

На правах рукописи

Черных Андрей Николаевич

Методы и алгоритмы решения задач оптимизации ресурсов в
нестационарных распределенных гетерогенных
вычислительных средах

05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей.

Автореферат

на соискание ученой степени
доктора физико-математических наук

Москва – 2021

Работа выполнена в Федеральном государственном бюджетном учреждении науки Институт системного программирования им. В.П. Иванникова Российской академии наук.

Научный консультант: Академик РАН, профессор РАН, д.ф.-м.н.,

Аветисян Арутюн Ишханович

Официальные оппоненты:

- Якобовский Михаил Владимирович, доктор физико-математических наук, член-корр. РАН, заместитель директора по научной работе Федерального государственного учреждения «Федеральный исследовательский центр Институт прикладной математики им.М.В.Келдыша Российской академии наук»,
- Ильин Вячеслав Анатольевич, доктор физико-математических наук, старший научный сотрудник, главный научный сотрудник Курчатовского комплекса НБИКС-природоподобных технологий Федерального государственного бюджетного учреждения «Национальный исследовательский центр «Курчатовский институт»,
- Шабанов Борис Михайлович, доктор технических наук, доцент, директор Межведомственного Суперкомпьютерного Центра РАН, заместитель директора по научной работе Федерального государственного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук».

Ведущая организация: Объединенный институт ядерных исследований.

Защита состоится 09 декабря 2021 г. в 15 часов на заседании диссертационного совета Д 002.087.01 при Федеральном государственном бюджетном учреждении науки Институт системного программирования им. В. П. Иванникова РАН по адресу: 109004, г. Москва, ул. А. Солженицына, дом 25.

С диссертацией можно ознакомиться в библиотеке и на сайте Федерального государственного бюджетного учреждения науки Институт системного программирования им. В. П. Иванникова РАН.

Автореферат разослан “ ____ ” _____ 2021 г.

Ученый секретарь
диссертационного совета Д.002.087.01,
кандидат физико-математических наук

Зеленов С.В.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования. Теория оптимизации в планировании вычислений находит широкое применение для эффективного решения фундаментальных и прикладных задач, обеспечивающих конкурентное преимущество в научной, производственной, экономической и других сферах человеческой деятельности. Современные распределенные компьютерные системы открыли принципиально новые возможности по увеличению вычислительных мощностей для решения сложных задач. Их характеризуют хорошая масштабируемость, возможность гибкого управления нагрузкой, надежность и отказоустойчивость, расширяемость, безопасность и живучесть в условиях кибератак, природных катастроф и др.

Существует множество различных парадигм реализации распределенных вычислений, и, как следствие, огромное разнообразие научных и технологических направлений, от кластеров, добровольных вычислений (англ., volunteer computing), одноранговых децентрализованных и пиринговых (англ., Peer-to-Peer – P2P) сетей до грид- (англ., grid computing) и облачных технологий (англ., cloud computing). В рамках облачных вычислений пул абстрактных, виртуализованных, динамически-масштабируемых вычислительных ресурсов, систем хранения данных, платформ и сервисов предоставляется внешним пользователям по запросу через Интернет. Хотя распределенные вычисления обладают многими неоспоримыми преимуществами, они все еще имеют много узких мест, особенно в области эффективности, безопасности, надежности и т. п.

подавляющее большинство исследований в области планирования распределенных вычислений предполагает наличие полной информации о работах и среде их выполнения. Однако облачные вычисления подвержены значительной нестационарности во время обеспечения доступа к ресурсам и их использования. Это создает дополнительные сложности для конечных пользователей, поставщиков ресурсов, сервисных провайдеров и систем планирования.

Объектом исследования в данной диссертации являются задачи планирования распределенных вычислений, которые, в свою очередь, являются подклассом более широкого класса, как правило, NP-трудных задач комбинаторной оптимизации. Ключевое положение планирования вычислений состоит в том, чтобы сопоставить набор работ с набором ресурсов, рассматривая один или несколько критериев оптимизации. Как правило, используются два способа планирования: статическое и динамическое. При статическом подходе

полная информация о работах, характеристиках процессоров, операционных системах, а также топологии вычислительной системы известна заранее. Производительность же, например, облачных ресурсов предсказать сложно, т. к. эти ресурсы независимы, гетерогенны и разделяются множеством пользователей. Их архитектура, топология, каналы связи и вычислительные характеристики заранее неизвестны.

При виртуализации ресурсов трудно получить точные знания о состоянии системы. Кроме того, провайдеры постоянно ищут новые пути модернизации инфраструктуры и методы управления ресурсами для улучшения характеристик виртуальной среды и поддержания качества обслуживания (англ., Quality of Service – QoS) ее пользователей.

Динамическое планирование вычислений, обеспечивающее как эффективность использования ресурсов, так и преодоление негативных последствий нестационарности, является одним из ключевых требований к современным вычислительным системам, что постоянно усиливает внимание к этой проблематике.

Недостаточная изученность такого рода ситуационного планирования, несовершенство методов и моделей приводят к существенному недоиспользованию возможностей вычислительных систем.

Решение для современных гетерогенных распределенных вычислительных сред (в дальнейшем именуемых ГРИД) видится в переходе от традиционного планирования к новым адаптивным подходам, совершенствующим существующие стратегии распределения ресурсов и учитывающим стремительную эволюцию вычислительных средств.

В диссертации рассматриваются роль неопределенности в планировании ресурсов вычислительных систем и алгоритмы, работающие в условиях нестационарности, к которой приводят: требование эластичности вычислений, изменение характеристик машин, виртуализация вычислений при слабой связи работ с инфраструктурой, на которой они выполняются, динамическая миграция работ, изменчивость времени предоставления ресурсов, неточность оценки времени выполнения работ, изменчивость времени обработки и передачи данных в зависимости от использования ресурсов другими пользователями, вариативность рабочей нагрузки, разнообразие временных ограничений обработки (директивные сроки), изменение пропускной способности, сбои, кибератаки и другие явления.

Оптимизация в таких сложных системах должна учитывать не только производительность и эффективность использования ресурсов, а также требования пользователей по качеству обслуживания, стоимости услуг, затратам провайдеров, среднему времени окончания работ, среднему времени ожидания

начала их выполнения и др.

Актуальной научной проблемой, на решение которой направлена данная работа, является выработка фундаментальных основ планирования нестационарных ресурсов, их анализ, и разработка новых адаптивных алгоритмов для различных сценариев.

Использование частных решений, разработанных как для стационарных, так и для динамических систем, приводит к необходимости существенной доработки алгоритмов для каждой существующей системы под конкретные условия использования, что увеличивает расходы и сроки внедрения этих систем.

Необходимо рассматривать подходы к решению с разных сторон: онлайн алгоритмы, планирование работ в отсутствие достоверной информации о параметрах системы, самих работах и т. п.

Анализ современных тенденций развития планирования ресурсов, базирующийся на теоретических и практических исследованиях ведущих российских и зарубежных ученых, позволяет сделать важный вывод, что для решения этой актуальной проблемы необходим комплексный подход к построению адаптивных планировщиков и математических моделей, учитывающих отсутствие точных знаний при формировании плана работ.

Для решения поставленной общей научной проблемы проведена ее декомпозиция на ряд частных задач, включающих:

- разработку концепции планирования ресурсов в нестационарных вычислительных средах;
- создание современных моделей и методов планирования и оптимизации ресурсов в условиях изменяющихся характеристик системы;
- разработку методов динамического планирования с механизмами адаптации к изменению параметров среды и непредсказуемости рабочей нагрузки;
- создание подходов к построению адаптивных систем планирования динамически-масштабируемых параллельных работ;
- поддержка механизмов адаптации к многообразию характеристик работ пользователей и требований к их выполнению с поддержкой разных уровней обслуживания и стоимости выполнения работ.

Решение этих проблем с помощью широко используемой теории расписаний и существующих алгоритмов неосуществимо в полной мере, т. к. они изначально разрабатывались для более стационарных систем.

Планирование работ на мультипроцессорах хорошо изучено в течение десятилетий. Существует множество результатов исследований различных вариаций проблемы для одной системы. Некоторые из них дают теоретические

знания, в то время как другие дают рекомендации по реализации реальных систем.

Успешные теоретические решения в области планирования распределенных вычислений в гетерогенных средах с помощью аппарата теории расписаний предложили U. Schwiegelshohn, M. Yahyapour, A. Garey, A. Graham, J. Lenstra, D. Shmoys, F. Pascual, D. Trystram, K. Jansen, P. Bouvry, A. Zomaya, F. Werner, J. Blazewicz, A. Steinberg, E. Pesch, H. Karatza, E. Vampis, M. Pinedo, P. Brucker, Н. Кузюрин. С. Жук, В. Танаев, Ю. Сотсков, В. А. Струсевич, А. Кононов, В. Топорков, А. Лазарев и другие авторы.

Теория планирования в ГРИД тесно связана с множественной упаковкой полос, которая известна как NP-трудная и имеет много реальных приложений. Машины в ГРИД рассматриваются как полосы, а приложения – как прямоугольники, высота и ширина которых равны, соответственно, времени работы и требуемому числу процессоров. При такой постановке задачи решения ищутся с ориентацией на статические характеристики программно-аппаратной инфраструктуры и пользовательских приложений. Для таких проблем, получены теоретические результаты, если время выполнения работ задано, как в оффлайн, так и в онлайн сценариях. Следовательно, они не могут в полной мере поддерживать эффективное функционирование в средах с нестационарными характеристиками.

Все вышесказанное позволяет сделать вывод о необходимости и актуальности разработки новых моделей, алгоритмов, и стратегий организации планирования в ГРИД.

Цель работы состоит в разработке новых стратегий планирования распределенных вычислений в нестационарных гетерогенных средах, обеспечивающих эффективное выполнение работ на основе использования динамического и адаптивного планирования при ограниченном знании о ресурсах и работах.

Основными задачами для достижения поставленной цели являются:

- анализ современных подходов к планированию работ в нестационарных вычислительных системах;
- разработка модели ГРИД, не требующей достоверной информации о параметрах среды и о времени выполнения работ;
- разработка алгоритмов эффективного планирования независимых параллельных работ для различных сценариев;

- получение теоретических оценок конкурентных и аппроксимационных факторов для статического, динамического и адаптивного планирования с различными критериями оптимизации.

Объектом исследования является планирование работ в распределенных гетерогенных вычислительных системах.

Предметом исследования выступают модели, алгоритмы и планировщики выполнения вычислительных работ в ГРИД.

Методы исследования. При решении поставленных задач использовались методы теории расписаний, теории алгоритмов, модели управления ресурсами в вычислительных системах, методы теоретического анализа конкурентных и аппроксимационных факторов статических и динамических расписаний в различных сценариях организации вычислений с различными критериями оптимизации.

Научную новизну диссертации представляют следующие основные научные результаты, выносящиеся на защиту и расширяющие существующий базис теории и практики планирования распределенных гетерогенных вычислений. Все результаты, представленные в данной диссертации, являются новыми.

- Сформулирована проблема планирования в гетерогенной распределенной вычислительной системе в условиях ее нестационарности и предложены подходы к ее решению путем динамической адаптации к изменению системных параметров.

- Показано, что планирование в ГРИД является более сложной задачей, чем соответствующее планирование для многопроцессорной системы, а также то, что хорошо известный списочный алгоритм составления расписаний не может гарантировать постоянного конкурентного фактора, хотя в случае с многопроцессорными системами он достигается.

- Доказано, что граница списочного алгоритма $2 - \frac{1}{m}$ Гарей и Грэхэма (одновременная подача работ), а также Naroska и Schwiegelshohn (построение онлайн расписания), к ГРИД не применима. Поэтому она не может быть гарантирована никаким полиномиальным алгоритмом, если только не $P = NP$.

- Представлены решения для планирования в ГРИД, расширяющие решения для многопроцессорного планирования. Поскольку обычное списочное расписание не подходит для ГРИД, представлен подход планирования, использующий несколько списков работ для каждой машины. В короткой нотации $\alpha | \beta | \gamma$ эта проблема характеризуется как $GP_m | size_j | C_{max}$ и $GP_m | r_j, size_j | C_{max}$, где α – характеристики машин, β – характеристики работ, γ – целевая функция задачи.

- Разработан алгоритм планирования работ с фиксированной степенью параллелизма и неизвестным временем обработки, использующий подход «кражи работ» (англ., job stealing), для которого доказан конкурентный фактор 5 для онлайн случая и аппроксимационный фактор 3 для оффлайн случая.
- Предложена и проанализирована адаптивная схема планирования работ с использованием концепции допустимого распределения, которая исключает определенные машины из набора машин, доступных для назначения определенной работы. Проанализирована двухуровневая модель ГРИД, которая интегрирует обе задачи планировщика: распределение параллельных работ с неопределенными требованиями ко времени выполнения по машинам и составление локальных расписаний.
- Получены 3-х и 9-ти аппроксимационные, а также 5-ти и 11-ти конкурентные факторы для алгоритмов, что улучшает и расширяет известные результаты. Решения по планированию принимаются без точной информации о производительности узлов и времени выполнения работ. Конкурентный и аппроксимационный факторы адаптивного алгоритма планирования варьируются путем изменения коэффициента допустимости.
- Представлено подробное исследование влияния коэффициента допустимости, включенного в политику планирования, на общую эффективность работы системы. Показано, как коэффициент допустимости может быть динамически скорректирован для того, чтобы справиться с динамическим изменением рабочей нагрузки, улучшая конкурентный фактор.
- Представлено новое семейство стратегий планирования, основанных на двух фазах, которые последовательно сочетают последовательное и параллельное выполнение работ.
- Предложено планирование работ с неизвестными временами выполнения, фокусирующееся на регулировании периодов простоя машин в контексте общей политики списочных расписаний.
- Обобщены известные предельные границы производительности планирования в наихудшем случае. Введены два дополнительных параметра, помимо числа процессоров и максимального параллелизма работ, рассматриваемых в литературе: штраф за распараллеливание работ и коэффициент регулирования простоя.
- Показано как при регулировании простоя найти компромисс между более ранним началом распараллеливания (когда распараллеливается больше работ, что приводит к существенным накладным расходам на параллелизацию) и задержкой их распараллеливания (когда больше процессоров остаются без

работы) до тех пор, пока не будет доступно меньшее число работ. Эта схема балансирует потребности пользователя с потребностями компьютерной системы.

- Основываясь на моделях реального времени, предложена оригинальная модель облачных вычислений для выполнения и планирования работ на основе понятия уровня обслуживания. Используя обозначение трех полей, проблема может быть описана как $1|prmp, r_{j,online}, S_i|\sum v_j$ или $P_m|prmp, r_{j,online}, S_i|\sum v_j$, в зависимости от выбранной модели машины. Каждый уровень обслуживания определяется слак-фактором и ценой за единицу времени обработки.

- Предложен ряд алгоритмов и проведен конкурентный анализ для различных сценариев с одним (SSL) или несколькими (MSL) уровнями обслуживания, с одной (SM) или несколькими (MM) машинами. Для этих сценариев проанализированы алгоритмы с жадным и ограниченным принятием работ и получены границы конкурентных факторов.

Практическая и теоретическая значимость. Работа носит теоретический характер. Полученные в ней результаты позволяют понять природу различных аспектов планирования в распределенных гетерогенных вычислительных системах в условиях нестационарности, и получить теоретические оценки границ оптимизации. Предложены новые приближенные алгоритмы с улучшенными по сравнению с известными в литературе аппроксимационными и конкурентными факторами.

Применение вышеперечисленных результатов диссертационного исследования обеспечивает повышение эффективности планирования параллельных работ в современных вычислительных системах путем его адаптации к изменению параметров рабочей нагрузки и системы.

Также отметим, что предложенные алгоритмы имеют низкую вычислительную сложность, и при этом позволяют получать хорошие приближенные решения.

Практическая и теоретическая значимость полученных результатов и вклад диссертанта в развитие соответствующей отрасли знаний подтверждается цитированием результатов в международных изданиях: 2441 ссылки в Google Scholar (h-index = 25), 1318 ссылки в Scopus (h-index = 18).

Основные результаты диссертационного исследования были использованы в рамках следующих научно-технических работ:

МиноБР – Министерство образования и науки Российской Федерации, "Исследование и разработка передовых методов защиты информации, сохранения конфиденциальности и предотвращения утечки данных при обработке данных в распределенных средах" (Проект 075-15-2020-788), 2020-2022.

РФФИ – Российский фонд фундаментальных исследований, "Методы и алгоритмы сбора и обработки данных Internet of Things на основе облачных и туманных вычислительных систем для поддержки интеллектуальных систем мониторинга и автоматизации "Умный город" (проект 20-47-740005), 2020-2021 гг., "Разработка моделей, методов и алгоритмов календарного планирования для контейнерных вычислительных ресурсов при выполнении приложений документооборота в рамках концепции цифрового предприятия" (Проект 18-07-01224), 2018-2020 гг.

CICESE Research Center, Мексика, "Планирование в Федерации Облаков" (#634122), 2016-2121, "Оптимизация ресурсов в Гридах и системах реального времени" (№634107), 2008-2011 гг., "Проблемы оптимизации ресурсов в кластерах и GRID" (#634107), 2005-2007.

CONACYT. Национальный совет по науке и технике Мексики, Мексика, "Стратегии управления и оптимизации энергосистемных ресурсов с качеством обслуживания", 2012-2017 гг., – CONACYT #178415, "Многоуровневая иерархия стратегий планирования работ для вычислительного Грида", 2006. CONACYT #48385.

АНИ – Национальное агентство по изучению и инновациям, Уругвай, 2015, Сотрудничество с Институтом вычислений Инженерного факультета, Университет Республики. Уругвай.

FNR – (Национальный фонд научных исследований, Люксембург) сотрудничество с Люксембургским университетом, Люксембург, "Стратегии управления параллельными работами и их оптимизации в облаках", "Экологически безопасные стратегии параллельного управления работами и их оптимизация в облачных системах P-2-P".

DAAD – Германская служба академических обменов, сотрудничество с Университетом Геттингена – Институт вычислительной техники, Геттинген, Германия, "Управление ресурсами в облачных вычислениях", "Планирование рабочей нагрузки на двухуровневых архитектурах Грид", "Допустимые стратегии выбора ресурсов при двухуровневом планировании работ для вычислительной среды"

BSC – Барселонский суперкомпьютерный центр, Испания "Неопределенность в облачных вычислениях".

INRIA Lille – Nord Europe, Исследовательский Центр ИНРИА Лилль, challenges of efficient resource provisioning under Uncertainty.

Гренобльский университет в Альпах, ИНРИЯ, Гренобль, Франция, эффективное распределение ресурсов.

NSF Национальный научный фонд США – CONACYT сотрудничество с Университетом Южной Калифорнии, Лос-Анджелес, США, "Система

кэширования времени исполнения I-структуры как структура данных для параллельных кластерных вычислений", (CONACYT #32989-A), 2001-2004.

Под руководством Черных А. Н. разработан комплекс программ, позволяющий исследовать вопросы обеспечения безопасности и надежности хранимых и обрабатываемых данных в облаках в условиях неопределенности возникновения технических сбоев и различного рода хакерских атак, состоящий из: «Программа моделирования работы устройств концепции интернет вещей на базе системы остаточных классов» (Свидетельство о государственной регистрации программы для ЭВМ №2017660880, дата регистрации 28.09.2017); «Модуль оценки рисков безопасности облачных, краевых и туманных вычислений в условиях вычислительной неопределенности» (Свидетельство о государственной регистрации программы для ЭВМ №2018612694, дата регистрации 21.02.2018); «Среда моделирования адаптивной цифровой фильтрации в системе остаточных классов» (Свидетельство о государственной регистрации программы для ЭВМ №2019610639, дата регистрации 15.01.2019); «Распределенная система надежного хранения и обработки данных в мультиоблачной среде» (Свидетельство о государственной регистрации программы для ЭВМ №2019611375, дата регистрации 24.01.2019).

Разработанный комплекс программ используется при проведении совместных научных исследований, проводимых научными группами в Северо-Кавказском федеральном университете, Институте системного программирования им. В. П. Иванникова Российской академии наук, Южно-Уральском государственном университете, Московском физико-техническом институте (национальном исследовательском университете) и др.

Достоверность и обоснованность полученных в диссертации результатов подтверждена корректным применением классических методов исследования, строгими доказательствами и анализом эффективности разработанных моделей и алгоритмов. Результаты согласуются с проведенными численными экспериментами.

Соответствие диссертации паспорту специальности. Тема и основные результаты диссертации соответствуют следующим областям исследований паспорта специальности ВАК 05.13.11 – «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»:

- Модели, методы, алгоритмы, языки и программные инструменты для организации взаимодействия программ и программных систем;

- Модели и методы создания программ и программных систем для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования;
- Модели, методы, алгоритмы и программная инфраструктура для организации глобально распределенной обработки данных.

Апробация работы. Все результаты диссертационного исследования прошли апробацию на научных мероприятиях в России и за рубежом. Среди них выделим:

Российские конференции: Международная конференция "Суперкомпьютерные дни в России" 2020, 2019, 2018, 2017, 2015. Открытая конференция ИСП РАН им. В.П. Иванникова, 2020, 2019, 2018. Международная конференция «Инжиниринг & Телекоммуникации», 2020, 2019, 2017, 2016, 2015. Национальный Суперкомпьютерный Форум (НСКФ-2020) Россия, Переславль-Залесский, 2015.

Международные симпозиумы: IPDPS – IEEE International Parallel and Distributed Processing, 2021, 2019, 2018, 2017, 2012, 2010, 2008, 2006 (CORE Rank A). CCGRID – IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing 2020, 2016 (CORE Rank A). Euro-Par – International European Conference on Parallel and Distributed Computing 2019, 2013: 2008 (CORE Rank A). CloudCom – IEEE International Conference on Cloud Computing Technology and Science. 2015 (CORE Rank A). ICCS – International Conference on Computational Science (2015, 2018) (CORE Rank A). HPCS – International Conference on High Performance Computing & Simulation 2020, 2019, 2018, 2014, 2012 (CORE Rank B). InterCloud-HPC – International Symposium on Cloud Computing and Services for High Performance Computing Systems 2020 (CORE Rank B). PDCAT – International Conference on Parallel and Distributed Computing, Applications and Technologies. 2020 (CORE Rank B). DEXA – International Conference on Database and Expert Systems Applications 2017 (CORE Rank B). GLOBECOM – IEEE Global Communications Conference (2015) (CORE Rank B). CLOUD – IEEE International Conference on Cloud Computing. 2013 (CORE Rank B). CollaborateCom – International Conference on Collaborative Computing, 2019, (CORE Rank C). PPAM – International Conference on Parallel Processing and Applied Mathematics 2017, 2011, 2007, 2005, 2003 (CORE Rank C). UCC – IEEE/ACM International Conference on Utility and Cloud Computing Companion, 2018, 2017, 2016. CARLA – Latin America High Performance Computing Conference, 2019, 2018, 2017, 2016. ISUM – International Supercomputing Conference in México 2019. 2017, 2016, 2014, 2012, 2010.

Основные результаты были апробированы и обсуждены на приглашенных пленарных докладах и семинарах.

Huawei 2020 – Cloud Resource Scheduling and Optimization Workshop. Moscow, 2020. ICCS-DE 2020 – 2nd International Workshop on Information, Computation, and Control Systems for Distributed Environments, Irkutsk, Russia, 2020. ICCIS 2020 – 1st International Conference on Systems and Information Sciences. Manta, Ecuador, 2020. IFI – The Institute of Computer Science, Faculty of Mathematics and Computer Science, Georg-August-Universität Göttingen, Germany, 2019. ФГАОУ ВО «Северо-Кавказский федеральный университет», 2018. UNIANDÉS – Universidad de los Andes, Bogotá, Colombia, 2018. COMTEL 2018 – X International Conference on Computer Science and Telecommunication, Lima, Perú, 2018. ISUM 2018 – 9th International Supercomputing Conference in Mexico, Merida, México. 2018. MPEI – Moscow Power Engineering Institute (National Research University), Moscow. 2017. NNSU – Lobachevsky State University of Nizhni Novgorod, 2017. CARLA 2017 – Latin American Conference on High Performance Computing. Buenos Aires, Argentina, 2017. UCC 2017 – 1st International Workshop on Uncertainty in Cloud Computing. In conjunction with DEXA 2017, Lyon, France, 2017. PPAM 2017 – 12th International Conference on Parallel Processing and Applied Mathematics, Lublin, Poland, 2017. ISUM 2017 – 8th International Supercomputing Conference in Mexico, Guadalajara, México. 2017. Tsinghua University, Department of Computer Science and Technology, Beijing, China. 2016. SUSU – South Ural State University. Chelyabinsk, Russia. 2016. UES – State University of Sonora. San Luis Rio Colorado, Мексика 2016. ISUM 2016 – 7th International Supercomputing Conference in Mexico, Puebla, México. 2016. CCERD 2015 – The 6th International Conference “Cloud Computing. Education. Research. Development”. Moscow, Russia, 2015. UdelaR – University of the Republic, Uruguay, 2015. RuSCDays'15 – The Russian Supercomputing Days. 2015, Moscow. UMONS – University of Mons, Mons, Belgium. 2015. University of Dortmund, Dortmund, Germany. 2015, 2014, 2012. BSC – Barcelona Supercomputing Center, 2015. INRIA Lille – Nord Europe, 2015. CIMAT – Mexico, 2014. University of Notre Dame, South Bend, United States, 2013. University of Luxembourg. 2014, 2012. CGCT2009 – Collaborative and Grid Computing Technologies Workshop, Cancun, Mexico, 2009. PPAM 2003 – Fifth International Conference on Parallel Processing and Applied Mathematics, Czestochowa, Poland, 2003. Instytut Informatyki, Politechnika Poznanska, Poznan, 2003, 2001. SCI 2002 – 6th World Multiconference on Systemics, Cybernetics and Informatics. Orlando, USA, 2002. New Trends in Scheduling in Parallel and Distributed Systems, Marseille (FRANCE) 2008, 2001. INPG – Institut National Polytechnique de Grenoble, Grenoble, Francia, 2001. UJF – Université Joseph Fourier - Grenoble, Francia, 2000. Aussois – New Trend in Scheduling in Parallel and

Distributed Systems Aussois, Francia. 2014, 1998. Lessach – Workshop on Scheduling in Computer and Manufacturing System, Lessach, Austria. 1996.

Новизна результатов диссертации подтверждена в ходе приглашенных стажировок диссертанта в известных международных научных центрах: в Университете Цинхуа (Китай), Университете Геттингена, в Дортмундском университете, в Клаустальском техническом университете (Германия), Барселонском суперкомпьютерном центре (Испания). в Центре исследований ИНРИА Лилль – Северная Европа (Франция), Гренобльском Альпийском университете (Франция), Люксембургском университете (Люксембург), Национальном политехническом институте Гренобля (Франция), Калифорнийском университете Ирвайн (США), Университете Южной Калифорнии (США), Университете Жозефа Фурье (Франция), Университете Республики (Уругвай) и т. д.

Публикации. По теме диссертации автором было опубликовано 64 статьи, в том числе 21 статья в журналах из списка ВАК или включенных в международные базы данных Scopus и Web of Science из них 12 в Q1, 1 в Q2, 4 в Q3, 2 в Q4 [1–21], 42 работы – в трудах российских и международных конференций [22–63], получено 4 свидетельства о регистрации программ для электронных вычислительных машин (ЭВМ) (Приложение 1) и один патент [64] (Приложение 2).

Личный вклад автора. Диссертационная работа представляет собой многолетнее исследование автора, объединенное тематикой и методами исследования. В совместных работах соискателю принадлежат доказательства основных утверждений и теорем, а также основные идеи новых алгоритмов. Некоторые доказательства были сделаны в соавторстве при непосредственном участии соискателя. Конфликта интересов с соавторами нет.

В получение некоторых результатов были вовлечены 17 студентов магистратуры и 7 аспирантов под непосредственным руководством соискателя.

Структура работы. Диссертация состоит из введения, 7 глав, заключения, библиографии из 223 наименований, и 2 приложений. Общий объем основного текста работы – 294 страниц, включая 49 таблиц и 72 рисунка.

Краткое содержание работы.

В первой главе рассмотрены основные концепции планирования ресурсов в нестационарных вычислительных средах, приведен краткий обзор основных современных моделей и методов, используемых при решении соответствующих задач с помощью теории расписаний. Описана природа неопределенности и

приведена классификация неопределенностей.

Неопределенность возникает в результате неполного знания о процессах, контролируемых предоставлением сервисов, например, когда концептуальная модель системы, используемой для предоставления сервисов, не включает все влияющие процессы или факторы. Ее моделируют с помощью теории вероятностей, теории доказательств, теории возможностей и нечетких множеств.

Надежный дизайн планировщика сводит к минимуму влияние неопределенностей на производительность и поведение системы. Традиционно он осуществлялся либо на основе вероятностного подхода, либо на основе анализа наихудшего случая. Оба подхода рассматривают неопределенность либо как отсутствие информации, как случайные переменные, либо как интервальные переменные. На самом деле, неопределенность может представлять собой сочетание разных факторов.

Неопределенность – это важный параметр, который влияет на эффективность вычислений, создавая дополнительные проблемы при планировании. Она требует разработки новых стратегий распределения ресурсов. В первой главе рассмотрены подходы к предоставлению ресурсов, выполнению приложений в условиях изменяющихся характеристик системы. Показано, что стратегии должны обеспечить возможность динамического распределения ресурсов в ответ на изменяющиеся нагрузки, а также динамически адаптироваться к ним, чтобы справиться с различными потоками задач, учитывая свойства систем.

В главе проанализирована и классифицирована неопределенность облачных вычислений и рассмотрены подходы к ее снижению. Особое внимание уделено роли неопределенности в организации работы провайдеров, пользователей и системных администраторов. Рассмотрены динамические стратегии предоставления ресурсов и услуг, а также методы программирования при наличии неопределенности вычислительной среды. Эти проблемы являются весьма сложными и имеют ключевое значение для принятия решений по планированию работ. Другой важный вклад заключается в рассмотрении этих проблем в свете их сопоставления с другими проблемами и решениями. Кроме того, обсуждаются критерии оптимизации различных вариантов проблемы.

В главе представлено понимание того, как моделировать ГРИД с учетом неопределенности характеристик, динамики распределенных вычислений, масштабирования, отсутствия точной информации о времени вычисления работ и т. д. для предоставления надежных решений, где основной целью является не поиск абсолютно оптимальных решений, а нахождение приближенных решений нечувствительных к изменению параметров среды. Рассмотрены результаты исследований для различных вариаций проблемы планирования дающие

теоретические знания или рекомендации по реализации реальных систем.

Смещение акцентов в сторону сервис-ориентированной парадигмы привело к принятию SLA (соглашение об уровне сервиса) как очень важной концепции. Использование SLA при обслуживании пользователей является принципиально новым подходом к планированию работ. При таком подходе планировщики должны учитывать ограничения в отношении QoS независимо от поведения системы. Основная идея заключается в обеспечении различных уровней обслуживания, каждый из которых адресован различным клиентам, чтобы гарантировать время выполнения работы в зависимости от предоставляемых уровней сервиса.

Рассмотрены различные стохастические, реактивные, не требующие знаний и т. п., алгоритмы и эффективные альтернативы известным технологиям детерминированной оптимизации.

Во второй главе рассматриваются вопросы, связанные с онлайн-планированием параллельных работ с неизвестными параметрами и без прерываний. В прикладной базовой модели гетерогенная распределенная система состоит из большого числа процессоров, которые распределены по нескольким машинам разного размера. Работы являются независимыми, они имеют фиксированную степень параллелизма и поступают с течением времени. Работа может быть выполнена только на процессорах, принадлежащих одной и той же машине. Цель – минимизировать общую продолжительность выполнения всех работ.

В короткой нотации «машинная модель-ограничение-критерий оптимизации» эта проблема характеризуется как $GP_m | size_j | C_{max}$.

Показано, что производительность списочного алгоритма планирования Гарей и Грэхэма значительно хуже в распределенных системах, чем в мультипроцессорах.

Поскольку обычное списочное расписание не подходит для ГРИД, представлен планировщик, использующий несколько списков работ. Каждый из этих списков не требует какого-либо конкретного порядка.

Далее представлен алгоритм планирования, гарантирующий конкурентный фактор 5 и аппроксимационный фактор 3. Этот алгоритм может быть реализован с использованием подхода "кражи работ" (англ. "job stealing") и может хорошо подходить для использования в качестве отправной точки для алгоритмов планирования в реальных системах.

В главе сначала рассматривается случай одновременного поступления заявок $r_j = 0$, а затем результат распространяется на сценарий поступления заявок

в разное время.

Для определения нижней границы для конкурентного фактора рассмотрена соответствующая статическая проблема с полностью доступной информацией и $r_j = 0$. Эта проблема NP сложная, т. к. $P_m || C_{max}$ – особый случай проблемы $GP_m | size_j | C_{max}$.

Доказана следующая теорема о качественной оценке рассматриваемого планирования.

Теорема 2.1. *Не существует полиномиального алгоритма, который всегда выдает план S с $\frac{C_{max}(S)}{C_{max}^*} < 2$ для $GP_m | size_j | C_{max}$ и всех входных данных, за исключением $P=NP$.*

В соответствии с теоремой 2.1, граница списочного алгоритма $2 - \frac{1}{m}$ Гаррея и Грэхэма (одновременная подача работ), а также Naroska и Schwiegelshohn [**Ошибка! Источник ссылки не найден.**] (онлайн расписания), к ГРИД не применима. Более того, показано, что уже в случае одновременного поступления работ списочное планирование не может гарантировать постоянную границу $\frac{C_{max}}{C_{max}^*}$ для всех случаев проблемы.

Очевидно, что этот результат обусловлен дисбалансом нагрузки, т. к. машины с большим числом процессоров могут выполнять работы с небольшим параллелизмом, что приводит к вынужденному ожиданию выполнения параллельных работ. Это наблюдение предполагает сортировку списка по параллелизму работ в порядке убывания, так чтобы высоко параллельные работы планировались первыми, когда есть выбор. Однако показано, что такой подход также не гарантирует постоянного конкурентного фактора.

Общеизвестной нижней границей для оптимальной продолжительности в случае одновременной загрузки работ для решения проблемы планирования ГРИД является:

$$C_{max}^* \geq \max \left\{ \max_j p_j, \max_{1 \leq i \leq m} \frac{\sum_{j | size_j > m_{i-1}} p_j \cdot size_j}{\sum_{v=i}^m m_v} \right\}$$

По сравнению с границей проблемы $P_m || C_{max}$, эта граница также учитывает недоступность машин небольшого размера для обработки высоко параллельных работ из-за отсутствия возможности их многомашинного выполнения.

Алгоритм планирования ГРИД основан на специальном распределении работ по различным машинам. Это распределение представлено с помощью понятия категорий работ, назначаемых для каждой машины.

Определение 2.1. Для каждой машины M_i существует три различные категории работ:

1. $A_i = \{J_j \mid \max\{\frac{m_i}{2}, m_{i-1}\} < size_j \leq m_i\}$
2. $B_i = \{J_j \mid m_{i-1} < size_j \leq \frac{m_i}{2}\}$
3. $H_i = \{J_j \mid \frac{m_i}{2} < size_j \leq m_{i-1}\}$

Набор A_i содержит все работы, которые не могут быть выполнены на предыдущей (следующей меньшей) машине и требуют более 50% процессоров машины M_i . Набор B_i содержит все работы, которые не могут быть выполнены на предыдущей машине, но требуют не более 50% процессоров машины M_i . Набор H_i содержит все работы, которые требуют более 50% процессоров машины M_i , но также могут быть выполнены на предыдущей машине.

Доказана следующая теорема об аппроксимационном факторе в случае одновременной подачи работ.

Теорема 2.2. Алгоритм одновременной подачи данных (*Concurrent-Submission*) гарантирует $\frac{C_{max}}{C_{max}^*} < 3$ для всех входных данных и всех конфигураций ГРИД.

Доказана следующая теорема о конкурентном факторе в случае подачи работ во времени.

Теорема 2.3. Алгоритм *Over-Time-Submission* гарантирует $\frac{C_{max}}{C_{max}^*} < 5$ для всех входных данных в случае подачи работ во времени.

В третьей главе анализируются алгоритмы планирования работ, интегрирующие обе задачи планировщика ГРИД: распределение работ по машинам и их локальное расписание. Предлагается и анализируется адаптивная схема распределения работ с использованием концепции допустимого распределения (англ., *admissible allocation*). Основная идея этой схемы заключается в том, чтобы установить ограничения по распределению маленьких работ на большие машины и динамически адаптировать их к различным рабочим нагрузкам и свойствам системы. Это приводит к снижению возможного дисбаланса нагрузки, когда машины с большим числом процессоров могут выполнять работы с небольшим параллелизмом, что приводит к вынужденному ожиданию выполнения параллельных работ.

Представлены 3-х-аппроксимационный и 5-ти конкурентный алгоритмы, названные MLBa PS и MCTa PS для случая, когда все работы могут быть выполнены на самой маленькой машине, и представлены 9-аппроксимационный и 11-конкурентный алгоритмы для общего случая.

Алгоритмы планирования для двухуровневых моделей ГРИД можно разделить на глобальную часть распределения работ и локальную часть их

планирования. Таким образом, MPS рассматривается как двухэтапная стратегия планирования: $MPS = MPS_Alloc + PS$. На первом этапе каждая работа назначается на подходящую машину, используя заданное ограничение. На втором этапе алгоритм PS применяется к каждой машине для работ, назначенных на предыдущем этапе.

Легко заметить, что конкурентный фактор алгоритма MPS ограничен снизу конкурентным фактором алгоритма PS , учитывая вырожденную ГРИД, которая содержит только одну машину. Лучший из возможных онлайн алгоритмов PS имеет конкурентный фактор $2 - 1/m$, где m обозначает число процессоров в одной параллельной машине; см. результаты работы Naroska и Schwiegelshohn [**Ошибка! Источник ссылки не найден.**]. Следовательно, нижняя граница конкурентного фактора для любого общего двухуровневого онлайн MPS расписания составляет не менее $2 - 1/m$.

Проанализированы два алгоритма под названием $MLBa + PS$ и $MCTa + PS$, где не все машины допустимы для данной работы, хотя и способны выполнять эту работу. Показано, что производительность стратегий распределения зависит от коэффициента допустимости, который может быть динамически адаптирован к различным рабочим нагрузкам и изменениям в конфигурации. Для этого необходимо проанализировать рабочую нагрузку за определенный промежуток времени, чтобы определить соответствующий коэффициент. Временной интервал должен устанавливаться в соответствии с характеристиками нагрузки, динамикой ее изменения и конфигурацией системы.

Коэффициент допустимости $0 < a \leq 1$ параметризует допустимость машин, используемых для распределения работ. Обратите внимание, что, по крайней мере, одна машина допустима, т. к. a строго больше 0, в то время как $a = 1$ определяет, что все доступные машины допустимы.

Представлено подробное исследование влияния коэффициента допустимости, включенного в политику распределения работ, на общую эффективность работы системы.

Теорема 3.1. *Оффлайн-планирование параллельных работ на ГРИД с идентичными процессорами по алгоритму $MCTa + PS$ с коэффициентом допустимости $0 < a \leq 1$ обладает аппроксимационным фактором*

$$\rho \leq \begin{cases} 1 + \frac{2}{a^2}, & \text{if } a \leq \frac{S_{f,l}}{S_{f_0,m}} \\ 1 + \frac{2}{a(1-a)}, & \text{if } a > \frac{S_{f,l}}{S_{f_0,m}}, \end{cases}$$

где $1 \leq f_0 \leq f \leq l \leq t$ являются параметрами, которые зависят от конфигурации машины и рабочей нагрузки.

Теорема 3.2. *Оффлайн-планирование параллельных работ на ГРИД с идентичными процессорами по алгоритму MLBa+PS с коэффициентом допустимости $0 < a \leq 1$ имеет аппроксимационный фактор*

$$\rho \leq \begin{cases} 1 + \frac{2}{a^2}, & \text{if } a \leq \frac{S_{f,l}}{S_{f_0,m}} \\ 1 + \frac{2}{a(1-a)}, & \text{if } a > \frac{S_{f,l}}{S_{f_0,m}} \end{cases}$$

с параметрами $1 \leq f_0 \leq f \leq l \leq m$, которые зависят от конфигурации машины и нагрузки.

Теорема 3.3. *Онлайн-планирование параллельных работ на ГРИД с идентичными процессорами с использованием алгоритмов MCTa + PS и MLBa + PS с коэффициентом допустимости $0 < a \leq 1$ имеет конкурентный фактор*

$$\rho \leq \begin{cases} 3 + \frac{2}{a^2}, & \text{if } a \leq \frac{S_{f,l}}{S_{f_0,m}} \\ 3 + \frac{2}{a(1-a)}, & \text{if } a > \frac{S_{f,l}}{S_{f_0,m}}, \end{cases}$$

с параметрами $1 \leq f_0 \leq f \leq l \leq m$, которые зависят от конфигурации машины и нагрузки.

Теорема 3.4. *Планирование параллельных работ на распределенных системах с идентичными процессорами с использованием алгоритмов MCTa +PS, MLBa +PS и работ с размерами, которые могут быть выполнены на самой маленькой машине, имеет аппроксимационный фактор 3 и конкурентный фактор 5.*

Для оценки алгоритмов на разных инфраструктурах НРС проведен их экспериментальный анализ с использованием рабочих нагрузок, основанных на реальных производственных трассах из архивов параллельных рабочих нагрузок (PWA) (Parallel Workload Archive) и грид-рабочих нагрузок GWA (Grid Workload Archive).

Рассмотрены три сценария Грид. В Grid1 используются 7 существующих вычислительных центров с общим числом 4442 процессоров (KTH, SDSC-SP2, HPC2N, CTC, LANL, SDSC-BLUE, SDSC-DS) с 100, 128, 240, 430, 1024, 1152 и 1368 процессорами, соответственно.

В Grid2 рассматриваются 9 центров, из них 5 центров Grid DAS, а также KTH, HPC2N, CTC и LANL с общим числом 2194 процессоров. Размеры машин составляют 64, 64, 64, 64, 100, 144, 240, 430 и 1024.

В сценарии Grid3 рассматривается гораздо меньший Грид из 11 машин с общим числом 136 процессоров.

Проведена экспериментальная оценка двухэтапных стратегий планирования $MPS = MPS_Alloc + PS$. Разработано девять стратегий распределения MPS_Alloc и девять стратегий распределения с допустимым коэффициентом, включенным в политику планирования. Представлено подробное исследование его влияния, на общую производительность Грид.

Показано, что с точки зрения рассматриваемых критериев, стратегии распределения с допустимым коэффициентом превосходят алгоритмы, которые используют все доступные машины для распределения работ. Такие адаптивные стратегии планирования надежны и стабильны даже в сильно различающихся условиях и способны успешно справляться с различными рабочими нагрузками.

Четвертая глава посвящена задаче распараллеливания, оптимизация которой не всегда эффективна классическими методами из-за разнообразия реальных параллельных и распределенных платформ и/или сред.

Адаптивные алгоритмические схемы, способные динамически изменять распределение работ во время исполнения для оптимизации поведения глобальной системы, являются лучшей альтернативой для решения этой задачи.

В главе рассматривается планирование неclairvoyantных (англ., non-clairvoyant) параллельных работ с известными требованиями к ресурсам, но с неизвестным временем выполнения, фокусируясь на регулирование периодов простоя (idle) при выполнении.

Рассматривается новое семейство стратегий планирования, основанных на двух фазах, которые последовательно сочетают последовательное и параллельное выполнение работ.

Обобщаются известные предельные границы производительности в наихудшем случае (аппроксимационный фактор), учитывая, помимо числа процессоров и максимальных требований к процессору, рассматриваемых в литературе, два дополнительных параметра, а именно: штраф за распараллеливание работ и коэффициент регулирования простоя. Кроме того, доказывається, что регулирование простоя может улучшить аппроксимационный фактор планирования параллельных работ в режиме разделения пространства.

Для нахождения компромисса между сложностью реальных параллельных систем и желаемой простотой их моделей для теоретического изучения рассмотрена модель распараллеливания, основанная на штрафном факторе. Такая модель используется в реальных планировщиках. Идея заключается в том, чтобы добавить к времени идеального параллельного выполнения накладные расходы, которые включают в себя время, потерянное на коммуникацию, синхронизацию, прерывания и любые дополнительные факторы, которые возникают при управлении параллельным выполнением.

Штрафной фактор неявно учитывает некоторые факторы и ограничения при распараллеливании, когда они неизвестны или их очень сложно определить формально. Он скрывает реальную природу и характеристики приложения или параметры компьютера, хотя и известные, но слишком сложные для использования в качестве формальных ограничений планирования. Его можно рассматривать как отношение идеального ускорения исполнения к наблюдаемому.

Штраф за выполнение одной работы или всей рабочей нагрузки также можно оценить на основе эмпирического анализа, сравнительного анализа, расчетов, профилирования, оценки производительности с помощью моделирования, измерения, или на основе информации, предоставленной пользователем.

Рассмотрены несколько качественных моделей штрафа как функции от числа процессоров, выделенных на выполнение работы: константная, линейная и выпуклая. Это наиболее распространенные классы при распараллеливании реальных приложений.

Параллельная работа T_i характеризуется тройкой $T = \{p_i, k_i, \mu_k^i\}$, а именно: временем ее выполнения на одном процессоре (суммарный объем работы), числом запрошенных процессоров k_i , а также штрафным фактором за ее параллельное выполнение на k_i процессорах. Время ее параллельного выполнения $p_k^i = \frac{\mu_k^i p_i}{k_i}$.

Предполагается, что p_k^i не увеличивается при увеличении k_i , по крайней мере, для разумного числа процессоров, и $p_i = p_k^i$. Пусть $\bar{p} = \max_{1 \leq i \leq n} \{p_i\}$, $\bar{p}_{\parallel} = \max_{1 \leq i \leq n} \{p_k^i\}$, $\underline{k} = \max_{1 \leq i \leq n} \{k_i\}$, и $\bar{k} = \max_{1 \leq i \leq n} \{k_i\}$.

Вычисляя ускорение S_k^i при выполнении работы T_i на k процессорах как $\frac{p_1^i}{p_k^i}$, штраф μ_k^i учитывается как отношение идеального ускорения к фактическому $\mu_k^i = \frac{k}{S_k^i}$. Предполагается, что штрафной фактор работы T_i не убывает от числа выделенных процессоров k : $\mu_k^i \leq \mu_{k+1}^i$ для любого $1 \leq k < m$. Если работа назначена на один процессор, то $\mu_k^i = 1$.

Пусть $\bar{\mu} = \max_{1 \leq i \leq n} \{\mu_k^i\}$ и $\underline{\mu} = \max_{1 \leq i \leq n} \{\mu_k^i\}$. Согласно свойству монотонности работ, назначение работе большего числа процессоров уменьшает время ее выполнения, по крайней мере, до определенного порога, но увеличивает ее вычислительную площадь, следовательно

$$\frac{\mu \bar{p}}{\bar{k}} \leq \frac{\bar{p} \bar{\mu}}{\bar{k}} \leq \bar{p}_{\parallel} \leq \frac{\mu \bar{p}}{\underline{k}}$$

Рассмотрено влияние регулирования простоев на общее планирование на основе двухэтапной стратегии. Эта стратегия состоит из двух последовательных фаз. На первой фазе, когда число работ велико, в стратегии распределяются процессоры последовательно без простоев и коммуникаций. Когда достаточное количество работ выполнено, стратегия переходит во вторую фазу с многопроцессорным исполнением. При анализе предполагается, что на первой фазе каждый процессор выполняет одну работу, каждая работа закреплена за одним процессором, таким образом, загрузка процессоров равна 1, а расписание является оптимальным. Неэффективность проявляется, когда остается менее t работ.

Доказана следующая теорема для случая, когда на второй фазе (политика распределения $1 - t$) рассматривается специальный тип расписания, называемый групповым расписанием (gang scheduling).

Теорема 4.1. Для заданного набора из n независимых работ, каждая из которых имеет штрафной коэффициент от $\underline{\mu}$ до $\bar{\mu}$, назначенных на один и/или на t процессоров, гарантии производительности двухфазной стратегии с регулировкой простоев по параметру a могут быть оценены как:

$$\rho_{\parallel}^* = \max\{\rho_{\parallel}^{*1}, \rho_{\parallel}^{*2}\} \quad \text{и} \quad \rho^* = \max\{\rho^{*1}, \rho^{*2}\}, \quad \text{где} \quad \rho_{\parallel}^{*1} \leq \frac{1}{\underline{\mu}} \left(1 + \frac{a-1}{m}\right), \quad \rho_{\parallel}^{*2} \leq \frac{\bar{\mu}}{\underline{\mu}} \left(1 - \frac{a}{m}\right) + \frac{a}{\underline{\mu}t}, \quad \rho^{*1} \leq 1 + \frac{a-1}{m}, \quad \rho^{*2} \leq \bar{\mu} \left(1 - \frac{a}{m}\right) + \frac{a}{m}.$$

Кроме того, ρ_{\parallel}^{*1} и ρ^{*1} достижимы в случае $t \leq \frac{a\bar{\mu}-1}{\bar{\mu}-1}$ и ρ^{*2} достижимы в случае $t > \frac{a\bar{\mu}-1}{\bar{\mu}-1}$.

Показано, что гарантии производительности (аппроксимационный фактор) любого списочного алгоритма для планирования независимых параллельных жестких работ ограничены как:

$$\rho_{\parallel}^* \leq \frac{2m-\bar{k}}{m-\bar{k}+1} \quad \text{и} \quad \rho^* \leq \frac{1}{m-\bar{k}+1} \left(\bar{\mu}t + \frac{\underline{\mu}}{\underline{k}} (m - \bar{k}) \right)$$

Последовательная гарантия производительности любого списочного алгоритма планирования независимых параллельных и последовательных работ ограничена следующим образом: $\rho^* \leq \bar{\mu} \frac{2m-\bar{k}}{m-\bar{k}+1}$.

Теорема 4.2. (Последовательная конкурентность - Sequential competitiveness). Для заданного набора из n независимых параллельных работ с изменением штрафных коэффициентов от $\underline{\mu}$ до $\bar{\mu}$, назначенных на фиксированное количество процессоров от \underline{k} до \bar{k} , последовательный аппроксимационный фактор двухфазной стратегии может быть оценен как:

$$\rho^* = \max\{\rho^{*1}, \rho^{*2}\}, \text{ с } \rho^{*1} \leq 1 + \frac{a-1}{m} \text{ и } \rho^{*2} \leq \frac{a}{m} + \frac{1}{m-\bar{k}+1} \left(\underline{\mu}(m-a) + \frac{\mu}{\underline{k}}(m-\bar{k}) \right)$$

Теорема 4.3. (Последовательная конкурентность) для смеси работ). Для заданного набора n независимых параллельных и последовательных работ с вариациями штрафных факторов от 1 до $\bar{\mu}$, назначенных на фиксированное число процессоров, от 1 до \bar{k} , последовательный аппроксимационный фактор двухфазной стратегии может быть оценен как:

$$\rho^* \leq \bar{\mu} \frac{2m-\bar{k}}{m-\bar{k}+1} - a \left(\frac{\bar{\mu}}{m-\bar{k}+1} - \frac{1}{m} \right).$$

Теорема 4.4. Для заданного набора из n независимых параллельных работ, назначенных на фиксированное число процессоров от \underline{k} до \bar{k} , с минимальным штрафным фактором $\underline{\mu}$, параллельный аппроксимационный фактор двухфазной стратегии может быть оценен как:

$$\rho_{\parallel}^* = \max\{\rho^{*1}, \rho^{*2}\}, \text{ с } \rho_{\parallel}^{*1} \leq 1 + \frac{a-1}{m} \text{ и } \rho_{\parallel}^{*2} \leq \frac{2m-\bar{k}}{m-\bar{k}+1} - a \left(\frac{\bar{\mu}}{m-\bar{k}+1} - \frac{1}{m} \right).$$

Теорема 4.5. Конкурентность для смеси работ. Для заданного набора из n независимых параллельных и последовательных работ, назначенных на фиксированное число процессоров от 1 до \bar{k} , параллельный аппроксимационный фактор двухфазной стратегии может быть оценен как:

$$\rho_{\parallel}^* \leq \frac{2m-\bar{k}}{m-\bar{k}+1} - a \left(\frac{1}{m-\bar{k}+1} - \frac{1}{m} \right).$$

При таком регулировании простоя существует компромисс между более ранним началом распараллеливания, когда a мало (следовательно, распараллеливается больше работ, что приводит к большим накладным расходам на распараллеливание), и задержкой их распараллеливания, когда a большое (что приводит к тому, что больше процессоров остаются без работы), до тех пор, пока не будет доступно меньшее число работ. Эта схема балансирует потребности пользователя (работ) с потребностями компьютерной системы.

В пятой главе рассматриваются проблемы планирования виртуализированных вычислительных ресурсов, предоставляемых пользователям через Интернет в форме облачного сервиса (Infrastructure as a Service - IaaS). В типичном сценарии IaaS поставщик инфраструктуры предлагает свои ресурсы по требованию и с различными уровнями сервиса своим клиентам. Эти уровни сервиса в основном отличаются количеством вычислительной мощности, которое заказчик гарантированно получит в течение определенного периода времени.

Для формализации уровня сервиса вводится слак-фактор и цена за единицу времени обработки. Если провайдер принимает работу, она гарантированно выполняется к установленному директивному сроку, который определяется как

время ее подачи плюс время обработки, умноженное на слак-фактор заданного уровня обслуживания. После того, как работа была получена, провайдер должен немедленно решить, принимает ли он эту работу или отказывается от нее.

Предлагаются различные алгоритмы и приводится конкурентный анализ для обсуждения различных сценариев для данной модели. Эти сценарии объединяют фиксированные уровни обслуживания с одной или несколькими машинами. Демонстрируется, как можно достигнуть лучшего конкурентного фактора.

Уровень обслуживания S_i связан со слак-фактором $f_i > 1$ и ценой единицы времени обработки u_i . Поскольку работы поступают со временем $(r_{j,online})$, время обработки p_j и запрашиваемый уровень обслуживания S_i работы J_j становятся известны только в момент ее поступления $r_j \geq 0$. Работа J_j имеет крайний срок выполнения $d_j = r_j + f_i \cdot p_j$, который зависит от слак-фактора или стрейч-фактора f_i уровня обслуживания S_i .

Принятая работа с f_i будет получать в среднем не менее $1/f_i$ ресурсов машины в интервале времени от поступления до завершения работы. Чтобы сохранить ранее предоставленные гарантии уровня обслуживания, работа J_j может быть принята только в том случае, если существует такое расписание, что ни J_j , ни какая-либо ранее принятая работа не выходят за пределы своих сроков, т.е. для всех принятых работ J_j выполняется $C_i \leq d_i$, при этом C_i – это время окончания работы J_j .

Целевая функция представляет собой функцию поставщика инфраструктуры, который хочет максимизировать свой общий доход. Работа J_i с уровнем обслуживания S_i генерирует доход $v_j = x_j \cdot p_j \cdot u_i$. Двоичная переменная $x_j \in \{0, 1\}$ обозначает, принимается ли работа J_j ($x_j = 1$) или нет ($x_j = 0$).

Используя обозначение трех полей, проблема описывается либо как $1|prmp, r_{j,online}, S_i|\sum v_j$ либо как $P_m|prmp, r_{j,online}, S_i|\sum v_j$ в зависимости от выбранной модели машины. Проблема требует максимизации целевой функции.

Для SSL-SM, установлена верхняя граница для конкурентного фактора.

Теорема 5.1. $c_V \leq 1 - \left(1 - \frac{p_{min}}{p_{max}}\right) \cdot \frac{1}{f_1}$ справедливо для SSL-SM с сервисным уровнем S_1 .

Для SSL-SM, используется простой алгоритм, который называется жадным принятием. Этот алгоритм принимает каждую новую работу, если эта работа и все ранее принятые работы могут быть выполнены вовремя.

Теорема 5.2. Жадное принятие имеет конкурентный фактор $1 - \frac{1}{f_1}$ для SSL-SM с уровнем обслуживания S_1 .

Теорема 5.3. $c_V \leq 1 - \frac{1}{f_I}$ справедливо для SSL-PM с уровнем обслуживания S_I , если разность между самым большим и наименьшим временем обработки работы может быть произвольно большой.

Теорема 5.4. $c_V \leq \frac{f_I}{1+f_I \cdot (1-\frac{p_{\min}}{p_{\max}})}$ справедливо для SSL-PM с уровнем обслуживания S_I и слак-фактором $f_I < \frac{p_{\max}}{p_{\min}}$

Теорема 5.5. Жадное принятие имеет конкурентный фактор $1 - \frac{1}{f_I}$ для SSL-PM с сервисным уровнем S_I .

Теорема 5.6. $c_V \leq \max \left\{ \frac{\frac{p_{\min}}{p_{\max}}}{(f_I-1)}, \frac{f_I-1+\frac{p_{\min}}{p_{\max}}}{f_I-1+\frac{u_I}{u_{II}}} \right\}$ справедливо для MSL-SM с двумя уровнями обслуживания S_I и S_{II} и $f_{II} < 2$.

Теорема 5.7. Жадный алгоритм принятия имеет конкурентный фактор $\frac{u_{II}}{u_I} \left(1 - \frac{1}{f_I}\right)$ для MSL-SM с уровнями обслуживания S_I и S_{II} .

Рассмотрен алгоритм под названием ограниченное принятие для MSL-SM. Этот алгоритм использует коэффициент принятия $h_{II} \geq 1$ и принимает работу J_j с уровнем обслуживания S_{II} только в том случае, если $d_i - C_i \geq h_{II} \cdot p_i$ для всех работ J_i с уровнем обслуживания S_{II} и $d_i \geq d_j$. Здесь C_i обозначает время завершения работы J_i в EDD-расписании с прерываниями после принятия работы J_i .

Теорема 5.8. Ограниченное принятие имеет конкурентный фактор $\min \left\{ \frac{h_{II}-1}{f_{II}}, \frac{f_I-1}{f_I} - \frac{h_{II}}{f_{II}} \cdot \left(1 - \frac{u_{II}}{u_I}\right) \right\}$ для MSL-SM с уровнями обслуживания S_I и S_{II} , и $h_{II} < f_{II} \cdot \left(1 - \frac{1}{f_I}\right)$.

Рассмотрен сценарий MSL-PM с различными уровнями обслуживания и параллельными машинами. MSL-PM близко соответствует реальным требованиям стандартов IaaS, которые, как правило, используют множество машин для поддержки масштабирования. Чтобы эффективно использовать большое количество машин, поставщик IaaS нуждается во многих независимых клиентах. Для повышения гибкости предлагаются различные уровни обслуживания.

Теорема 5.9. Жадное принятие имеет конкурентный фактор $\frac{u_{II}}{u_I} \left(1 - \frac{1}{f_I}\right)$ для MSL-PM с уровнями обслуживания S_I и S_{II} .

Некоторым машинам можно дать более высокий приоритет в обслуживании, чтобы получить лучший результат. Точнее, работы с уровнем обслуживания S_I могут быть распределены на каждую машину, в то время как работы с уровнем обслуживания S_{II} могут быть распределены только на

некоторые машины, т.е. используются ограничения допустимости для машин с так называемой иерархической топологией сервера или уровнем обслуживания. Этот алгоритм использует допустимое принятие. Аналогичный подход также использовался в контексте планирования параллельных работ в ГРИД [17].

Теорема 5.10. *Допустимое принятие имеет конкурентный фактор*

$$\frac{1 - \frac{1}{f_{II}}}{1 + \frac{\frac{1}{f_{II}} u_{II}}{1 - \frac{1}{f_I}} u_I}$$

для *MSL-PM* с уровнями сервисов S_I и S_{II} .

Конкурентный фактор допустимого принятия больше, чем конкурентный фактор обычного жадного принятия.

Шестая глава рассматривает модификацию модели IaaS, проанализированную в главе 5. В отличие от предыдущей главы, где рассмотрены сценарии с одним или двумя уровнями обслуживания, здесь анализируются более реалистичные сценарии, где для клиентов предоставляются несколько уровней обслуживания $SL = [SL^1, SL^2, \dots, SL^l, \dots, SL^k]$, а также рассматривается двухкритериальная оптимизация: увеличение дохода провайдера и снижение энергопотребления.

Для данного SL^l работа J_j требует производительность s_j^l , которая гарантируется предоставлением соответствующей виртуальной машины, и взимается стоимость u_j^l за единицу времени выполнения в зависимости от требуемой срочности. Каждая работа J_j описывается кортежем (r_j, w_j, d_j, SL_j^l) , содержащим время выпуска r_j , объем работы w_j , описывающий вычислительную нагрузку, которая должна быть выполнена до требуемого времени ответа, директивный срок выполнения d_j и уровень обслуживания $SL_j^l \in SL$.

Хотя большое количество уровней приводит к высокой гибкости для клиентов, оно также приводит к значительным накладным расходам на планирование. В главе предложен метод поиска подходящего компромисса.

Рассмотрены гетерогенные машины $M = [M_1, M_2, \dots, M_m]$, каждая из которых описывается кортежем (s_i, eff_i) , указывающим ее относительную скорость обработки s_i и ее энергоэффективность eff_i .

Используется двухуровневый подход к планированию. На верхнем уровне система проверяет, может ли работа быть принята или нет, используя политику принятия Greedy. Если работа принята, то система выбирает машину из набора допустимых машин для ее выполнения на нижнем уровне.

Предложены восемь стратегий планирования. Они характеризуются типом и объемом информации, используемой для принятия решения и делятся на три

группы: i) без знаний о системе, без информации о работах и ресурсах; ii) с информацией об энергопотреблении; iii) с информацией о скорости машин.

Экспериментальный анализ проведен с использованием работ НРС, полученных из архива параллельных рабочих нагрузок (PWA) и архива рабочих нагрузок Grid (GWA). Эти трассы являются логами реальных параллельных компьютерных систем и дают хорошее представление о том, как предлагаемые схемы будут работать с реальными пользователями.

Проведен совместный анализ двух критериев сначала на основе ухудшения производительности каждой стратегии по каждой метрике; затем на основе Парето-фронта.

Предложена стратегия, которая превосходит другие алгоритмы. Она доминирует почти во всех тестах. Стратегия стабильна даже в значительно отличающихся условиях. Она обеспечивает незначительное снижение производительности и справляется с различными требованиями. Она использует минимальную информацию, что важно в нестационарной среде, и характеризуется небольшой вычислительной сложностью; тем не менее, достигает хорошего улучшения критериев и обеспечивает гарантии качества обслуживания.

В седьмой главе формулируются и исследуются проблемы планирования, связанные с облачными системами VoIP. Они учитывают динамическую нагрузку, вариативность времени запуска VM, свойства работ, инфраструктуры, наличия других пользователей, которые совместно используют сервис, и т. д.

Введена CVoIP инфраструктура, которая состоит из m разнородных кластеров узлов связи с различными скоростями обработки вызовов. Каждый узел запускает виртуальные машины, которые описываются началом аренды, временем их развертывания, и вычислительной мощностью.

Рабочая нагрузка состоит из независимых вызовов, которые должны быть спланированы на множестве узлов. Вызов описывается кортежем, который состоит из времени его начала, продолжительности и вклада в загрузку процессора из-за используемого кодека и типа (голосовой почты, видео/аудио конференций, интерактивных телефонных меню, передачи изображений и текста, перераспределения вызовов и т. д.). Время появления вызова недоступно до его начала, а его продолжительность неизвестна до завершения вызова.

В дополнение к описанным выше параметрам VM характеризуется загруженностью (утилизацией) в момент времени t , которая зависит от используемого кодека, типа вызова, других пользователей и вычислительной мощности VM. Каждый кодек предоставляет определенное качество голоса

только при достаточно низкой загрузке процессора. Теоретически, загрузка процессора на 100% обеспечивает наилучшую ожидаемую производительность. Однако при загрузке уже до 85% процессор не справляется с нагрузкой, появляются дрожание и разрывы звука.

Введена модель стоимости как функция, зависящая от числа арендуемых VM, и времени их аренды. А также функция снижения качества сервиса в зависимости от загруженности VM. Проанализирован дополнительный критерий качества, связанный с числом звонков, находящихся в режиме ожидания (on hold).

Сформулированная задача рассматривается как частный случай динамической упаковки в контейнеры. Контейнеры представляют собой виртуальные машины, а высота элементов определяет вклад вызова в загрузку виртуальной машины. Планировщик знает только вклад вызова в загрузку VM. Все решения принимаются без информации о продолжительности вызова, скорости поступления вызовов и т. д.

Принципиальной новизной данной проблемы является временное существование элементов (вызовов) в упаковке контейнера. В отличие от стандартной формулировки, контейнеры всегда открыты и динамичны, даже если они полностью заполнены. Элементы в контейнерах могут быть удалены (завершение вызова), а утилизация VM может быть уменьшена в любой момент времени, тогда VM могут использовать свободное пространство для обработки новых вызовов.

Для адаптации к разнообразным гетерогенным средам и вызовам разработаны три настраиваемых параметра: *Utilization Threshold* (UT) порог утилизации, при достижении которого, процессор не может справиться с обработкой звонков и могут появиться джиттеры и прерывистый звук. *Порог аренды* (RT) – это интервал времени, оставшийся до окончания периода аренды VM. Он позволяет избежать ситуации, когда вызов поступает незадолго до окончания времени аренды и завершается после окончания периода аренды, что приводит к продолжению аренды VM с дополнительной оплатой еще на один час, хотя другие VM свободны. *Интервал прогнозирования* (PI - Prediction interval) — это интервал времени, на который выполняется прогнозирование. Предиктор выполняет оценку утилизации на каждом временном шаге (TS – Time Step). Меньший TS обеспечивает лучшее предсказание, но увеличивает накладные расходы системы. Наоборот, больший TS уменьшает накладные расходы системы, но снижает точность предсказания.

Параметры могут быть настроены и динамически адаптированы к различным предпочтениям, рабочим нагрузкам и свойствам облака.

Результаты экспериментов на реальных данных компании MiXvoip (Люксембург) показывают, что предложенные алгоритмы с методами

прогнозирования нагрузки превосходят известные стратегии, обеспечивая высокое качество обслуживания и более низкую стоимость и могут быть эффективно использованы в облачной среде VoIP.

ЗАКЛЮЧЕНИЕ

Работа носит теоретический характер. Полученные в ней результаты позволяют изучить различные аспекты планирования работ в распределенных гетерогенных вычислительных системах в условиях нестационарности, и исследовать теоретические оценки границ оптимизации.

Заявленные и представленные выше результаты можно сгруппировать следующим образом:

- Сформулирована проблема планирования работ в гетерогенной распределенной вычислительной системе в условиях ее нестационарности для различных сценариев и предложены подходы к ее решению путем динамической адаптации планировщика к различным рабочим нагрузкам, изменениям конфигурации системы и ее параметров.
- Разработаны алгоритмы с гарантированным конкурентным фактором для онлайн случая и аппроксимационным фактором для оффлайн случая с использованием подхода динамической "кражи работ" (англ. "job stealing").
- Разработана двухуровневая модель гетерогенной распределенной вычислительной системы, которая интегрирует две задачи планировщика: распределение параллельных работ по машинам и локального планирования. Решения по планированию принимаются без точной информации о производительности машин и времени выполнения работ.
- Для снижения влияния неполноты информации и эффективного решения соответствующей проблемы оптимизации, введен и исследован коэффициент допустимости (англ. "admissible factor"), включенного в политику планирования. Показано как он может быть динамически скорректирован для того, чтобы справиться с непредсказуемой рабочей нагрузкой улучшая конкурентный фактор. Получены теоретических оценки конкурентных и аппроксимационных факторов алгоритма улучшая и расширяя известные результаты.
- Разработана модель планирования с регулированием периодов простоя машин. Получены новые и обобщены известные предельные границы производительности планирования в наихудшем случае.

- Разработаны модели планирования облачных вычислений на основе уровней обслуживания. Проведен конкурентный анализ алгоритмов для различных сценариев с одним или несколькими уровнями обслуживания, с одной или несколькими машинами, а также с жадным и ограниченным принятием работ.
- Сформулирован, разработан и исследован облачный VoIP планировщик, который учитывает динамическую рабочую нагрузку, вариативность времени запуска VM, свойства вызовов, инфраструктуры, и т. д. Он адаптируется к поведению VM из-за числа и типа вызовов (голосовой почты, видео/аудио конференций, передачи изображений и текста, и т. п.) для предотвращения снижения качества сервиса.
- Задача планирования рассмотрена как частный случай динамической упаковки в контейнеры. Временное существование элементов (вызовов) при упаковке является принципиальной новизной данной проблемы. В отличие от стандартной формулировки, контейнеры всегда открыты, даже если они полностью заполнены, так как элементы в контейнерах могут быть удалены (завершение вызова). Экспериментальный анализ на реальных данных компании MiXvoip (телекоммуникационный и Интернет-провайдер, Люксембург) показал, что предложенные алгоритмы с методами прогнозирования нагрузки и разработанными настраиваемыми параметрами, превосходят известные стратегии, обеспечивая высокое качество обслуживания и более низкую стоимость и могут быть эффективно использованы в облачной среде VoIP.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи автора в журналах, рекомендованных ВАК РФ, Scopus, Web of Science

1. Armenta-Cano, F. A., Tchernykh, A., Cortés-Mendoza, J. M., Yahyapour, R., Drozdov, A. Y., Bouvry, P., Kliazovich, D., Avetisyan, A., Nasmachnow, S. Min_c: Heterogeneous concentration policy for energy-aware scheduling of jobs with resource contention // *Programming and Computer Software*. – 2017. – Т. 43. – №. 3. – С. 204-215.
2. Cortés-Mendoza, J. M., Tchernykh, A., Simionovici, A. M., Bouvry, P., Nasmachnow, S., Dorransoro, B., & Didelot, L. VoIP service model for multi-objective scheduling in cloud infrastructure // *International Journal of Metaheuristics*. – 2015. – Т. 4. – №. 2. – С. 185-203.
3. Cortés-Mendoza, J.M.; Tchernykh, A.; Armenta-Cano, F.A.; Bouvry, P.;

- Drozdov, A.Y.; Didelot, L. Biobjective VoIP service management in cloud infrastructure // *Scientific Programming*. – 2016. – T. 2016.
4. Cristobal-Salas A, Tchernykh, A., Gaudiot, J. L., & Lin, W. Y. Non-strict execution in parallel and distributed computing // *International Journal of Parallel Programming*. – 2003. – T. 31. – №. 2. – C. 77-105.
 5. Díaz A. R., Tchernykh A., Ecker K. H. Algorithms for dynamic scheduling of unit execution time tasks // *European Journal of Operational Research*. – 2003. – T. 146. – №. 2. – C. 403-416.
 6. Hiraes-Carbajal, A., Tchernykh, A., Yahyapour, R., González-García, J. L., Röblitz, T., & Ramírez-Alcaraz, J. M. Multiple workflow scheduling strategies with user run time estimates on a grid // *Journal of Grid Computing*. – 2012. – T. 10. – №. 2. – C. 325-346.
 7. Ivashko, E. E., Ivashko, A. A., Safonov, G. R., Tchernykh, A. Cost-Efficient Strategy in Clouds with Spot Price Uncertainty // *Automation & Remote Control*. – 2020. – T. 81. – №. 4.
 8. Kliazovich, D., Pecero, J. E., Tchernykh, A., Bouvry, P., Khan, S. U., & Zomaya, A. Y. CA-DAG: Modeling communication-aware applications for scheduling in cloud computing // *Journal of Grid Computing*. – 2016. – T. 14. – №. 1. – C. 23-39.
 9. Ramírez-Alcaraz, J. M., Tchernykh, A., Yahyapour, R., Schwiegelshohn, U., Quezada-Pina, A., González-García, J. L. Hiraes-Carbajal, A. Job allocation strategies with user run time estimates for online scheduling in hierarchical grids // *Journal of Grid Computing*. – 2011. – T. 9. – №. 1. – C. 95-116.
 10. Ramírez-Velarde, R., Tchernykh, A., Barba-Jimenez, C., Hiraes-Carbajal, A., Nolazco-Flores, J. Adaptive resource allocation with job runtime uncertainty // *Journal of Grid Computing*. – 2017. – T. 15. – №. 4. – C. 415-434.
 11. Tchernykh A., Ecker K. Worst case behavior of list algorithms for dynamic scheduling of non-unit execution time tasks with arbitrary precedence constraints // *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*. – 2008. – T. 91. – №. 8. – C. 2277-2280.
 12. Tchernykh, A.; Babenko, M.; Chervyakov, N.; Miranda-Lopez, V.; Avetisyan, A.; Drozdov, A.Y.; Rivera-Rodriguez, R.; Radchenko, G.; Du, Z. Scalable data storage design for nonstationary IoT environment with adaptive security and reliability // *IEEE Internet of Things Journal*. – 2020. – T. 7. – №. 10. – C. 10171-10188.
 13. Tchernykh, A.; Bychkov, I.; Feoktistov, A.; Gorsky, S.; Sidorov, I.; Kostromin, R.; Edelev, A.; Zorkalzev, V.; Avetisyan, A. Mitigating Uncertainty in Developing and Applying Scientific Applications in an Integrated Computing Environment // *Programming and Computer Software*. – 2020. – T. 46. – №. 8. –

- C. 483-502.
14. Tchernykh, A.; Cortés-Mendoza, J.M.; Bychkov, I.; Feoktistov, A.; Didelot, L.; Bouvry, P.; Radchenko, G.; Borodulin, K. Configurable cost-quality optimization of cloud-based VoIP // *Journal of Parallel and Distributed Computing*. – 2019. – T. 133. – C. 319-336.
 15. Tchernykh, A.; Lozano, L.; Schwiegelshohn, U.; Bouvry, P.; Pecero, J.E.; Nesmachnow, S.; Drozdov, A.Y. Online Bi-Objective Scheduling for IaaS Clouds Ensuring Quality of Service // *Grid Computing*. – 2015.
 16. Tchernykh, A.; Pecero, J.E.; Barrondo, A.; Schaeffer, E. Adaptive energy efficient scheduling in Peer-to-Peer desktop grids // *Future Generation Computer Systems*. – 2014. – T. 36. – C. 209-220
 17. Tchernykh, A.; Schwiegelshohn, U.; On-line hierarchical job scheduling on grids with admissible allocation // *Journal of Scheduling*. – 2010. – T. 13. – №. 5. – C. 545-552.
 18. Tchernykh, A.; Schwiegelshohn, U.; Talbi, E.; Babenko, M. Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability // *Journal of Computational Science*. – 2019. – T. 36. – C. 100581.
 19. Tchernykh, A.; Stepanov, A.; Lupenko, A.; Tchernykh, N. Abstract Network Machine: parallel computation on associative networks with incomplete data structures // *Dyna*. – 2003. – T. 70. – №. 140. – C. 71-88.
 20. Tchernykh, A.; Trystram, D.; Brizuela, C.; Scherson, I. Idle regulation in non-clairvoyant scheduling of parallel jobs // *Discrete Applied Mathematics*. – 2009. – T. 157. – №. 2. – C. 364-376.
 21. Quezada-Pina, A., Tchernykh, A., González-García, J. L., Hiraes-Carbajal, A., Ramírez-Alcaraz, J. M., Schwiegelshohn, U., Yahyapour, R., & Miranda-López, V. Adaptive parallel job scheduling with resource admissible allocation on two-level hierarchical grids // *Future Generation Computer Systems*. – 2012. – T. 28. – №. 7. – C. 965-976.

Другие публикации автора по теме диссертации

22. Armenta-Cano, F.; Tchernykh, A.; Cortés-Mendoza J, M.; A., A.; Yahyapour, D.; Drozdov A, Y.; Bouvry P, K. Heterogeneous job consolidation for power aware scheduling with quality of service. // *In Proceedings of the Russian Conference on Supercomputing* – 2015. – C. 687-697.
23. Armenta-Cano, F., Tchernykh, A., Yahyapour, R., & Nabrzyski, J. Cost Optimization of Virtual Machine Provisioning in Federated IaaS Clouds // *5th International Conference on Supercomputing in Mexico*. – 2015.

24. Babenko, M., Tchernykh, A., Chervyakov, N., Murga, D., Dvoryaninova, I., Miranda-López, V., Kucherov, N. Multi-Clouds Workload Distribution for the Secure and Reliable Storage of Data under Uncertainty //Applications”. Irkutsk: ESI SB RAS, 2017, 224 p. – 2017. – C. 30.
25. Barrondo, A., Tchernykh, A., Schaeffer, E., & Pecero, J. Energy efficiency of knowledge-free scheduling in peer-to-peer desktop grids //2012 International Conference on High Performance Computing & Simulation (HPCS). – IEEE, 2012. – C. 105-111.
26. Canosa, R., Tchernykh, A., Cortés-Mendoza, J. M., Rivera-Rodriguez, R., Rizk, J. L., Avetisyan, A., Du Z., Radchenko G., Morales, E. R. C. Energy consumption and quality of service optimization in containerized cloud computing // 2018 Ivannikov Ispras Open Conference (ISPRAS). – IEEE, 2018. – C. 47-55.
27. Combarro, M., Tchernykh, A., Kliazovich, D., Drozdov, A., & Radchenko, G. Energy-aware scheduling with computing and data consolidation balance in 3-tier data center //2016 International Conference on Engineering and Telecommunication (EnT). – IEEE, 2016. – C. 29-33.
28. Cortés-Mendoza, J. M., Tchernykh, A., Bychkov, I., Feoktistov, A., Bouvry, P., Didelot, L. Load-aware strategies for cloud-based VoIP optimization with VM startup prediction // 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). – IEEE, 2017. – C. 472-481.
29. Cortés-Mendoza, J. M., Tchernykh, A., Drozdov, A. Y., Didelot, L. Robust cloud VoIP scheduling under VMs startup time delay uncertainty // 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC). – IEEE, 2016. – C. 234-239.
30. Cortés-Mendoza, J. M., Tchernykh, A., Drozdov, A., Bouvry, P., Simionovici, A. M., Kliazovich, D., Avetisyan, A. Distributed adaptive VoIP load balancing in hybrid clouds // Russian Supercomputing Days 2015. – 2015.
31. Cortés-Mendoza, J. M., Tchernykh, A., Radchenko, G., Drozdov, A. Y. RoC Prediction for Bi-Objective Cost-QoS Optimization of Cloud VoIP Call Allocations //2017 IVth International Conference on Engineering and Telecommunication (EnT). – IEEE, 2017. – C. 119-123.
32. Cristóbal-Salas A., Tchernykh A., Gaudiot J. L. Non-strict evaluation of the FFT algorithm in distributed memory systems // European Parallel Virtual Machine/Message Passing Interface Users’ Group Meeting. – Springer, Berlin, Heidelberg, 2003. – C. 188-195.
33. Cristóbal-Salas, A., Chernykh, A., Rodríguez-Alcantar, E., & Gaudiot, J. L. Exploiting single-assignment properties to optimize message-passing programs by code transformations // Symposium on Implementation and Application of Functional Languages. – Springer, Berlin, Heidelberg, 2004. – C. 1-16.

34. Galaviz-Alejos, L. A., Armenta-Cano, F., Tchernykh, A., Radchenko, G., Drozdov, A. Y., Sergiyenko, O., & Yahyapour, R. Bi-objective heterogeneous consolidation in cloud computing // Latin American High Performance Computing Conference. – Springer, Cham, 2017. – C. 384-398.
35. Garibay Martinez, R.; Tchernykh, A.; Ecker, K.H. Re allocation Cost Minimization in Distributed Dynamic Real Time Systems. // In Proceedings of the 16th International Conference on Real-Time and Network Systems. – 2008. – C. 49–52.
36. Gómez, C. E., Chavarriaga, J., Tchernykh, A., Castro, H. E. Improving reliability for provisioning of virtual machines in desktop clouds //European Conference on Parallel Processing. – Springer, Cham, 2019. – C. 669-680.
37. Hiraes-Carbajal, A., Tchernykh, A., Röblitz, T., Yahyapour, R. A Grid simulation framework to study advance scheduling strategies for complex workflow applications // 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW). – IEEE, 2010. – C. 1-8.
38. Iturriaga, S., Nesmachnow, S., Tchernykh, A., and Dorronsoro, B. The processor working set and its Multiobjective workflow scheduling in a federation of heterogeneous green-powered data centers //2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). – IEEE, 2016. – C. 596-599.
39. Kliazovich, D., Pecero, J. E., Tchernykh, A., Bouvry, P., Khan, S. U., Zomaya, A. Y. CA-DAG: Communication-aware directed acyclic graphs for modeling cloud computing applications // 2013 IEEE sixth international conference on cloud computing. – IEEE, 2013. – C. 277-284.
40. Lopez-Falcon, E. C., Miranda-López, V., Tchernykh, A., Babenko, M., Avetisyan, A. Bi-objective analysis of an adaptive secure data storage in a multi-cloud // Latin American High Performance Computing Conference. – Springer, Cham, 2018. – C. 307-321.
41. Miranda V., Tchernykh A., Kliazovich D. Dynamic communication-aware scheduling with uncertainty of workflow applications in clouds //International Conference on Supercomputing in Mexico. – Springer, Cham, 2015. – C. 169-187.
42. Schwiegelshohn U., Tchernykh A. Online scheduling for cloud computing and different service levels // 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum. – IEEE, 2012. – C. 1067-1074.
43. Schwiegelshohn U., Tchernykh A., Yahyapour R. Online scheduling in grids //2008 IEEE International Symposium on Parallel and Distributed Processing. –

- IEEE, 2008. – C. 1-10.
44. Tchernykh A. From Static Scheduling Towards Understanding Uncertainty // Algorithms and Scheduling Techniques to Manage Resilience and Power Consumption in Distributed Systems. – 2016. – T. 5. – № 7. – C. 19. (Reports)
 45. Tchernykh A. Idle Regulation in On-Line Scheduling of Multiprocessor Jobs // International Seminar on Scheduling in Computer and Manufacturing Systems. 04231, Dagstuhl, Germany – 2004. (abstracts).
 46. Tchernykh A. Multi criteria optimization with Quality of Service in Cloud Computing // CLOUDCOMS - Eastern Europe-Luxembourg Workshop on Cloud Computing Communications, Security and Services. The 3rd IEEE International Conference on Cloud Networking (IEEE CloudNet 2014). – 2014.
 47. Tchernykh A. Online Scheduling for Cloud Computing and Different Service Levels and Quality of Service // New Challenges in Scheduling Theory. // Marseille-Lumini-CIRM – 2014. (book of abstracts).
 48. Tchernykh A. Online Scheduling in Grids. New Challenges in Scheduling Theory // Marseille-Lumini-CIRM. – 2008. (book of abstracts).
 49. Tchernykh A., Lozano Rizk J. Big Data-aware Scheduling with Uncertainty in Cloud Computing // Big Network Workshop, CENIC - Corporation for Education Network Initiatives in California. – 2013.
 50. Tchernykh A., Trystram D. Adaptive strategy for on-line scheduling of real world parallel applications //SOCAL. Southern California Workshop on Parallel and Distributed Processing and Architecture (book of abstracts), Santa Barbara, USA, – 2002. – T. 2. – C. 7-8.
 51. Tchernykh A., Trystram D. Online scheduling of multiprocessor jobs with idle regulation //International Conference on Parallel Processing and Applied Mathematics. – Springer, Berlin, Heidelberg, 2003. – C. 131-144.
 52. Tchernykh A., Trystram D., Rapine C. Adaptive (a, b, c)-scheme strategy for on-line scheduling //New Trends in Scheduling in Parallel and Distributed Systems, CIRM, Luminy, Marseille, France. – 2001.
 53. Tchernykh, A., Babenko, M., Kuchukov, V., Miranda-López, V., Avetisyan, A., Rivera-Rodriguez, R., & Radchenko, G. Data reliability and redundancy optimization of a secure multi-cloud storage under uncertainty of errors and falsifications //2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). – IEEE, 2019. – C. 565-572.
 54. Tchernykh, A., Cortés-Mendoza, J. M., Pecero, J. E., Bouvry, P., & Kliazovich, D. Adaptive energy efficient distributed VoIP load balancing in federated cloud infrastructure //2014 IEEE 3rd International Conference on Cloud Networking (CloudNet). – IEEE, 2014. – C. 27-32.
 55. Tchernykh, A., Facio-Medina, A., Pulido-Gaytan, B., Rivera-Rodriguez, R.,

- Cortés-Mendoza, J. M., Radchenko, G., Babenko M., Chernykh I., Kulikov I., Neschachnow, S. Toward digital twins' workload allocation on clouds with low-cost microservices streaming interaction //2020 Ivannikov Ispras Open Conference (ISPRAS). – IEEE, 2020. – C. 115-121.
56. Tchernykh, A., Lozano, L., Bouvry, P., Pecero, J. E., Schwiegelshohn, U., & Neschachnow, S. Energy-aware online scheduling: ensuring quality of service for IaaS clouds //2014 International Conference on High Performance Computing & Simulation (HPCS). – IEEE, 2014. – C. 911-918.
57. Tchernykh, A., Lozano, L., Schwiegelshohn, U., Bouvry, P., Pecero, J. E., Neschachnow, S. Bi-objective online scheduling with quality of service for IaaS clouds // 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet). – IEEE, 2014. – C. 307-312.
58. Tchernykh, A., Ramírez, J. M., Avetisyan, A., Kuzjurin, N., Grushin, D., Zhuk, S. Two level job-scheduling strategies for a computational grid // PPAM International Conference on Parallel Processing and Applied Mathematics. – Springer, Berlin, Heidelberg, 2005. LNCS 3911, Springer-Verlag, 2006. – C. 774-781.
59. Tchernykh, A., Schwiegelshohn, U., Yahyapour, R., Kuzjurin, N. Online hierarchical job scheduling on Grids //From Grids to Service and Pervasive Computing. – Springer, Boston, MA, 2008. – C. 77-91.
60. Tchernykh, A., Schwiegelsohn, U., Alexandrov, V., & Talbi, E. G. Towards understanding uncertainty in cloud computing resource provisioning //Procedia Computer Science. – 2015. – T. 51. – C. 1772-1781.
61. Tchernykh, A., Schwiegelsohn, U., Alexandrov, V., Talbi, E. G. Uncertainty in clouds: Challenges of efficient resource provisioning // Russian Conference on Supercomputing – 2015. – C. 698-699.
62. Yaurima-Basaldua, V. H., Tchernykh, A., Castro-Garcia, Y., Villagomez-Ramos, V. M., & Burtseva, L. Genetic algorithm calibration for two objective scheduling parallel jobs on hierarchical Grids //International Conference on Parallel Processing and Applied Mathematics. – Springer, Berlin, Heidelberg, 2011. – C. 61-70.
63. Zhuk, S., Chernykh, A., Avestiyan, A., Gaissaryan, S., Kuzjurin, N., Pospelov, A., & Grushin, D. Comparison of scheduling heuristics for grid resource broker //Proceedings of the Fifth Mexican International Conference in Computer Science, 2004. ENC 2004. – IEEE, 2004. – C. 388-392.
64. Патент № 2744815 Российская Федерация, МПК G06F 7/72. Устройство для перевода чисел из системы остаточных классов и расширения оснований: № 2020120649; заявл. 22.06.2020; опубл. 16.03.2021 / Бабенко М.Г., Кучуков В.А., Черных А.Н., Кучеров Н.Н.; заявитель и патентообладатель

Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». – 13 с.

65. Свидетельство о государственной регистрации программы для ЭВМ №2017660880, дата регистрации 28.09.2017 «Программа моделирования работы устройств концепции интернет вещей на базе системы остаточных классов»;
66. Свидетельство о государственной регистрации программы для ЭВМ №2018612694, дата регистрации 21.02.2018 «Модуль оценки рисков безопасности облачных, краевых и туманных вычислений в условиях вычислительной неопределенности»;
67. Свидетельство о государственной регистрации программы для ЭВМ №2019610639, дата регистрации 15.01.2019 «Среда моделирования адаптивной цифровой фильтрации в системе остаточных классов»;
68. Свидетельство о государственной регистрации программы для ЭВМ №2019611375, дата регистрации 24.01.2019 «Распределенная система надежного хранения и обработки данных в мультиоблачной среде».