулированным в главе 1, и отличается от известных решений возможностью интеграции с произвольными функционирующими СУЗ (включая СУППЗ) без модификации их исходного кода и механизмом обратной связи для адаптивной настройки параметров моделирования в соответствии с характеристиками входного потока заданий.

В четвертой главе представлены результаты экспериментов по определению влияния методики ФПЗТ на показатели качества обработки заданий в СУЗ и границ применимости этой методики. Было проведено 3 серии экспериментов.

Для исследования было сформировано две группы входных потоков. Первая группа под условным названием flow1 была сформирована на основе генератора Lublin&Feitelson, предложенного в фундаментальной работе U. Lublin и G. Feitelson³. Генератор формирует модельный поток заданий на основе статистики работы нескольких известных суперкомпьютеров. Каждая характеристика задания рассматривается как случайная величина с определённым видом и параметрами распределения, соответствующими входным потокам реальных суперкомпьютеров.

Потоки группы flow1 характеризуются существенной неоднородностью, которая выражается в значительном разбросе времени обработки заданий от 1 секунды до нескольких суток. Для обеспечения большей однородности генератор Lublin&Feitelson был модифицирован автором диссертации таким образом, чтобы при генерации отбрасывались задания со временем выполнения менее e^5 (148 секунд) и более e^9 (2980 секунд = 50 минут). С использованием модифицированного генератора была сформирована группа с условным названием flow2, включающая более однородные по сравнению с flow1 входные потоки. В состав групп flow1 и flow2 было включено по 3 входных потока из 5000 заданий в каждом. Каждый поток обеспечивает один из трёх уровней полной загрузки: 0,85 (умеренная загрузка), 0,9 (высокая загрузка) и 0,95 (интенсивная загрузка).

Результаты первой серии экспериментов, приведённые на рисунке 6, продемонстрировали положительное влияние методики ФПЗТ на показатели качества

³ Lublin U., Feitelson G. The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Job // Journal of Parallel and Distributed Computing Archive. 2003. V. 63. № 11. P. 542–546. DOI: 10.1016/S0743-7315(03)00108-4.

обработки заданий. Применение пакетирования позволило существенно увеличить полезную загрузку (3). НРО были рассчитаны в соответствии с (1). При НРО в 10% полезная загрузка за счёт пакетирования возросла с 86% до 99%, при НРО в 100% возросла с 48% до 89%.

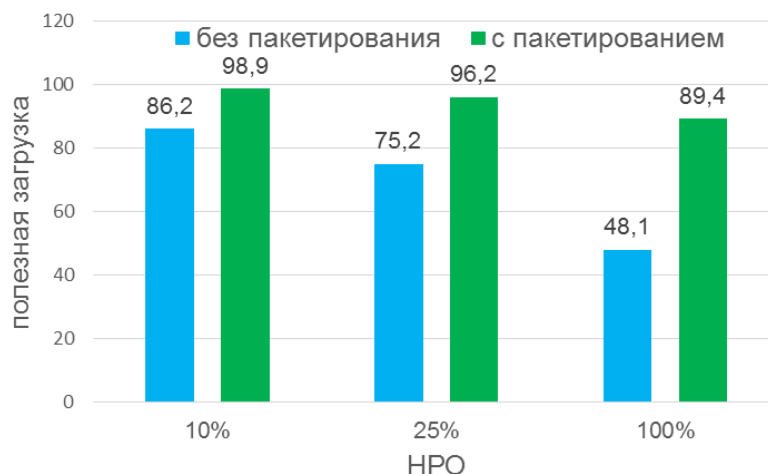


Рисунок 6 – Влияние накладных расходов на полезную загрузку для НРО 10%, 25%, 100% с формированием пакетов и без

В ходе второй серии экспериментов были определены границы применимости методики ФПЗТ, при которых пакетирование улучшает показатели качества обработки заданий. Были определены минимальные значения НРО для входных потоков разной интенсивности и однородности. Результаты приведены в таблице 2.

Таблица 2. Минимальные накладные расходы на обработку задания, для которых методика ФПЗТ обеспечивает улучшение показателей качества

Вид потоков	Неоднородные			Однородные		
	Умеренная (0,85)	Высокая (0,90)	Интенсивн. (0,95)	Умеренная (0,85)	Высокая (0,90)	Интенсивн. (0,95)
Минимальные НРО	10%	8%	6%	9%	6%	3%

Для неоднородного потока группы flow1 с умеренной загрузкой 0,85 пакетирование обеспечивает улучшение показателей качества СУЗ, начиная с 10% средних накладных расходов на обработку задания. С ростом интенсивности и повышением однородности входного потока положительное влияние пакетирования на показатели качества достигается с 6% НРО для неоднородного входного потока и с 3% для однородного входного потока.

В третьей серии экспериментов для входных потоков групп flow1 и flow2 исследовано влияние показателя масштабирования как главного параметра методики

ФПЗТ на качество обработки заданий. На основе результатов третьей серии экспериментов в диссертации предложена методика автоматизированного определения показателя масштабирования, рекомендуемая для применения при настройке системы управления заданиями системным администратором. Методика позволяет для фиксированного входного потока заданий в автоматизированном режиме итеративно изменять показатель масштабирования для поиска значения, которое обеспечит требуемый баланс показателей качества обработки заданий, таких как полная загрузка и среднее время ожидания задания в очереди.

В **заключении** результаты диссертации обобщены как комплекс новых научно обоснованных технических, технологических и архитектурных решений, внедрение которых вносит вклад в развитие высокопроизводительных вычислительных систем коллективного пользования как составной части научной инфраструктуры страны.

Разработанный программный комплекс имитационного моделирования СУЗ внедрён в практику МСЦ РАН – Отделения суперкомпьютерных систем и параллельных вычислений НИЦ «Курчатовский институт» и использован при выполнении 7 научно-исследовательских работ по программам фундаментальных научных исследований государственных академий наук на 2013-2020 годы и на 2021-2030 годы, в том числе:

- FNEF-2022-0014 «Исследование, разработка и развитие методов и средств организации высокопроизводительных вычислений, интеграции информационных ресурсов различного вида, формирования цифрового пространства научных знаний и интегрированной инфраструктуры научных, образовательных и ведомственных информационных сетей»;

- FNEF-2022-0016, FNEF-2024-0016 «Разработка архитектур, системных решений и методов для создания и использования высокопроизводительных вычислительных комплексов, в том числе гетерогенных суперкомпьютеров и нетрадиционных архитектур микропроцессоров».

Основные результаты диссертации:

1. Предложенный метод имитационного моделирования систем управления заданиями позволяет оценивать точность исследуемых моделей при помощи количественного показателя близости выходных потоков заданий и улучшать характеристики систем управления заданиями для заданного входного потока заданий в ходе итеративного процесса настройки параметров модели. Метод отличается от известных применением количественного показателя точности имитационных моделей.

С помощью экспериментального исследования показано, что предложенный метод адекватно ранжирует априорно разные имитационные модели по точности моделирования.

2. Предложенная методика формирования пакетов заданий по типам, применяемая для обработки потока заданий разных типов в системах управления заданиями позволяет повысить полезную загрузку вычислительных ресурсов за счёт однократной инициализации заданий одного типа и определять момент запуска, размер пакета заданий, требуемое количество вычислительных узлов для обработки сформированного пакета. Для методики формирования пакетов заданий по типам определены границы применимости в виде минимальных значений НРО для входных потоков различной интенсивности и однородности. Методика была реализована в программе для ЭВМ «Подсистема объединения суперкомпьютерных заданий в группы по типам» [15].

3. Предложенная архитектура программного комплекса имитационного моделирования систем управления заданиями позволяет проводить динамическую настройку систем управления заданиями в соответствии с характеристиками входного потока заданий за счет реализации механизма обратной связи для адаптивной настройки параметров моделирования и обеспечивает интеграцию с различными системами управления заданиями без модификации их исходного кода. Архитектура отличается от известных количественной оценкой точности имитационного моделирования, адаптивностью ко входному потоку заданий и интеграционной совместимостью с произвольной системой управления заданиями.

Результаты работы могут быть использованы при совершенствовании систем управления задания в центрах коллективного пользования.

СПИСОК ОСНОВНЫХ ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Lyakhovets D.S., A.V. Baranov. Group Based Job Scheduling to Increase the High-Performance Computing Efficiency // Lobachevskii Journal of Mathematics. 2020. Vol. 41, No. 12, pp. 2558–2565. URL: <https://doi.org/10.1134/S1995080220120264>.
2. Lyakhovets D.S., Baranov A.V. Efficiency thresholds of group based job scheduling in HPC systems// Lobachevskii J. Math. 2022. Vol. 43, pp. 2863–2876. URL: <https://doi.org/10.1134/S1995080222130261>
3. Savin G.I., Lyakhovets D. S., Baranov A. V. Influence of Job Runtime Prediction on Scheduling Quality // Lobachevskii Journal of Mathematics. 2021. Vol. 42, No. 11. P. 2562–2570. URL: <http://doi.org/10.1134/S1995080221110196>.

4. Lyakhovets D.S., Baranov A.V., Telegin P.N. Scale Ratio Tuning of Group Based Job Scheduling in HPC Systems // Lobachevskii J Math. 2023. Vol. 44, pp. 5012–5026. URL: <https://doi.org/10.1134/S1995080223110240>.
5. Baranov A. V., Lyakhovets D. S. Accuracy Comparison of Various Supercomputer Job Management System Models // Lobachevskii Journal of Mathematics. 2021. Vol. 42, No. 11. P. 2510-2519. URL: <http://doi.org/10.1134/S199508022111007X>
6. Simulator of a Supercomputer Job Management System as a Scientific Service / G. Savin, B. Shabanov, D. Lyakhovets, A. Baranov, P. Telegin // ACSIS, 2020, Vol. 21, pp. 413–416, URL: <http://doi.org/10.15439/2020F208>.
7. Lyakhovets D. S, Baranov A.V., Konstantinov P.A. A Method for Combining Heterogeneous Workflows in HPC Systems // Lobachevskii Journal of Mathematics. 2024. Vol. 45, No. 10. P. 5111-5125. URL: <http://doi.org/10.1134/S1995080224606131>.
8. Баранов А.В., Ляховец Д.С. Имитационная модель системы пакетирования суперкомпьютерных заданий на базе симулятора Alea // Программные продукты и системы. 2022. № 4. С. 631-643. URL: <https://doi.org/10.15827/0236-235X.140.631-643>.
9. Баранов А.В., Ляховец Д.С. Методы и средства моделирования системы управления суперкомпьютерными заданиями // Программные продукты и системы. 2019. № 4. С. 581-594. URL: <https://doi.org/10.15827/0236-235X.128.581-594>.
10. Баранов А.В., Ляховец Д.С. Влияние пакетирования на эффективность планирования параллельных заданий // Программные системы: теория и приложения. 2017. Том 8, вып. 1. С. 193-208. URL: <https://doi.org/10.25209/2079-3316-2017-8-1-193-208>.
11. Баранов А.В., Киселев Е.А., Ляховец Д.С. Квазипланировщик для использования простаивающих вычислительных модулей многопроцессорной вычислительной системы под управлением СУППЗ // Вестник Южно-Уральского государственного университета. Серия: вычислительная математика и информатика, М.: Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск), 2014, т.3. 124 с, с. 75-84. URL: <http://dx.doi.org/10.14529/cmse140405>.
12. Baranov A., Lyakhovets D., Savin G., Shabanov B., Telegin P. Measure of adequacy for the supercomputer job management system model. Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019, ACSIS, Vol. 18, pp. 423-426. DOI: 10.15439/2019F186.
13. Баранов А.В., Ляховец Д.С. Исследование механизма пакетирования суперкомпьютерных заданий в средстве моделирования Alea / // Суперкомпьютерные дни в России: Труды международной конференции, Москва, 26–27 сентября 2022 года. – Москва: ООО "МАКС Пресс", 2022. – С. 144-145.
14. Ляховец Д.С., Баранов А.В., Кудрин А.Ю. Симулятор системы управления суперкомпьютерными заданиями с внешним интерфейсом управления // Труды научно-исследовательского института системных исследований Российской академии наук. 2024. Т. 14, № 4. С. 75-83 – ISSN 2225-7349.

Свидетельство о государственной регистрации программы для ЭВМ:

15. Баранов А.В., Ляховец Д.С., Шабанов Б.М., «Подсистема объединения суперкомпьютерных заданий в группы по типам». РФ, свидетельство о государственной регистрации программы для ЭВМ № RU2025683856, 09.09.2025.