

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение
высшего образования

**«Сибирский государственный университет науки и технологий
имени академика М.Ф. Решетнева»**

На правах рукописи

Брюханова Евгения Романовна

**ГИБРИДНЫЙ МЕТОД УПРАВЛЕНИЯ РЕСУРСАМИ В
РАСПРЕДЕЛЕННЫХ ДИНАМИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ
СИСТЕМАХ**

2.3.1 – Системный анализ, управление и обработка информации, статистика

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук,
доцент Антамошкин О.А.

Красноярск – 2023

ОГЛАВЛЕНИЕ

Введение.....	4
1 Основные особенности задачи управления ресурсами в распределенных динамических вычислительных системах	13
1.1 Особенности распределенных вычислительных динамических систем.....	13
1.2 Обзор методов управления ресурсами и их применение в распределенных динамических вычислительных системах	17
Выводы к Главе 1	25
2 Моделирование и анализ распределенной динамической вычислительной системы	26
2.1 Описание структуры распределенной динамической вычислительной системы и её характеристики.....	26
2.2 Разработка математической модели для динамического поведения распределенной вычислительной системы.....	28
2.3 Имитационное моделирование распределенных динамических вычислительных систем и анализ факторов, влияющих на энергопотребление и производительность	32
2.4 Гибридный подход к управлению ресурсами распределенных динамических вычислительных систем	41
2.4.1 Планирование задач инструментами миграции, репликации, задержки ..	41
2.4.2 Динамическое управление частотой и напряжением	46
2.4.3 Подход к управлению ресурсами распределенных динамических вычислительных систем	49
Результаты Главы 2	51
3 Гибридный метод управления ресурсами в распределенных динамических системах	53

3.1	Исследование методов решения поставленной задачи.....	53
3.2	Алгоритм гибридного подхода на базе обнуляющих нейронных сетей.....	60
3.2.1	Структура нейронных сетей обнуления для гибридного метода	62
3.2.2	Подготовка данных.....	64
	Результаты Главы 3	69
4	Программная реализация и экспериментальные исследования эффективности подхода к управлению ресурсами распределенных динамических вычислительных систем на базе нейронных сетей обнуления Чжана.....	70
4.1	Описание программной реализации подхода	70
4.2	Проведение экспериментов для проверки эффективности разработанного подхода управления	72
4.3	Сравнение результатов с существующими методами и анализ полученных данных	83
	Результаты Главы 4	86
	Заключение	88
	Список использованных источников	90
	Приложение А. Свидетельство о государственной регистрации программы для ЭВМ.....	105
	Приложение Б. Акты об использовании результатов.....	105

ВВЕДЕНИЕ

Актуальность темы исследования. В настоящее время эффективное использование ресурсов имеет прямое влияние на производительность систем, экономическую эффективность (минимизацию потребления различного рода ресурсов) и удовлетворение возможных ограничений, накладываемых на работу лицами, принимающими решения. Управление ресурсами в распределенных динамических системах является задачей с высокой вычислительной сложностью, требующей эффективных алгоритмов и методов для ее решения [1]. Существующие методы управления ресурсами не всегда способны оперативно и адекватно реагировать на динамические изменения и обеспечивать оптимальное использование ресурсов при различной, постоянно меняющейся, конфигурации сети. Так в 2015 году на долю информационно-коммуникационных технологий (ИКТ) приходилось до 5% мирового спроса на энергию. Фактически, только на центры обработки данных приходился 1% этого спроса, затмевая общее потребление энергии во многих странах. По прогнозам, к 2030 году на долю ИКТ будет приходиться 7% мирового спроса на энергию [2-3].

Вычислительные системы, особенно крупные центры обработки данных и серверные фермы, могут потреблять большие объемы электроэнергии, что может вести к значительным выбросам углекислого газа, особенно если электроэнергия производится с использованием источников, основанных на ископаемых топливах.

Для снижения влияния вычислительных систем на окружающую среду, многие организации переходят на более энергоэффективные технологии, такие как возобновляемые источники энергии (ветряные и солнечные установки), а также внедряют методы энергосбережения и оптимизации работы вычислительных ресурсов [4].

В современных условиях, когда вопросы экологии и устойчивого развития становятся все более актуальными, необходимость интеграции экологических аспектов в управление динамическими распределенными вычислительными системами становится неотъемлемой частью обеспечения эффективности и устойчивости процессов.

Исследование и учет экологических аспектов в динамических распределенных системах сталкивается с рядом вызовов [5]. Распределенные вычислительные системы включают в себя физические компьютерные ресурсы, в том числе каналы передачи данных, программное обеспечение, используемое как для обработки данных, так и для собственных нужд. Целью таких систем является объединение компьютерных ресурсов для решения трудоемких вычислительных задач [6]. При этом, для оценки эффективности их работы помимо классических критериев, таких как время доступа к ресурсам и время вычисления, утилизация и энергопотребления ресурсов, необходимо также учитывать негативное воздействие работы системы на окружающую среду. На сегодня одним из основных критериев оценки такого воздействия является эмиссия углекислого газа, как наиболее значимый по воздействию на климат неконденсируемым парниковым газом [7]. Это обуславливает необходимость создания модельно-алгоритмического обеспечения для управления динамическими распределенными вычислительными системами, основным критерием эффективности которых будет является снижение выбросов углекислого газа в ходе их функционирования.

В настоящее время предложено и в достаточной мере исследовано большое количество различных подходов управления распределенными системами, основанных как на точных, так и на эвристических алгоритмах. Эффективность их работы различна и зависит от уровня, на котором они используются, модели и структуры системы, стохастичности и устойчивости системы и прочих факторов [8]. Очевидным решением повышения эффективности управления распределенными динамическими вычислительными системами является применение гибридного подхода, позволяющего добиться максимальной полезности применения на соответствующих уровнях методов и алгоритмов, показывающих наилучший результат. Кроме того, применение гибридного подхода позволяет включать и исследовать эффективность новых для данной области инструментов, одним из которых являются нейронные сети обнуления (иначе нейронные сети Чжана). Нейронные сети обнуления показывают высокую эффективность при решении различных задач оптимального управления от

экономических (управление портфелем ценных бумаг) до сугубо технических (управление роботизированным манипулятором) [9]. Расширение класса решаемых задач оптимального управления инструментарием нейронных сетей обнуления является актуальной научной задачей для научной специальности системный анализ, управление и обработка информации.

Несмотря на заметный прогресс как в области оптимального управления различными системами на базе НСО, так и в классе задач управления распределенными динамическими вычислительными системами, существует множество нерешенных вопросов и проблем, которые требуют дальнейших исследований. Некоторые из них включают в себя:

1. Управление неопределенностью использования для работы системы энергии, при получении которой различна эмиссия углекислого газа.

2. Масштабируемость инструментов управления ресурсами в больших и сложных распределенных динамических вычислительных системах. Согласно закону Мура, сложность таких систем растет экспоненциально, что определяет необходимость разработки методов, способных эффективно ими управлять и обеспечивать высокую производительность при режимах работы в реальном времени.

3. В распределенных динамических системах необходимо решать задачи координации между различными участниками системы, следовательно нужны методы распределенного обучения и координации, которые позволят эффективно обмениваться информацией и совместно принимать решения.

4. Необходимо повышать точность и адаптивность моделей нейронных сетей обнуления, которые позволят более эффективно управлять ресурсами в сложных и динамических системах.

Таким образом, разработка гибридного подхода к управлению ресурсами распределенных динамических вычислительных систем на базе нейронных сетей обнуления (НСО) имеет высокую степень актуальности в различных научных и прикладных областях.

Степень разработанности темы. В исследования оптимального управления ресурсами распределенных динамических систем внесли существенный вклад множество ученых и исследовательских коллективов из разных стран. Некоторые из известных имен в этой области включают Рудольфа Калмана, Ричарда Беллмана, Харольда Курца, Ричарда Понсе, Стефана Рекхенбаха, Стивена Бойда. В России исследованиями в области управления динамическими распределенными вычислительными системами занимаются Михаил Гуревич, Леонид Федоров, Валерий Устюжанин, Олег Гусев, Олег Кравец, Александр Дулесов и многие другие. Коллективы, занимающиеся этой темой, включают Исследовательский центр «Нелинейные и адаптивные системы», Лабораторию управления и динамических систем Массачусетского технологического института, Центр оптимального управления и операций Стэнфордского университета, Институт управления сложными системами Российской академии наук и другие.

Исследователи занимающиеся разработкой и исследованием применимости метода НСО к различным задачам оптимального управления: профессора Китайского университета в Гонконге Юнонг Чжан и Цзюнь Ван, профессор Обернского университета штата Алабама Френк Улик, профессор Нишского университета Республики Сербия Предраг Станимирович, профессор Цзинаньского университета в Китае Джин Джин, профессор Клиффорд Стайнер из Колумбийского университета в Нью-Йорке, Адам Веллер из Университета Калифорнии в Лос-Анджелесе [10-15].

Объект исследования являются распределенные динамические вычислительные системы.

Предметом исследования является инструменты управление ресурсами в распределенных динамических системах.

Целью исследования является повышение эффективности управления распределенными динамическими вычислительными системами (РДВС) с учетом влияния их работы на окружающую среду.

Для достижения поставленной цели были поставлены и решены следующие задачи:

1. Анализ существующих методов управления ресурсами распределенных динамических вычислительных систем.

2. Разработка обобщенной математической и имитационной модели распределенной динамической вычислительной системы.

3. Факторный анализ обобщенной модели РДВС на основе данных имитационного моделирования для выявления информативных признаков и сокращения размерности задачи.

4. Разработка метода к динамическому управлению ресурсами в РДВС с учетом влияния их работы на окружающую среду.

5. Разработка алгоритма применения подхода к динамическому управлению ресурсами в РДВС.

6. Программная реализация полученных подхода и алгоритма.

Проведение численного эксперимента по оценке эффективности применения разработанного подхода и алгоритма к динамическому управлению ресурсами в РДВС с учетом влияния их работы на окружающую среду с использованием разработанного программного комплекса.

Научная новизна заключается в следующем:

1. Предложена обобщенная математическая модель распределенной динамической вычислительной системы, отличающаяся от известных учётом экологических последствий через расход, затрачиваемый на вычисления энергии.

2. Впервые предложена имитационная модель распределенной динамической вычислительной системы, отличающаяся от известных включением данных о технических параметрах вычислительных узлов и каналах передачи данных, задачах, информации о конфигурации РДВС, а также об экологических последствиях работы РДВС.

3. Впервые предложен гибридный метод адаптивного управления ресурсами в распределенных динамических вычислительных системах, направленный на минимизацию негативных экологических последствий, включающий в себя: алгоритм планирования задач миграции, репликации, задержки и вычислений;

технологии динамического управления частотой и напряжением работы процессоров; адаптивный алгоритм управления ресурсами РДВС.

4. Разработан обнуляющий нейросетевой алгоритм управления ресурсами распределенной динамической вычислительной системы, отличающийся от известных выбором формулы ошибки требуемого порядка, для более точного восстановления целевой функции.

Теоретическая значимость данной диссертационной работы заключается в развитии методов оптимального управлению ресурсами в динамических распределенных вычислительных системах. Работа расширяет аппарат нейросетевых методов оптимизации в нестационарных системах, расширяя и углубляя понимание проблем и методов управления динамическими гетерогенными системами обработки информации.

В частности, представленные в работе инструменты гибридного адаптивного управления ресурсами на основе обнуляющих нейронных сетей Чжана открывает новое направление в области разработки нейросетевых алгоритмов оптимизации для нестационарных динамических систем. Этот подход позволяет эффективно управлять использованием ресурсов в динамическом окружении, с учетом ограничений и заданных требований.

Практическая значимость связана с применением полученных результатов в реальных задачах управления ресурсами в таких системах, как центры обработки данных, облачные вычисления, распределенные вычисления, автоматизированные системы управления предприятием. В частности, использование предложенного подхода может помочь снизить негативное влияние на окружающую среду при этом сохранив эффективность работы распределенных динамических вычислительных систем. Кроме того, результаты исследования могут использоваться для разработки новых алгоритмов управления ресурсами, которые могут быть применены в различных сферах деятельности. Практическая значимость работы состоит в предоставлении практических решений и подходов к управлению ресурсами в распределенных динамических вычислительных

системах, которые могут быть применены для оптимизации работы и повышения производительности в различных областях и приложениях.

Основные результаты работы докладывались и обсуждались на конференциях всероссийского и международного уровня. По результатам работы подготовленно и опубликовано 17 научных работ, из которых 4 в рецензируемых научных журналах рекомендованных ВАК России, 7 в изданиях индексируемых в наукометрической базе Scopus, 6 – в изданиях РИНЦ. 4 программные разработки зарегистрированы как программы для ЭВМ.

Реализация результатов работы: диссертационная работа выполнена в рамках проекта Мегагрант «Гибридные методы моделирования и оптимизации в сложных системах», No 075-15-2022-1121 2022-н.в.

В работе были использованы **методы исследования:** системного анализа и статистики, математического моделирования, нейросетевые методы, процедурного программирования, численного эксперимента.

Основные положения, выносимые на защиту:

1. Предложена обобщенная математическая модель распределенной динамической вычислительной системы, отличающаяся от известных учётом экологических последствий через расход, затрачиваемый на вычисления энергии.

2. Впервые предложена имитационная модель распределенной динамической вычислительной системы, отличающаяся от известных включением данных о технических параметрах вычислительных узлов и каналах передачи данных, задачах, информации о конфигурации РДВС, а также об экологических последствиях работы РДВС.

3. Впервые предложен гибридный метод адаптивного управления ресурсами в распределенных динамических вычислительных системах, направленный на минимизацию негативных экологических последствий, включающий в себя: алгоритм планирования задач миграции, репликации, задержки и вычислений; технологию динамического управления частотой и напряжением работы процессоров; адаптивный алгоритм управления ресурсами РДВС.

4. Разработан обнуляющий нейросетевой алгоритм управления ресурсами распределенной динамической вычислительной системы, отличающийся от известных выбором формулы ошибки требуемого порядка, для более точного восстановления целевой функции.

Соответствие научной специальности. Основные положения соответствуют пункту 4 (разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации), 5 (разработка специального математического и программного обеспечения систем анализа, оптимизации, управления, принятия решений и обработки информации) и 11 пункту (методы и алгоритмы прогнозирования и оценки эффективности, качества, надежности функционирования сложных систем управления и их элементов) паспорта научной специальности «Системный анализ, управление и обработка информации, статистика».

Достоверность полученных научных результатов обеспечена корректным использованием теории информации, системного анализа и обработки данных, структурной надежности, а также соответствием теоретических значений и экспериментальных данных, полученных при апробации разработанных решений.

Апробация работы. Основные положения и результаты работы докладывались на Международной школе-семинаре «НММОС-2022: Гибридные методы моделирования и оптимизации в сложных системах» и на конференции «Ключевые тренды развития ИИ: наука и технологии», МГТУ им. Баумана, 2023, «III Всероссийская научная конференция с международным участием «Наука, технологии, общество: Экологический инжиниринг в интересах устойчивого развития территорий» (НТО-III)» 2022. Конференция студентов, аспирантов и молодых ученых «Перспектив-2022». Красноярск: СФУ, 2022. III Международная конференция MIST: Aerospace-III 2019: Передовые технологии в аэрокосмической отрасли, машиностроении и автоматизации. I Международный семинар MIST: Aerospace-I 2018: Передовые технологии в аэрокосмической отрасли, машиностроении и автоматизации.

Диссертация была представлена на расширенных семинарах кафедры системного анализа и исследования операций Сибирского государственного университета науки и технологий имени академика М.Ф. Решетнева.

Публикации. По теме диссертационной работы опубликовано 17 научных работ, из которых 4 в рецензируемых научных журналах рекомендованных ВАК России, 7 в изданиях индексируемых в наукометрической базе Scopus, 6 – в изданиях РИНЦ. 4 программные разработки зарегистрированы как программы для ЭВМ.

Структура и объем диссертации. Диссертация состоит из введения, четырёх глав, заключения, списка литературы и приложений.

1 Основные особенности задачи управления ресурсами в распределенных динамических вычислительных системах

Современный информационный мир характеризуется активным развитием распределенных динамических вычислительных систем (РДВС), которые играют важную роль в обработке и анализе данных, решении сложных задач и обеспечении эффективной работы множества устройств и компонентов. Управление ресурсами в таких системах становится критическим аспектом для обеспечения их производительности и минимизации углеродного следа [16].

Рассмотрим базовые понятия и терминологию, используемую в данной области, чтобы обеспечить ясное понимание ключевых концепций. Изучим понятие РДВС, в котором рассмотрим структуру и сложности, с которыми сталкиваются такие системы. Следующим шагом проведем обзор методов управления ресурсами и их применение в распределенных динамических вычислительных системах. Эта глава позволит лучше понимать контекст и значимость тем, которые будут рассмотрены далее.

1.1 Особенности распределенных вычислительных динамических систем

Введем основные термины и определения:

Оптимальное управление – это задача проектирования системы, обеспечивающей для заданного объекта управления или процесса закон управления или управляющую последовательность воздействий, обеспечивающих максимум или минимум заданной совокупности критериев качества системы [17].

Динамическая система – любой объект или процесс, для которого однозначно определено понятие состояния, как совокупности некоторых величин в некоторый момент времени, и задан закон, описывающий эволюцию начального состояния с течением времени [18].

Распределенная система – это набор компьютерных программ, использующих вычислительные ресурсы нескольких отдельных вычислительных узлов для достижения одной общей цели. Распределенные системы направлены на устранение узких мест или единых точек отказа в системе [19].

Вычислительная система – это комплекс аппаратных и программных компонентов, предназначенных для выполнения вычислительных задач. Она включает в себя центральный процессор (CPU), память, устройства ввода и вывода (клавиатура, монитор, мышь и другие), операционную систему и прикладное программное обеспечение [20].

Ресурсы системы – это любые элементы, которые используются для достижения целей системы. Они могут быть физическими или нематериальными и могут включать в себя такие элементы, как люди, деньги, время, знания, технологии, материалы, оборудование и т.д. Ресурсы могут быть использованы в различных процессах системы и могут влиять на ее производительность, эффективность и стоимость. Их оптимальное использование позволяет системе достигать своих целей с наименьшими затратами [21-22].

Искусственные нейронные сети – это математическая модель, а также ее программные или аппаратные реализации, построенная в некотором смысле по образу и подобию сетей нервных клеток живого организма [23].

Нейронные сети обнуления (НСО) – это специальный тип нейронных сетей, который позволяет достигать оптимизации путем обнуления (отбрасывания) ненужных параметров в процессе обучения. Это позволяет создавать компактные модели, которые быстрее обучаются и могут быть более эффективно использованы в распределенных динамических вычислительных системах [24].

Моделирование – это процесс создания упрощенной абстрактной модели системы с целью лучшего ее понимания и анализа. Моделирование может быть использовано для определения свойств системы, их изменения во времени, их взаимодействия с другими системами, а также для прогнозирования результатов различных сценариев поведения системы [25].

Математическая модель – это формализованное представление системы, описывающее ее функционирование в виде математических уравнений и зависимостей. Эта модель может использоваться для анализа и оптимизации системы, а также для прогнозирования ее поведения в различных условиях [26].

Формальная модель – это тип математической модели, которая использует формальные математические методы и языки для описания и анализа системы. Она представляет собой абстрактное, формальное описание структуры и функционирования системы, основанное на формальных определениях и символах [27].

Распределенные динамические вычислительные системы представляют собой сложные компьютерные сети, состоящие из взаимосвязанных устройств, которые выполняют распределенные вычисления и обработку данных. Они широко применяются в различных областях, включая облачные вычисления, мобильные сети, системы управления транспортом, медицинские информационные системы и многие другие.

Для лучшего понимания выделим основные характеристики РДВС [28-33]:

1. **Распределенность.** РДВС состоит из множества вычислительных узлов, которые могут быть физически разделены и распределены по разным местоположениям. Это позволяет системе обрабатывать задачи параллельно и увеличивает ее масштабируемость.

2. **Динамичность.** РДВС часто работает в условиях изменяющейся нагрузки и ресурсов. Узлы могут входить в систему или выходить из нее, ресурсы могут меняться во времени. Это требует адаптивности и способности системы менять свою конфигурацию в реальном времени.

3. **Масштабируемость.** РДВС спроектированы так, чтобы обеспечивать масштабируемость, то есть способность эффективно управлять вычислительными ресурсами при увеличении числа участников и задач.

4. **Гибкость.** они предоставляют гибкость в развертывании и настройке, что делает их адаптированными под различные задачи и сценарии использования.

5. **Адаптивность.** РДВС способны адаптироваться к изменяющейся нагрузке и условиям, оптимизируя распределение ресурсов.

6. **Высокая производительность.** Одной из ключевых целей РДВС является обеспечение высокой производительности и параллельной обработки

задач. Система должна оптимально использовать ресурсы для ускорения выполнения вычислений.

7. Управление ресурсами. Важной характеристикой РДВС является способность эффективно управлять ресурсами, такими как процессорное время, память, сетевая пропускная способность и другие. Это включает в себя задачи планирования ресурсов и балансировки нагрузки.

8. Отказоустойчивость. РДВС должна быть устойчивой к отказам вычислительных узлов или другим сбоям. Это достигается путем резервирования и репликации данных, а также механизмами автоматического восстановления.

9. Безопасность. С учетом распределенности и динамичности, безопасность данных и сетевой коммуникации является важным аспектом для РДВС. Это включает в себя аутентификацию, шифрование и управление доступом.

10. Согласованность данных. В системах с распределенными данными необходимо обеспечивать согласованность данных между разными узлами. Это достигается через применение алгоритмов согласования данных.

11. Сообщения и коммуникация. РДВС использует механизмы сообщений и коммуникации между узлами для передачи данных и согласования действий.

Эти характеристики являются основой для понимания и проектирования РДВС и играют ключевую роль в ее эффективном функционировании.

Современные информационные технологии тесно связаны с использованием РВДС, так как они играют важную роль в обеспечении высокой доступности, масштабируемости и производительности различных приложений и сервисов. Понимание характеристик и сложностей РДВС, а также осознание их роли в современных информационных технологиях и экологических аспектов управления ресурсами, является ключевым шагом для разработки эффективных стратегий управления ресурсами в РВДС с учетом негативных воздействий на окружающую среду (выбросов углекислого газа) [34].

1.2 Обзор методов управления ресурсами и их применение в распределенных динамических вычислительных системах

Методы управления ресурсами играют важную роль в обеспечении эффективности и устойчивости распределенных динамических вычислительных систем (РДВС). В данном разделе представлен обзор существующих методов и стратегий, а также их роли и применения в контексте РДВС.

Существует множество классических подходов к оптимизации управления ресурсами, которые применимы к распределенным динамическим вычислительным системам, каждый из которых имеет свои преимущества и недостатки. Выбор метода зависит от конкретной задачи и доступных ресурсов.

Для управления ресурсами в вычислительной распределенной динамической системе могут использоваться следующие методы системного анализа:

Моделирование и анализ производительности:

Математические модели, такие как сетевые модели, модели массового обслуживания и модели сетей Петри, используются для описания производительности системы. Они могут быть представлены в виде систем уравнений или стохастических процессов [35].

Вероятностные методы, такие как методы Монте-Карло и анализ Марковских цепей, могут использоваться для симуляции и анализа производительности системы [36].

Методы теории очередей применяются для моделирования систем с очередями и определения характеристик, таких как среднее время ожидания и пропускная способность [37].

Оптимизация управления ресурсами также может осуществляться с помощью нижеперечисленных методов:

Прогнозирование и управление нагрузкой может осуществляться такими методами, как:

Методы временных рядов, такие как экспоненциальное сглаживание и авторегрессионные модели, используются для прогнозирования будущей нагрузки

на систему на основе исторических данных. Они позволяют предсказать будущие потребности в ресурсах и принять соответствующие меры.

Методы кластеризации и классификации могут использоваться для анализа и классификации нагрузки на основе различных характеристик. Это помогает в определении стратегий управления нагрузкой, таких как масштабирование ресурсов или балансировка нагрузки между узлами системы [38-41].

Контроль и управление производительностью может осуществляться методами:

Методы контроля производительности, такие как PID-регуляторы, используют уравнения и алгоритмы обратной связи для регулирования системных параметров и поддержания заданных показателей производительности [42].

Методы управления, такие как метод динамического программирования или методы оптимального переключения, основаны на математическом анализе и оптимизации функций стоимости или целевых функций системы [43].

Принятие решений на основе данных:

Методы машинного обучения включают различные алгоритмы, такие как линейная регрессия, деревья решений, метод опорных векторов и нейронные сети. Они используются для анализа данных, обучения на основе исторических данных и принятия решений на основе обнаруженных паттернов и зависимостей [44].

Алгоритмы решения задач оптимизации, такие как генетические алгоритмы, алгоритмы роя частиц и алгоритмы отжига, применяются для нахождения оптимальных решений в условиях неопределенности и сложных ограничений.

Также используются адаптивные алгоритмы для управления РДВС [45-46]:

Адаптивные алгоритмы управления могут включать в себя алгоритмы с обратной связью, алгоритмы на основе модели или алгоритмы на основе анализа. Они могут использовать математические методы, такие как динамическое программирование или методы оптимизации, для определения оптимальных стратегий управления ресурсами и регулирования их выделения в реальном времени.

Для того чтобы решить задачу определения оптимальной программы управления, сначала строится математическая модель управляемого объекта или процесса. Эта модель описывает поведение объекта со временем при воздействии управляющих воздействий и учитывает текущее состояние объекта.

Важно отметить, что часто задачи управления ресурсами могут быть рассмотрены как задачи математического программирования, и для их решения используются численные методы. Эти методы позволяют найти оптимальные стратегии управления, которые удовлетворяют заданным критериям качества и ограничениям на ресурсы.

При оптимизации управления ресурсами в многоуровневых системах, таких как крупные химические производства, металлургические комплексы и энергетические объекты, применяются многоцелевые и многоуровневые иерархические системы управления. Эти системы включают в себя разработку критериев качества управления для каждого уровня управления и координацию действий между уровнями [47].

В случае, когда задача оптимального управления связана с конфликтом или неопределенностью, применяется теория дифференциальных игр [48].

Для решения задач управления ресурсами в условиях ограниченной исходной информации и наличия ошибок в измерениях используется метод максимального правдоподобия. Системы управления ресурсами, способные учиться и улучшать свою производительность на основе накопленного опыта, называются обучающимися системами управления ресурсами [49].

Реальное поведение объектов или систем всегда отличается от идеального в силу различных факторов, таких как неточность начальных данных, неполная информация о внешних воздействиях и ошибка при реализации управления. Для минимизации отклонений используются системы автоматического регулирования [50].

Для решения задач управления ресурсами с очень большой размерностью, которые не могут быть решены классическими методами математики, используют методы ситуационного управления [51].

Управление ресурсами в РВДС представляет собой сложную задачу из-за изменчивой нагрузки и динамических условий. Эффективное распределение ресурсов становится особенно важным с учетом современных экологических требований. Снижение углеродного следа РВДС является актуальной задачей, которая может быть достигнута через повышение эффективности управления ресурсами РДВС и применения энергосберегающих технологий [52].

Одними из основных особенностей оптимизации управления ресурсами распределенных динамических системам являются их сложность и нелинейность. Такие системы могут содержать большое число взаимосвязанных элементов, которые влияют друг на друга и находятся в состоянии динамического изменения. В таких условиях традиционные методы управления и оптимизации могут быть недостаточно эффективными или даже неприменимыми.

Существует множество комплексных и адаптивных подходов к управлению ресурсами распределенных динамических систем. Некоторые из них ориентированы на оптимизацию производительности, другие на минимизацию затрат, а третьи на минимизацию времени или энергии. Также большую популярность этих методов обусловило появление удаленных вычислительных центров, в которых управление ресурсами чаще осуществляется путем планирования задач, в таблице 1 перечислены основные методы планирования [53].

Таблица 1. Краткое изложение методов [54-61]

Метод	Техника	Преимущества	Ограничения	Параметры
В порядке живой очереди	Последние поступившие задания должны дожидаться окончания выполнения предыдущих.	Простой в реализации и эффективный.	Увеличивает время ожидания задач и размер неучтенных задач. Дисбаланс нагрузки и снижение локальности данных.	-

Продолжение таблицы 1. Краткое изложение методов [54-61]

Метод	Техника	Преимущества	Ограничения	Параметры
Сначала самое короткое задание	Выбирает самую короткую задачу, которая будет выполнена первой.	Сокращает время выполнения по сравнению с FCFS и RR.	Неравномерное распределение нагрузки на серверы.	Время выполнения. Время ответа.
Круговой турнир	Циркулярно распределяет задачи.	На каждую задачу отводится равное количество процессорного времени.	Более высокое среднее время ожидания.	-
Шьям и Манви	Миграция виртуальных машин.	Максимизирует использование ресурсов при минимизации времени и бюджета.	Нужно было больше агентов для поиска лучшего ресурса.	Время выполнения. Время макияжа. Время ответа. Использование ресурсов.
Wang et al.	Динамическая подготовка ресурсов.	Учитывает такие ресурсы, как ЦП, память и хранилище.	Локальность данных и репликация не рассматриваются.	Стоимость выполнения. Наличие.
Zhao et al.	Энергоэффективный метод, в котором наборы данных и задачи рассматриваются как двоичное дерево с использованием алгоритма корреляционной	Минимизирует энергопотребление облачных центров обработки данных.	Онлайн-планирование не учитывается.	Стоимость выполнения. Использование ресурсов. Энергия потребление.

Продолжение таблицы 1. Краткое изложение методов [54-61]

Метод	Техника	Преимущества	Ограничения	Параметры
	кластеризации данных.			
Dubey et al.	Алгоритм планирования на основе IBA (Improved Backfill Algorithm).	Минимизирует время выполнения и увеличивает использование ресурсов.	Чем больше задач, тем меньше производительность. Учитывает приоритет задачи.	Время выполнения. Makespan. Использование ресурсов.
Elseoud et al.	Эвристический алгоритм Online Potential Finish Time.	Сокращает время и стоимость выполнения в облачных вычислениях, выполняя задачи с наименьшей задержкой.	Локальность данных и репликация не рассматриваются.	Сроки/стоимость выполнения. Makespan. Время ответа. Использование ресурсов.
Алгоритм задержки	Назначайте задачи на основе расположения входных данных. Он откладывает задачи до тех пор, пока не будут доступны необходимые данные.	Простота планирования. Обеспечивает локальность данных.	Дисбалансирующая нагрузка, которая может привести к более высоким задержкам в выполнении задач и снижению пропускной способности.	Время выполнения.
Li et al.	Онлайн-планирование заданий на основе переноса данных	Обрабатывает перенос данных.	Планирует задачи последовательно, что увеличивает время ожидания.	Производительность.

Окончание таблицы 1. Краткое изложение методов [54-61]

Метод	Техника	Преимущества	Ограничения	Параметры
	на основе компромисса между стоимостью передачи данных и стоимостью времени ожидания.		Характеристики систем не рассматриваются. Репликация данных не обсуждается.	
Редди и др.	Модифицированная оптимизация Ant Colony.	Учитывает оперативную память виртуальных машин, пропускную способность, хранилище, Скорость обработки и Makespan в фитнес-функции.	Онлайн-задачи с интенсивным использованием данных не решаются.	Использование ресурсов. Makespan. Балансировка нагрузки.
Biswas et al.	Динамичный круговой турнир.	Динамически определяет квант времени.	С голодом не справлялись.	Перелом. Время ожидания.
Soltani et al.	Генетическая метаэвристика.	Многоцелевой взвешенный генетический алгоритм для повышения производительности.	Онлайн-задачи с интенсивным использованием данных не решаются.	Время ответа. Время ожидания. Makespan.
Mohseni et al.	Алгоритм планирования жесткого диска и процессора (HCS).	Планирование нескольких задач между многоядерными системами.	Память, пропускная способность и задержка многоядерных систем не учитываются.	Время выполнения. Потребление энергии.

Современные методы обычно рассматривают различные адаптивные подходы к управлению ресурсами, основанные на таких параметрах, как балансировка нагрузки, оптимизация энергопотребления и использование машинного обучения. Эти исследования акцентируют внимание на оптимизации производительности и энергоэффективности в распределенных вычислительных системах. Однако при этом они часто не учитывают динамические изменения и специфику работы в реальном времени, что ограничивает их применимость в контексте быстро развивающихся технологий [62].

В работе [61] представлен более глубокий анализ конкретных методов оптимизации на основании таких факторов, как количество публикаций, распространение метода в зависимости от применимости, точности и объяснимости. В результате анализа было обнаружено, что смешанное целочисленное линейное программирование является наиболее часто используемым методом оптимизации для решения проблемы управления в распределенных гетерогенных сетях. В дополнение к этому необходимо отметить, что роевые и генетические алгоритмы являются широко используемыми метаэвристическими алгоритмами, а использование мультиагентных и теоретико-игровых алгоритмов начинает увеличиваться с появлением более децентрализованных систем. Мультиагентные системы оказываются более выгодными в децентрализованной среде, и очевидна необходимость комплексного решения децентрализованного инструмента моделирования поведения распределенных, постоянно изменяющихся систем. Кроме того, проблемы обязательств отдельных узлов и экономической диспетчеризации были изучены в большей степени, учитывая влияние результатов на эффективность и стоимость системы. Увеличение доли проблем в управлении распределенными вычислительными сетями, решаемых с помощью гибридных методов, показало тот факт, что применение гибридных методов для решения является перспективным, и направления исследований должно быть к достижению более высокой эффективности за счет использования преимуществ отдельных методов для коллективного достижения лучших результатов.

Выводы к Главе 1

В первой главе настоящей диссертации были рассмотрены теоретические основы, которые служат фундаментом для дальнейшего исследования в области оптимального управления ресурсами в распределенных динамических вычислительных системах (РВДС). В ходе анализа темы были получены следующие основные выводы:

1. Введены и обозначены ключевые термины и определения, необходимые для более глубокого понимания предметной области и проведения исследования в данной области.

2. Распределенные динамические вычислительные системы (РДВС) представляют собой активно развивающуюся область, широко применяемую в современном мире. Они обладают высокой эффективностью и способностью обрабатывать огромные объемы данных в режиме реального времени. Однако управление ресурсами в РВДС представляет собой сложную задачу, особенно при изменяющейся нагрузке и ограниченных ресурсах. Это делает актуальным исследование в области управления ресурсами динамических распределенных вычислительных системах.

3. Обзор методов управления ресурсами позволил выделить проблемы и недостатки существующих подходов и выявить значимые признаки управления в РДВС. Это предоставило базу для разработки новых методов и подходов к оптимизации управления ресурсами.

Эта глава является отправной точкой для дальнейшего исследования и разработки новых методов повышения эффективности управления ресурсами в РДВС. Результаты данной главы оказывают влияние на все последующие разделы диссертации и предоставляют теоретическую основу для практических исследований в данной области.

2 Моделирование и анализ распределенной динамической вычислительной системы

Во второй главе исследования мы переходим от теоретических основ к более практическому анализу и моделированию распределенных динамических вычислительных систем (РДВС). Эта глава посвящена изучению структуры РДВС, ее характеристик и особенностей, а также разработке математической и имитационной моделей, которые позволят описать динамическое поведение таких систем.

2.1 Описание структуры распределенной динамической вычислительной системы и её характеристики

Распределенные динамические вычислительные системы (РДВС) представляют собой сложные и многокомпонентные системы, которые выполняют вычисления, обрабатывают данные и управляют ресурсами в динамически изменяющейся среде [63]. В данном разделе представлено более подробное описание структуры РДВС и рассмотрены их основные характеристики, которые оказывают влияние на управление ресурсами.

Архитектура рассматриваемой нами РДВС представляет собой совокупность компонентов и их взаимосвязей, которые обеспечивают функционирование системы. Основные компоненты включают в себя:

1. Узлы вычислений. Это вычислительные устройства, которые выполняют задачи и обрабатывают данные. Узлы могут быть физическими серверами, виртуальными машинами или даже устройствами Интернета вещей (IoT).
2. Сетевая инфраструктура. Сеть связывает узлы вычислений и обеспечивает коммуникацию между ними. Это может быть локальная сеть (LAN) или глобальная сеть (интернет).
3. Системы управления. Системы управления контролируют и управляют всеми компонентами РДВС, включая вычисления, сети и ресурсы [64].

Понимание основных компонентов РДВС является важным шагом в разработке оптимальных методов управления ресурсами. Эти компоненты

оказывают влияние на выбор стратегий управления и определение оптимальных решений для обеспечения эффективной работы системы. Но также необходимо учитывать ресурсы, в которых функционирует система. Современные предприятия располагают следующими ресурсами [65]:

Вычислительные ресурсы. В состав динамических распределенных вычислительных систем входят серверы, компьютеры, кластеры и облачные инфраструктуры. Эти ресурсы используются для выполнения вычислительных задач, обработки данных и предоставления услуг. Оптимальное распределение и использование этих ресурсов с учетом экологических параметров может значительно влиять на эффективность и стойкость системы.

Энергетические ресурсы. Для обеспечения работы вычислительных и технических устройств требуется энергия. Потребление электроэнергии оказывает влияние на эффективность и экологическую стойкость системы. Рассмотрение и оптимизация энергопотребления становятся ключевыми аспектами для снижения негативного воздействия на окружающую среду.

Технические устройства. Оборудование, такое как серверы, маршрутизаторы, коммутаторы и датчики, играют важную роль в функционировании динамических распределенных вычислительных систем. Оптимальное управление и использование технических устройств может существенно повысить производительность системы и уменьшить потребление ресурсов.

Временные ресурсы. Эффективное управление временными ресурсами, такими как время выполнения задач, временные интервалы, дедлайны и периоды активности, является важным аспектом для обеспечения непрерывности работы системы и снижения негативного воздействия на окружающую среду.

Информационные ресурсы. Информация и данные являются ключевыми активами в динамических распределенных вычислительных системах. Эффективное управление информационными потоками, их передачей и обработкой влияет на производительность и эффективность системы [66].

На данном этапе описывается общее определение и анализ ресурсов, осуществляющих работу системы для дальнейшего ограничения круга ресурсов, интересующих нас в контексте данного исследования. Эти факторы составляют фундаментальную основу для оценки описания функционирования системы.

Для общего представления задачи опишем концептуальную модель, описывающую вход, ресурсы системы, управляющие параметры и выход, представлена на рисунке 1.



Рисунок 1 – Концептуальная модель задачи

Следующим шагом будет построена математическая модель для анализа динамического поведения РДВС и методов управления ресурсами в этой среде.

2.2 Разработка математической модели для динамического поведения распределенной вычислительной системы

Для разработки стратегии управления ресурсами РДВС создадим ее математическую модель. Для начала опишем характеристики узлов и их обозначения:

$U = \{u_i\}$ – множество узлов РДВС, где u_i обозначает i -й узел.

sc_i – целочисленное значение, представляющее емкость хранилища узла u_i (МБ).

r_i – вещественное значение, представляющее скорость чтения (получения данных) узла u_i (МБ/с).

w_i – вещественное значение, представляющее скорость записи (передачи данных) от узла u_i (МБ/с).

$RAM [u_i]$ – целочисленное значение, представляющее доступный объем оперативной памяти узла u_i (Байт).

$N_CPU [u_i]$ – количество процессорных ядер узла u_i .

$P_CPU [u_i]$ – целочисленное значение, представляющее производительность каждого ядра процессора на узле u_i (миллион инструкций в секунду — MIPS).

b_{ij} – вещественное значение, представляющее пропускную способность соединения между узлами u_i и u_j (МБ/с).

β_{ij} – время передачи данных между узлами u_i и u_j , определяемое формулой:

$$\beta_{ij} = \begin{cases} 0 & \text{если } i = j \\ \frac{1}{b_{ij}} & \text{в противном случае} \end{cases} \quad (1)$$

$PP[u_i]$ – это целочисленное значение, представляющее вычислительную мощность узла u_i (в миллионах инструкций в секунду — MIPS), которое вычисляется следующим образом:

$$PP[u_i] = N_CPU[u_i] \times P_CPU[u_i], \quad (2)$$

где $N_CPU [u_i]$ – количество ядер u_i , а $P_CPU [u_i]$ – производительность каждого ядра в u_i .

TP_i определяет список выполняемых задач в u_i .

Опишем модели целевой функции, используемой в задаче управления ресурсами в распределенных вычислительных системах. Мы предположили, что распределенная вычислительная система включают в себя множество вычислительных узлов (CU – computing nodes), представленных $CU = \{cu_1, cu_2, cu_3, \dots\}$. Процессоры вычислительных узлов в момент времени t_n , могут работать на различной частоте, что, в свою очередь, потребует различное

потребление энергии. Множество $DCSCP(t_n) = \{CU_i(P_j(F_j, V_j, E_j))\}$, где CU_i – i -й вычислительный узел, P_j – j -й процессор этого узла работающий на частоте F_j , напряжении V_j и потребляющий для работы энергию E_j . Для оценки экологичности процесса вычисления введем переменную E_{eco} , показывающую количество выброса углекислого газа для получения этой энергии.

Для расчета экологических последствий (environmental impacts) работы распределенной вычислительной системы в любой момент времени будем использовать формулу

$$EI = \sum_{i,j} CU_i(P_j(F, V, E(t))) \times E_{eco}(t)_i. \quad (3)$$

Важно отметить, что величина E_{eco} на каждом из вычислительных узлов i будет иметь различное значение, т.к. система является распределенной в пространстве и экологичность потребляемой таким узлом энергии будет отличаться в зависимости от механизма ее получения.

Для вычисления среднего времени выполнения (AETIME) задачи T_k на j -ом процессоре P_j i -ого вычислительного узла CN_i будем использовать формулу

$$AETIME(T_k, CU_i(P_j)) = \frac{1}{J} \sum_{c=1}^N \frac{Task_len(T_k)}{CU_i(P_j)_c} \quad (4),$$

где $Task_len(T_k)$ является сложностью k -ой задачи измеряемую в миллионах инструкций процессора в секунду, необходимых для вычисления этой задачи. $CU_i(P_j)_c$ подразумевает характеристики работы j -ого вычислительного узла CU включающего j процессоров P , из J множества процессоров.

Самое раннее время начала каждой задачи в этой модели можно рассчитать следующим образом:

$$EST(T_k, CU_i(P_j)) = \begin{cases} 0, & \text{если } T_k \text{ является входной задачей для узла } \max(avail(CU_i(P_j)) \\ & (FT(T_k) + CT(T_i, T_k)) \text{ в противном случае} \end{cases} \quad (5),$$

, для каждого T_i предшествующего (T_k)

где $avail(CU_i(P_j))$ является моментом, когда i -й узел готов к выполнению указанной задачи. Время выполнения каждой задачи FT можно определить с помощью следующего уравнения:

$$FT(T_k, CU_i(P_j)) = \{deadline(w_k), \text{ если } T_k \text{ является выходной задачей } EStime(CU_i(P_j)) + Ave\ Exection\ Time(CU_i(P_j))\}, \text{ в противном случае} \quad (6)$$

Ave Exection Time – среднее время выполнения

Здесь $deadline(w_k)$ – крайний срок выполнения k -го задачи, СТ – время передачи данных между T_k и T_i , которое может быть рассчитано с использованием следующего уравнения:

$$CT(T_k, T_i) = \begin{cases} 0, \text{ если } CU(P(T_k)) = CU(P(T_i)) \\ \frac{Date(T_k, T_i)}{Bandwidth(CU(P(T_k)), CU(P(T_i)))} \text{ в противном случае} \end{cases} \quad (7)$$

$Date(T_k, T_i)$ указывают объем данных, которыми необходимо обмениваться между задачами, а $Bandwidth(CU(P(T_k)), CU(P(T_i)))$ обозначает пропускную способность между узлами, выполняющими задачи T_k и T_i . Кроме того, продолжительность задачи может быть рассчитана на основе:

$$makespan(w_i) = \{(FT(T_k)) | T_k \in w_i\} \quad (8).$$

Таким образом

$$f(x, y) \rightarrow \min$$

при ограничениях:

$$\sum_{i=1}^I C(x)_k^t \leq \sum_{j=1}^J N(y)_j^t,$$

$$T(C(x)) \leq const_k, k = 1, \dots, g,$$

где $f(x, y)$ – функция, отражающая совокупное энергопотребление решения вычислительных задач обработки данных на отрезке времени в один временной такт. Здесь x – критерий, отражающий технические параметры работы вычислительных узлов, к примеру тактовую частоту процессоров, y – размещение задач на узлах.

$C(x)_k^t$ – ресурсы, необходимые для решения k -ой задачи в момент времени t ,

$N(y)_j^t$ – свободная мощность j -го ресурса,

$T(C_k)$ – строго регламентированное предельное время решения k -й задачи, обусловленное сохранением производительности системы.

Функция, отражающая совокупное время решения задач обработки данных на любом отрезке времени, представляется в виде задачи составления расписания.

Имеется множество вычислительных задач A_1, \dots, A_m с потребностями ресурсов на их решение a_1, \dots, a_m и множество вычислительных узлов B_1, \dots, B_n с наборами ресурсов для вычислений b_1, \dots, b_n сумма задач меньше либо равна сумме ресурсов:

$$\sum_{i=1}^m a_i \leq \sum_{j=1}^n b_j \quad (9)$$

Время решения (вычисления) t_{ij} каждой задачи A_i на любом B_j заданы в виде

матрицы $C_{ij} = \begin{vmatrix} t_{11} & \dots & t_{1j} \\ \dots & \dots & \dots \\ t_{i1} & \dots & t_{ij} \end{vmatrix}.$

Требуется выбрать такое решение C_{ij} , чтобы все множество поступивших в систему задач A было распределено по узлам, удовлетворялись условия 1 и $f(x, y)$ стремилось к минимуму.

Предложена модель процессов управления ресурсами в распределенных динамических вычислительных системах, которая описывает поведение ресурсов в системе и позволяет оценить их эффективное использование.

2.3 Имитационное моделирование распределенной динамической вычислительной сети и анализ факторов, влияющих на энергопотребление и производительность

Распределенный и многозадачный характер распределенных динамических вычислительных систем делает сложным проведение экспериментов на реальных системах. В силу их географического разнообразия и непрерывных изменений в нагрузке и структуре участвующих узлов, традиционные методы тестирования и оптимизации на реальных аппаратных платформах становятся непрактичными. Для исследования и разработки эффективных стратегий управления и оптимизации ресурсов в таких системах необходимо обратиться к методам моделирования. Сложный стохастический характер многих процессов в распределенных

динамических системах, включая появление новых задач, возможность отказа узлов и изменчивость сетевых каналов, делает аналитическое моделирование практически невозможным. Имитационное моделирование, позволяя учитывать эти случайные и динамические процессы, становится более подходящим методом для исследования и оптимизации распределенных динамических вычислительных систем.

Имитационное моделирование в распределенных динамических вычислительных системах позволяет соблюсти основные принципы моделирования систем [67]. При имитационном моделировании мы воспроизводим функционирование системы во времени, учитывая ее распределенную и динамическую структуру. Имитационные модели в данном контексте позволяют учесть различные элементарные явления, такие как появление новых задач, отказы узлов и изменения в нагрузке на сеть. Эти модели сохраняют логическую структуру системы и последовательность событий, отражая динамические процессы в системе с учетом ее сложного распределенного характера [68].

Имитационное моделирование в распределенных динамических вычислительных системах обладает рядом ключевых преимуществ [69]:

1. Имитационные модели позволяют исследовать различные стратегии распределения вычислительных ресурсов. Анализируя поведение системы в различных условиях и сценариях, можно оптимизировать распределение ресурсов для сохранения производительности при минимизации энергопотребления.

2. Управление ресурсами в распределенных системах должно учитывать случайные события, такие как отказы узлов или внезапные изменения нагрузки. Имитационное моделирование позволяет создавать сценарии с различными стохастическими элементами для тестирования стратегий управления и их оценки.

3. Распределенные системы часто подвергаются изменениям в нагрузке, структуре и требованиях. Имитационное моделирование позволяет управлять и адаптироваться к динамическим изменениям, прогнозировать их влияние на ресурсы и разрабатывать стратегии управления, способные эффективно реагировать на изменяющиеся условия.

С использованием имитационной модели мы проводим анализ производительности и энергопотребления РДВС. Это включает в себя изучение характеристик системы в различных сценариях работы, выявление узких мест и оценку эффективности использования ресурсов [70]. Имитационное моделирование также позволяет апробировать возможные методы управления ресурсами и проверять их работоспособность. Путем имитации различных сценариев и анализа их воздействия на систему, можно разрабатывать стратегии управления ресурсами, которые не только сохраняют производительность, но и обеспечивают минимизацию выбросов углекислого газа в условиях переменчивой нагрузки и неопределенности в данных.

В качестве среды моделирования для управления ресурсами в распределенных динамических вычислительных системах была выбрана среда имитационного моделирования Matlab Simulink [71]. Интегрированный в Matlab Simulink предоставляет графический интерфейс для создания детальных моделей системы, позволяя анализировать и оптимизировать стратегии управления ресурсами в реальном времени. С широкой библиотекой предопределенных блоков для моделирования вычислительных узлов и сетей связи, Simulink обеспечивает возможность анализа динамических характеристик системы и реагирования на изменяющиеся условия, обеспечивая точные и реалистичные результаты в управляемой среде.

Разработанная имитационная модель для управления ресурсами в распределенных динамических вычислительных системах использует дискретно-событийный подход к моделированию. Время и стохастические события, такие как изменения в загрузке или появление новых задач, обрабатываются как дискретные события. Эта модель обеспечивает возможность адаптироваться к динамическим изменениям в системе, позволяя эффективно управлять ресурсами в условиях переменчивой нагрузки и неопределенности, и может быть успешно использована для моделирования систем с тысячами вычислительных узлов на стандартном персональном компьютере, обладающем ограниченными ресурсами.

Структура модели включает следующие компоненты:

Узлы. Каждый узел моделируется как сущность с определенными характеристиками, которые зависят от параметров ресурсов. Эти характеристики включают в себя емкость хранилища, объем оперативной памяти, скорость чтения и записи данных, а также производительность процессора. Узлы представляют вычислительные устройства в системе.

Сеть. Мы моделируем сеть как ресурс, обладающий определенными характеристиками, такими как пропускная способность и вероятность передачи данных между узлами. Эти характеристики сети влияют на эффективность обмена данными и коммуникацию между узлами в системе.

Задачи. Задачи моделируются как процессы, которые выполняются на узлах системы. Задачи потребляют процессорное время, выполняют операции чтения и записи данных, и имеют возможность мигрировать между узлами в зависимости от текущей нагрузки и доступных ресурсов. Моделирование задач позволяет учитывать динамические изменения в системе.

Ограничения и критерии оптимизации. Мы устанавливаем ограничения на использование ресурсов, такие как ограничение процессорного времени для каждого узла. Кроме того, мы определяем критерии оптимизации, включая сохранение производительности системы и минимизацию энергопотребления (энергоэффективность). Система должна учитывать потребление ресурсов каждой задачей и выполнять их в установленные сроки, обеспечивая эффективное использование ресурсов системы.

Эти компоненты формируют основу нашей модели распределенной динамической вычислительной системы, позволяя анализировать и оптимизировать работу системы в различных условиях и сценариях использования.

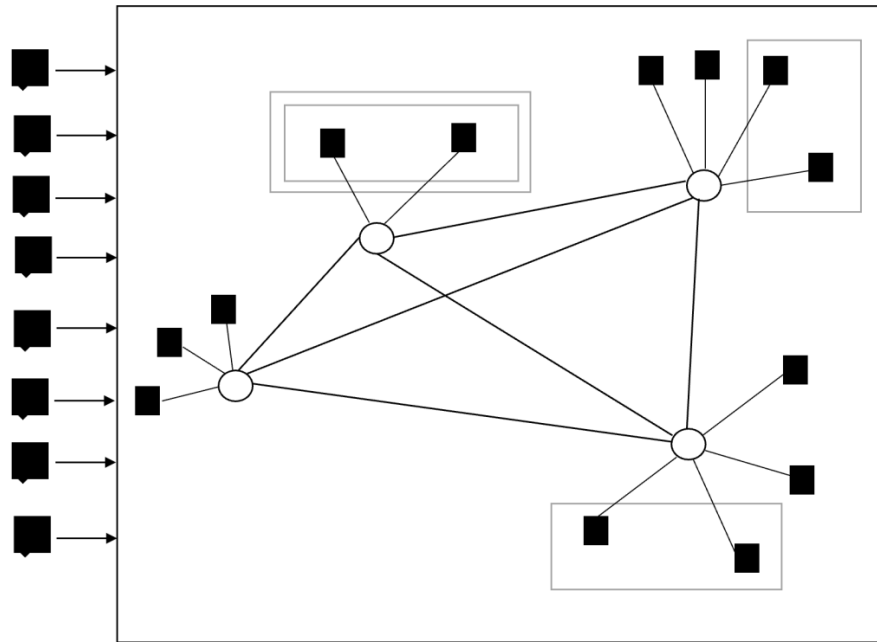


Рисунок 2 – Общая схема структуры системы имитационной модели РДВС

Количество элементов различных типов задается пользователем имитационной модели распределительной динамической вычислительной системы. На рисунке 2 показана общая схема структуры системы имитационной модели, на вход системы поступают задачи, система состоит из сетей, а сети из вычислительных узлов. Вычислительные узлы имитационной модели РДВС обладают рядом изменяемых параметров. Параметры, описывающие узел одного типа объединены в профили вычислительного узла, их можно либо задать для каждого узла, либо профиль будет присваиваться вычислительному узлу случайным образом. Профиль вычислительного узла включает следующие параметры: полный объем оперативной памяти, объем оперативной памяти, занятый на момент начала моделирования, процент загрузки процессора на момент начала моделирования, объем жесткого диска, свободное место на жестком диске на момент начала моделирования [72].

Для вычислительных задач создаются аналогичные профили задачи, включающие в себя требуемый для вычисления объем оперативной памяти, нагрузку на процессор при вычислении задачи.

Для одной РДВС может быть смоделировано множество допустимых конфигураций вариантов ресурсов системы, используемых в работе, максимальное количество которых определяются как:

$$N = np * npres * npnet + ne * nesom ,$$

где np – количество вычислительных узлов в системе; $npres$ – количество внесенных ресурсов; $npnet$ – количество типовых сетей; ne – количество задач в системе; $nesom$ – количество типовых сложностей задачи.

Параметры имитационной модели. Имитационная модель (ИМ) разработана с учетом динамического характера моделируемых систем в распределенной вычислительной среде. Конфигурация параметров ИМ позволяет учитывать основные аспекты, критически важные для выбора управления ресурсами системы, Общая схема ИМ представлена на рисунке 3.

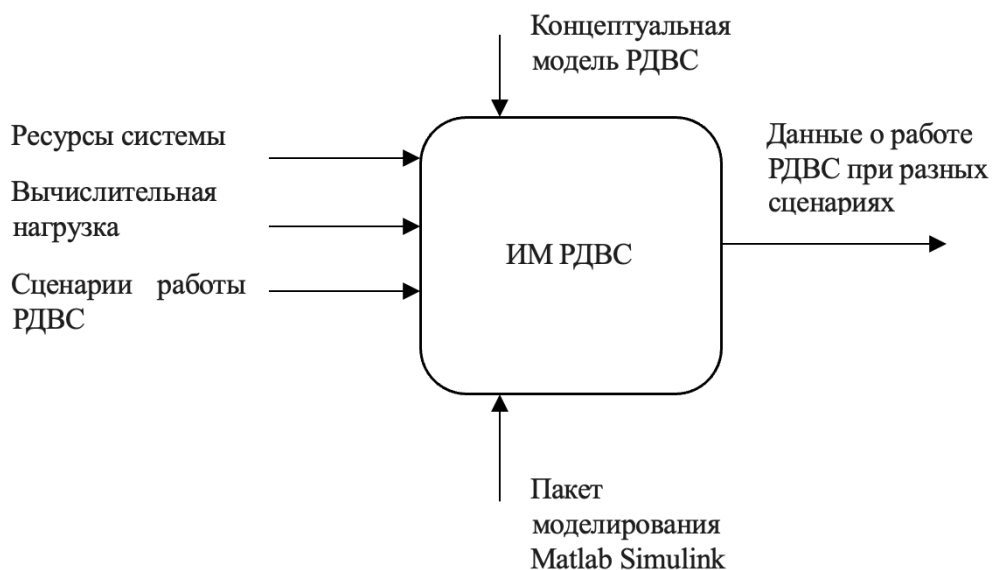


Рисунок 3 – Контекстная схема ИМ РДВС.

Каждая задача (вычислительная нагрузка) имеет объем входных данных – объем информации, необходимой для расчета. Сценарии РДВС для имитационной модели указаны в таблице 2.

Таблица 2. Сценарии для ИМ РДВС

Сценарий	
1	Высокая скорость чтения/записи, низкая сложность задачи, маленькое количество ядер (2 ядра), высокая пропускная способность (1000 Мбит/с)
Сценарий	
2	Низкая скорость чтения/записи, высокая сложность задачи, много ядер (16 ядер), низкая пропускная способность (100 Мбит/с)
Сценарий	
3	Средняя скорость чтения/записи, средняя сложность задачи, среднее количество ядер (8 ядер), средняя пропускная способность (500 Мбит/с).
Сценарий	
4	Высокая скорость чтения/записи, высокая сложность задачи, много ядер (16 ядер), высокая пропускная способность (1000 Мбит/с).
Сценарий	
5	Низкая скорость чтения/записи, низкая сложность задачи, маленькое количество ядер (2 ядра), низкая пропускная способность (100 Мбит/с)
Сценарий	
6	Средняя скорость чтения/записи, высокая сложность задачи, много ядер (16 ядер), высокая пропускная способность (1000 Мбит/с).
Сценарий	
7	Высокая скорость чтения/записи, низкая сложность задачи, много ядер (16 ядер), низкая пропускная способность (100 Мбит/с).
Сценарий	
8	Низкая скорость чтения/записи, высокая сложность задачи, много ядер (16 ядер), низкая пропускная способность (100 Мбит/с).
Сценарий	
9	Средняя скорость чтения/записи, низкая сложность задачи, среднее количество ядер (8 ядер), средняя пропускная способность (500 Мбит/с).
Сценарий	
10	Высокая скорость чтения/записи, средняя сложность задачи, много ядер (16 ядер), высокая пропускная способность (1000 Мбит/с).

На основе данных, полученных с имитационной модели, выполняется факторный анализ с целью выявления наиболее значимых параметров и факторов, влияющих на производительность и минимизацию выбросов в РДВС [73].

Для выполнения факторного анализа используется его традиционный метод максимального правдоподобия. Этот метод направлен на оценку параметров модели на основе имеющихся данных. Он позволяет определить наиболее значимые параметры для определения производительности и энергопотребления РДВС на основе экспериментальных данных с имитационной модели.

Для определения функции правдоподобия (likelihood function) в данной задаче управления ресурсами в распределенных динамических системах, учитывая

множество факторов, мы воспользовались многомерным подходом. Предположим, что каждый из параметров (скорость чтения, скорость записи, количество ядер, потребление энергии, пропускная способность, объем памяти, сложность задачи, производительность процессора, список задач, напряжение, частота, миграция, репликация, время передачи данных) подчинен некоторому вероятностному распределению.

В данном случае, можно представить функцию правдоподобия следующим образом:

$$\text{myLikelihood}(\theta \mid \text{данные}) = \prod_{i=1}^n P(\text{данные}_i \mid \theta),$$

где θ представляет собой вектор параметров, которые мы хотим оценить (в данной задаче, параметры, влияющие на производительность и энергопотребление), а данные_i – наблюдаемые данные для каждого из параметров.

Каждый из параметров может быть подчинен своему распределению в зависимости от его характеристик. Например, для непрерывных параметров, таких как скорость чтения, скорость записи, потребление энергии, пропускная способность, объем памяти, сложность задачи, производительность процессора, напряжение, частота, время передачи данных, используем нормальное распределение или другие соответствующие распределения, учитывая их характеристики и предполагаемую природу данных.

Для параметров, которые могут принимать ограниченные значения или являются категориальными (например, количество ядер, наличие миграции, репликации, список задач), используем дискретные распределения, такие как распределение Пуассона или распределение Бернулли.

Итак, функция правдоподобия будет представлена как произведение вероятностей каждого из наблюдаемых параметров при условии заданных параметров θ .

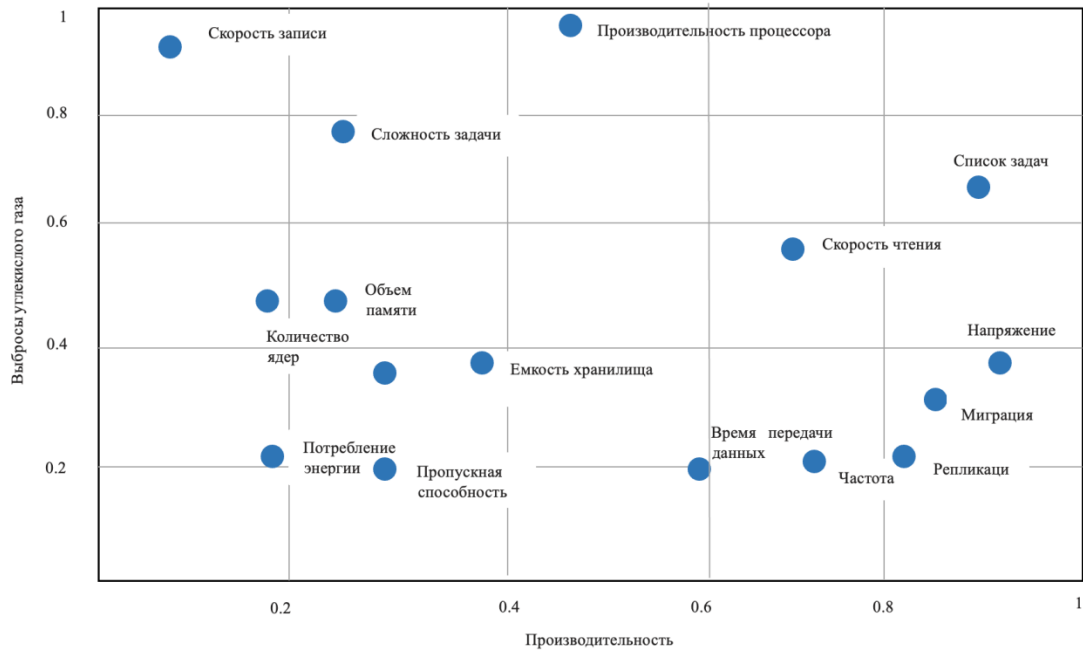


Рисунок 4 – Визуализация результатов анализа факторов влияния ресурсов РДВС на производительность и энергопотребление

На рисунке 4 представлена визуализация результатов факторного анализа. Выводы следующие:

Репликация и миграция задач оказывают существенное влияние на производительность. Оптимизация процессов репликации и миграции может привести к улучшению производительности и соответственно снижению энергопотребления.

Таким образом обобщая проведенную работу, предложим алгоритм действий при построении управляющего ресурсами механизма системы с учетом влияния на окружающую среду, приставлен на рисунке 5.



Рисунок 5 – Управление ресурсами в сложных системах с учетом влияния работы системы на окружающую среду

2.4 Гибридный подход к управлению ресурсами распределенных динамических вычислительных систем

Следующим шагом мы изучили современные научные источники на предмет методов управления выделенными выше параметрами. Выделили показавшие высокую эффективность.

2.4.1 Планирование задач инструментами миграции, репликации, задержки

Одна из проблем оптимизации управления в распределенных динамических вычислительных системах заключается в распределении входящих задач между вычислительными узлами и сокращении время отклика на задачи, избегая при этом перегруженных или недогруженных узлов. Поскольку миграция данных требует времени, очевидно, что нужно как можно больше локализовывать данные для задач, для сокращения времени отклика. При миграции данных в новые расположения в системе создаются их новые копии, называемые реплицированными данными. В целом, репликация данных повышает доступность

данных, тем самым достигая большей локальности и сокращая время отклика для следующих входящих задач [74-80].

Изучая эту проблему, мы обнаруживаем, что задачи в очереди, которые должны быть выбраны, – это те, которые будут выполняться на подходящем узле с лучшим временем отклика. Другими словами, алгоритм генерирует: локальность данных, то есть задача будет размещена непосредственно на вычислительном узле, содержащем все необходимые данные; или задача будет размещена на удаленном сервере, что дает минимальное время переноса данных; или задача будет отложена до тех пор, пока другой сервер не получит наилучшее время отклика, через местоположение данных или миграцию данных. Одновременно с этим нагрузка на узел также учитывается для минимизации энергопотребления всей системы при сохранения заданной производительности [81-95].

Чтобы лучше проиллюстрировать технологию, на рисунке 6 приведем пример схемы присвоения наборов данных вычислительным узлам.

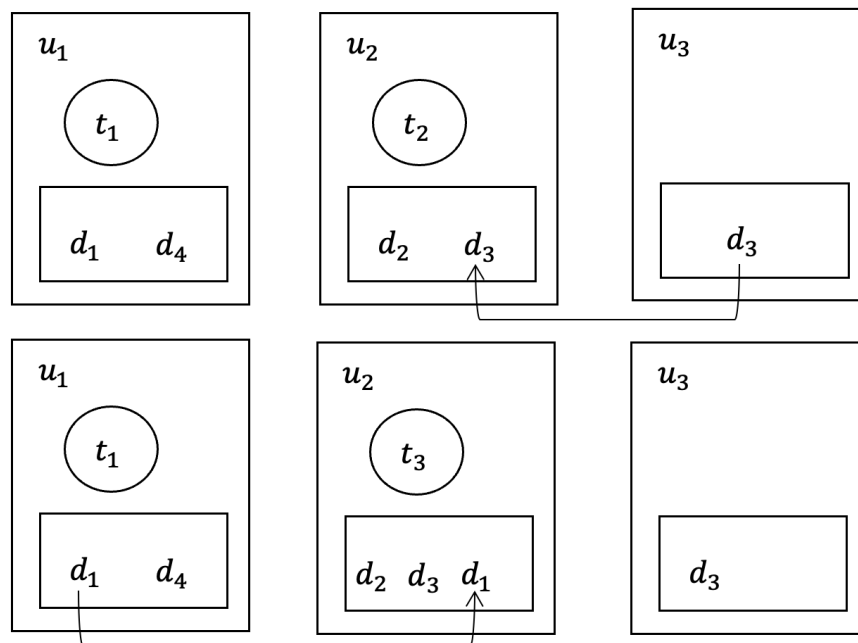


Рисунок 6 – Схема планирования задач миграции, репликации, задержки и вычислений.

Согласно представленной схеме, узел u_1 определен как оптимальный выбор для задачи t_1 , как показано на рисунке 5, поскольку она обеспечивает идеальную локальность данных с требуемыми наборами данных d_1 и d_4 , уже хранящимися на u_1 . Аналогичным образом, для задачи t_2 подбирается вычислительный узел u_1 в качестве наиболее эффективного решения, как показано на рисунке. Таким образом, выполнение t_2 на u_2 приведет к кратчайшему времени отклика из-за локально хранящихся требуемых данных d_2 и минимальному времени миграции для переноса требуемых данных d_3 из u_3 в u_2 . В результате t_1 и t_2 могут выполняться одновременно (параллельно). Кроме того, создается репликация d_3 в u_2 .

Наконец, алгоритм оценивает время отклика задачи t_3 на всех машинах. Используя этот подход, алгоритм предполагает, что предпочтительно отложить выполнение t_3 до тех пор, пока машина u_2 не станет доступной. Эта задержка представлена временным интервалом, обозначаемым как Δ . Несмотря на задержку, выполнение t_3 на u_2 , как ожидается, приведет к более низкому времени отклика по сравнению с назначением t_3 другим вычислительным узлам, что потребует большего времени миграции данных.

Важно отметить, что в случае, когда ограниченность ресурсов вычислительного узла становится существенной для выполнения назначенной задачи предлагаемый алгоритм продолжает работу либо пропуском этой узла в пользу другого, на котором может быть размещена текущая задача, или отсрочкой выполнения задачи до тех пор, пока у этого узла не появится достаточно ресурсов для размещения текущей задачи.

В результате мы минимизируем время отклика задачи при сохранении производительности и минимизации углеродного следа. Время отклика – это время, необходимое для завершения выполнения каждой задачи с момента ее поступления в очередь. Значение представляет собой комбинацию следующих метрик:

Время планирования (ST): время между поступлением задачи в очередь и ее выполнением.

Время задержки (Δ): время, в течение которого задача может ожидать доступности данного узла.

Время ожидания (WT): сумма времени планирования (ST) и времени задержки (Δ).

Время миграции (DMT): время, необходимое задаче для локального сбора всех необходимых удаленных наборов данных.

Время доступа к данным (DAT): время, необходимое задаче для считывания всех локальных необходимых наборов данных.

Время выполнения (ET): время выполнения задачи.

Общее время выполнения (TET): сумма времени переноса данных (DMT), времени доступа к данным (DAT) и времени выполнения (ET).

Время отклика (RT): сумма времени ожидания (WT) и общего времени выполнения (TET).

RAM [t_n] – целое число, представляющее объем памяти, необходимый для выполнения задачи t_n (Байт);

load – показатель загруженности ресурсов узла.

Задача сокращения времени отклика задачи t_i при запланированной в m_j может быть сформулирована следующим образом:

$$\begin{aligned} \min RT_{ij} &= \min(WT_{ij} + TET_{ij}) = \\ &= \min(ST_{ij} + \Delta_{ij} + DMT_{ij} + DAT_{ij} + ET_{ij}) \quad (13). \end{aligned}$$

Ограничения, связанные с нашей целевой функцией, показаны в уравнениях (2)–(3):

$$RAM[t_i] \leq RAM[m_j] - \sum_{t_k \in TP_j} RAM[t_k] \quad (10)$$

$$Load_{\min} \leq Load_j \leq Load_{\max} \quad (11)$$

Независимые пользователи отправляют задания на выполнение. Мы рассматриваем, что задачи поступают в режиме реального времени. Все онлайн-задачи совместно используют ресурсы и данные на серверах. Поскольку выполняемые задачи требуют больших объемов данных, с каждой задачей связаны два важных фактора: необходимые данные и ресурсы. Задачи выполняются

неупреждающим способом. Однако каждая задача определяется следующим образом:

$T = \{t_n\}$ набор задач, где t_n – n -я задача;

l_n обозначает длину n -й задачи (в миллионе инструкций – МІ);

$CPU[t_n]$ – целое число, представляющее количество MIPS, необходимое для выполнения задачи t_n ;

$V[t_n]$ – это целочисленное значение, представляющее общий размер всех необходимых наборов данных по задаче t_n ;

α_n – индекс конечного назначения узла (m_{α_n}) задачи t_n ;

ω_n – десятичное значение, представляющее время прибытия t_n ;

UR_{ni} – коэффициент использования CPU, определяющий, имеет ли машина u_i достаточное количество ресурсов для поддержки задачи t_n или нет.

Как упоминалось ранее, балансировка нагрузки является критически важным аспектом, который следует учитывать при разработке алгоритма планирования любых задач таким образом, чтобы оптимизировать использование ресурсов, максимизировать пропускную способность и минимизировать время отклика. Для этого определяем рабочую нагрузку каждого узла следующим образом:

$$load[u_i] = \frac{\sum_{t_j \in TP_i} l[t_j]}{PP[u_i]}, \quad (12)$$

где $load[u_i]$ – показатель загруженности ресурсов узла, указывающая на то, что u_i перегружен или недогружен. $load[u_i]$ вычисляется путем деления общего количества всех задач выполняемых в u_i на вычислительную мощность $PP[u_i]$.

Мы предполагаем, что на серверах изначально хранится фиксированное количество наборов данных. Каждый набор данных определяется следующим образом:

$D = \{d_k\}$ множество наборов данных, где d_k – k -й набор данных;

v_k целочисленное значение, обозначающее объем k -го набора данных (в Байтах);

$\psi = \{\psi_{ki}\}$ – это наборы данных для матрицы назначения машин. Уравнение (4) описывает вычисление матрицы ψ .

$$\psi = \begin{cases} 1 & \text{если } d_k \text{ входит в } u_i \\ 0 & \text{другое} \end{cases} \quad (13)$$

$F = \{f_{ki}\}$ – это присвоение наборов данных матрице задач. Мы устанавливаем матрицу F , потому что для выполнения задачи может потребоваться один или несколько наборов данных, и многие задачи могут использовать один и тот же набор данных. Матрица F генерируется в соответствии с уравнением (5).

$$f_{kn} = \begin{cases} 1 & \text{если } d_k \text{ требует } t_n \\ 0 & \text{в других случаях} \end{cases} \quad (14)$$

Для набора данных d_k может быть два варианта использования. Локальное использование – когда набор данных и его задача потребителя находятся на одном узле, в этом случае доступ к набору данных осуществляется локально. Удаленное использование – когда требуемый набор данных хранится в узле, отличном от того, на котором размещена задача; в этом случае требуется перенос данных из удаленного источника. Очевидно что из-за процесса миграции на время выполнения потребительской задачи влияет добавление времени переноса данных DMT , где DMT_{ni} – время миграции всех наборов данных, требуемых t_n , из их местоположений в u_i (u_i также является местом, где назначается t_n) [96-100].

2.4.2 Динамическое управление частотой и напряжением

Технология динамического управления частотой и напряжением позволяет еще больше снизить энергопотребление при ограничении времени выполнения задачи.

Если технология динамического управления частотой и напряжением применяется для снижения частоты выполнения высокопроизводительного ядра таким образом, чтобы высокопроизводительное ядро имело ту же производительность, что и низкопроизводительное ядро, высокопроизводительное ядро по-прежнему потребляет больше энергии, чем низкопроизводительное ядро [101]. Из-за требований к распределенности в РДВС, то есть задачи могут выполняться или передаваться только одна за другой, фактическое время начала ST_i задачи $task_i$ может быть позже, чем время ее готовности RT_i . В этом случае предшественники задачи $task_i$ могут выполняться с меньшей частотой, чтобы использовать пробел между ST_i и RT_i .

Простая реализация алгоритма динамического управления частотой и напряжением выглядит следующим образом: для каждой задачи мы пробуем каждый уровень частоты выполнения в порядке возрастания и перепланируем задачи до тех пор, пока не найдем частоту выполнения, которая может удовлетворить ограничение по времени завершения выполнения задачи. Если частота выполнения задачи изменяется, все расписание может быть изменено из-за требований приоритета задач. Нам нужно перепланировать все задачи после одного единственного изменения частоты выполнения задачи. Поэтому предложена другая реализацию алгоритма динамического управления частотой и напряжением для снижения временной сложности [102].

Предлагаемый алгоритм динамического управления частотой и напряжением: для каждой локальной задачи, если есть временной промежуток времени между временем окончания этой задачи $task_i$ и временем начала следующей задачи $task_j$ на том же локальном ядре, мы перебираем каждый уровень частоты выполнения в порядке возрастания (всего имеется M уровней частоты) до тех пор, пока не найдем частоту выполнения, которая не откладывает время начала задачи $task_j$.

Ниже представлен алгоритм динамического управления частотой и напряжением, который содержит:

Входные данные: результат планирования, сгенерированный алгоритмом переноса задач.

Выходные данные: расписание задач с новым назначением частоты выполнения для задач.

1. Для каждой локальной задачи $task_i$ используем
2. $flag = 0; m = 1;$
3. При флаге $flag == 0$ и $m < M$ выполняем
4. Вычисляем новое время завершения FT_i^{new} если $task_i$ выполняется с использованием m -й с использованием;
5. Если \exists следующая задача $task_j$ на том же ядре
6. $lim_1 = ST_j;$
7. Иначе $\{ \% task\ task_j \}$ последняя задача в этом ядре}
8. $lim_1 = T^{max};$

9. **Завершить если**
10. Если $task_j \notin exit\ tasks$ тогда
11. $lim_2 = \min_{task_j \in succ(task_j)} ST_j$;
12. **Иначе**
13. $lim_2 = T^{max}$;
14. **Завершаем, если**
15. Если $FT_i^{new} \leq lim_1$ и $FT_i^{new} \leq lim_2$ тогда
16. $flag = 1$;
17. Назначить m -й частоту для задачи $task_i$;
18. Обновите время завершения $task_j$;
19. **Конец если**
20. **Конец пока**
21. **Конец для**

В качестве примера граф задач на рисунке 6, а результат планирования задач, сгенерированный алгоритмом динамического управления частотой и напряжением (ДУЧИН), показан на рисунке 7.

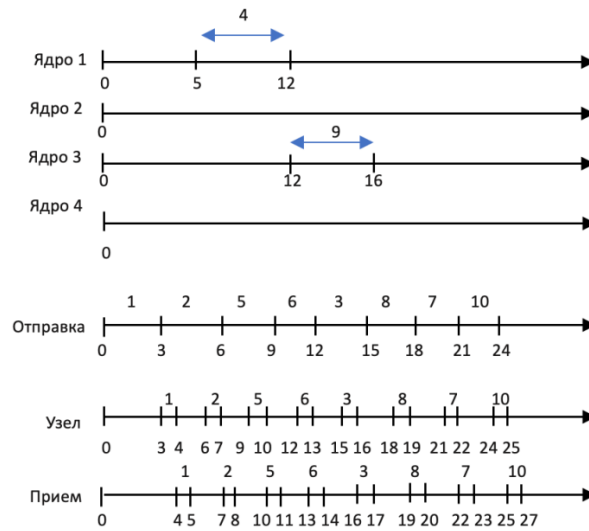


Рисунок 7 – Пример работы алгоритма ДУЧИН. Выполнено распределение вычислительных задач по ядрам на каждый такт работы, подобрана минимально допустимая частот работы процессора.

Предполагаем, что для каждого ядра существует $M = 3$ уровня рабочей частоты. Мы устанавливаем коэффициенты масштабирования частоты как $a_{k,1} = 0.2$, $a_{k,2} = 0.5$, $a_{k,3} = 1$ для $k = 1, 2, 3$. $\gamma_k = 2$ для $k = 1, 2, 3$.

2.4.3 Подход к управлению ресурсами распределенных динамических вычислительных систем

Исходя из проведенного анализа, предложен подход, который при минимальных затратах, может быть максимально полезным. Опишем основные шаги и компоненты алгоритма, представленного на рисунке 8, а также объясним, как он может способствовать оптимизации использования ресурсов и учету экологических параметров.



Рисунок 8 – Концептуальная схема гибридного метода управления ресурсами РДВС

На первом этапе алгоритма производится сбор и анализ информации о раннем состоянии работы системы. Затем осуществляется мониторинг текущего состояния системы. Это включает в себя оценку доступных ресурсов, уровень загрузки узлов, задачи в очереди и другие важные параметры системы. Отметим, что задачи могут поступать как от внешних источников, так и быть результатом выполнения предыдущих задач.

На следующем этапе проводится проверка доступности и производительности узлов, а также анализ уровня нагрузки и задержек в сети. Этот анализ выполняется с использованием вероятностных функций, построенных на статистических данных, что позволяет получить объективное представление о текущем состоянии системы.

На основе данных, собранных на предыдущих этапах, алгоритм по средством ранее описанной целевой функции производит расчет времени выполнения задач и необходимости их миграции между узлами системы.

Также одним из ключевых элементов алгоритма является применение технологии динамического изменения напряжения и частоты процессоров. Это позволяет определить оптимальную частоту работы процессоров, обеспечивая баланс между производительностью и энергопотреблением, что имеет важное значение для учета экологических параметров [103].

Таким образом, предложено гибридное управление ресурсами в распределенной динамической вычислительной системе с целью минимизации углеродного следа и сохранения производительности включающий в себя: алгоритм планирования задач миграции, репликации, задержки и вычислений; технологию динамического управления чистотой и напряжением работы процессоров; обнуляющий нейросетевой алгоритм управления ресурсами РДВС. Опишем основные этапы этого подхода как

1. Сбор статистических данных о работе РДВС. На этом шаге собираются данные о работе узлов системы в различных условиях нагрузки (решения задач), такие как частота и напряжение работы процессоров, миграции и репликации.
2. Определение состояния системы. Этот шаг включает в себя мониторинг текущего состояния вычислительной системы, включая состояние вычислительных узлов, их производительность, доступные ресурсы, загрузку и т.д.
3. Список новых задач. Новые задачи, которые поступают в систему, должны быть зарегистрированы. Это включает в себя информацию о требованиях к вычислениям, сроках и приоритетах задач.
4. Расчет производительности узлов и каналов. На основе текущего состояния системы и требований задачи производится расчет работоспособности вычислительных узлов и сетевых каналов. Это позволяет определить, какие узлы и каналы могут быть использованы для выполнения задачи.
5. Расчет времени и необходимости миграции и репликации. В случае, если требуется миграция задачи с одного узла на другой или создание ее реплики,

производится расчет времени, необходимого для выполнения этих операций, и их целесообразности с учетом минимизации углеродного следа и сохранения производительности.

6. Планирование задач: на основе всех рассчитанных данных и с учетом целей минимизации углеродного следа и сохранения производительности, производится планирование выполнения задач. Определяется, на каких узлах и в каком порядке будут выполняться задачи, какие ресурсы будут выделены каждой задаче и как будет управляться энергопотребление. Для реализации этого шага в следующей главе исследована возможность применения нейронных сетей.

Таким образом, путем имитационного моделирования и факторного анализа найдены взаимосвязи в системе и определены инструменты управления ресурсами в РДВС. Построена структура подхода минимизации углеродного следа (энергоэффективности) при работе распределенных динамических вычислительных сетей и отдельных узлов с помощью алгоритмов миграции, репликации, задержки и расчета необходимой частоты и напряжения, представлена [104].

Результаты Главы 2

Моделирование системы является фундаментальным этапом при разработке методов оптимального управления в распределенных динамических вычислительных системах (РДВС). В данном разделе мы провели следующие ключевые шаги:

Создали математическую модель РДВС, выделив такие важные аспекты, как пропускная способность, время отклика, загрузка ресурсов, мощность вычислительных ресурсов, объем памяти и энергопотребление. Эти параметры играют критическую роль в обеспечении эффективной работы системы.

Определили ключевые параметры частоту, напряжение и планирование задач с помощью имитационного моделирования и факторного анализа. Учет этих параметров позволит обеспечить баланс между энергопотреблением и производительностью работы РДВС.

Мы предложили гибридный подход управления ресурсами в РДВС с учетом экологических аспектов, таких как снижение углеродного следа, при оптимизации ресурсов и производительности системы.

Полученные данные и выводы будут использованы в следующей главе диссертации для разработки эффективного алгоритма управления и оптимизации системы с учетом требований к ресурсам и производительности. Этот раздел является ключевым шагом на пути к созданию устойчивого и высокоэффективного управления ресурсами в РДВС.

3 Гибридный метод управления ресурсами в распределенных динамических системах

В данной главе проводится исследование возможности и эффективности решения, поставленной задачи классическими методами оптимизации и методами нейронных сетей. Рассмотрены основные концепции и принципы работы существующих нейронных сетей. Решаемую оптимизационную задачу можно определить как многоэкстремальную, многоэкстремальную задачу с ограничениями и несколькими управляющими параметрами. Функция, описывающая эту задачу, является непрерывной в связи с работой РДВС в режиме постоянного функционирования.

3.1 Исследование методов решения поставленной задачи

Существует множество потенциальных методов решения поставленной задачи управления ресурсами распределенных динамических систем [105-123]:

Градиентный спуск – метод минимизации функции, который использует градиент функции для поиска локального минимума.

Стохастический градиентный спуск – это вариант градиентного спуска, который используется в машинном обучении для обучения моделей на больших наборах данных. В отличие от обычного градиентного спуска, стохастический градиентный спуск использует только случайно выбранные подмножества данных для обновления параметров модели.

Метод Ньютона – метод оптимизации, который использует вторую производную функции для нахождения глобального минимума. Этот метод требует больше вычислительных ресурсов, чем градиентный спуск, но может сходиться быстрее.

Методы оптимизации на основе градиентов высших порядков – это методы, которые используют производные высших порядков для оптимизации функции. Примеры включают методы Ньютона-Рафсона и методы оптимизации на основе кривизны.

Методы оптимизации без градиентов – это методы оптимизации, которые не требуют вычисления градиента функции. Примеры включают методы случайного поиска и генетические алгоритмы.

Методы оптимизации с ограничениями – это методы, которые оптимизируют функцию при условии, что ее параметры должны удовлетворять определенным ограничениям. Примеры включают методы проекции градиента и методы барьерных функций.

Метод наискорейшего спуска. Он заключается в том, чтобы двигаться в направлении антиградиента функции, то есть в направлении наискорейшего убывания функции. Для этого на каждом шаге вычисляется градиент функции и происходит шаг в противоположном направлении. Этот процесс повторяется до тех пор, пока значение функции не перестанет уменьшаться или пока не будет достигнут критерий останова.

Методы оптимизации на основе эволюционных алгоритмов, такие как генетические алгоритмы, эволюционные стратегии и другие. В этих методах используется идея эволюции, где решения представляются в виде генетических кодов, которые мутируют и скрещиваются, чтобы получить новые потенциально лучшие решения.

Методы оптимизации на основе комбинаторной оптимизации, которые используются для решения задач, связанных с комбинаторными проблемами, такими как задача коммивояжера и задача раскроя. В этих методах используется идея перебора всех возможных решений и выбора наилучшего.

Метод сопряженных градиентов. Использует направления, полученные из предыдущих градиентов функции, для определения направления поиска минимума. Хорошо работает для оптимизации квадратичных функций, но может быть менее эффективным для более сложных функций.

Метод имитации отжига. Использует стохастический процесс, в котором случайные изменения переменных функции используются для поиска минимума. Метод отжига хорошо работает в случаях, когда оптимизируемая функция имеет множество локальных минимумов.

Таким образом, классические методы оптимизации показывают низкую эффективность при решении задач управления ресурсами в распределенных динамических системах по нескольким причинам:

1. Динамические изменения и неопределенность. В распределенных динамических системах условия могут изменяться со временем, и могут возникать случайные факторы или неопределенность в параметрах системы. Классические методы оптимизации могут не учитывать эту динамику и неопределенность, что делает их результаты неустойчивыми.

2. Временные ограничения. Задачи управления ресурсами в распределенных системах часто связаны с выполнением задач в определенные сроки. Классические методы оптимизации могут не учитывать временные ограничения или не уметь эффективно работать с ними.

3. Вычислительная сложность. Классические методы оптимизации, особенно при работе с большими объемами данных и сложными зависимостями, могут иметь высокую вычислительную сложность. Это означает, что временные затраты на решение задачи оптимизации с использованием таких методов могут быть значительными, особенно когда рассматривается большое количество параметров или сложные функции целей.

4. Итерационные процессы. Некоторые классические методы оптимизации могут потребовать большого числа итераций для сходимости к оптимальному решению. Это может увеличивать время выполнения, особенно при работе с большими объемами данных.

5. Необходимость воспроизводимости. В некоторых приложениях, таких как исследования и разработки в области науки о данных и машинного обучения, требуется воспроизводимость результатов. Классические методы оптимизации могут сталкиваться с проблемой воспроизводимости из-за своей случайной или стохастической природы.

Из-за перечисленных сложностей часто используются эвристические методы, метаэвристики, агентные системы и нейросетевое моделирование для решения подобных задач управления ресурсами в распределенных динамических

системах, так как они могут более гибко адаптироваться к сложным и динамическим условиям.

Далее было принято решение рассмотреть в качестве инструмента нейронные сети, это было обусловлено их способностью моделировать сложные, нелинейные взаимосвязи между параметрами системы, обучаемостью на основе опыта и адаптивностью к изменяющимся условиям, что важно в динамических средах управления.

Нейронные сети представляют собой мощный инструмент в области обработки данных и машинного обучения. Эти вычислительные системы, вдохновленные биологическими нейронами, способны моделировать сложные взаимосвязи и выполнять разнообразные задачи, включая классификацию, регрессию, и даже управление ресурсами в распределенных динамических вычислительных системах (РДВС) [124]. В данном разделе рассматриваются ключевые концепции нейронных сетей и метода обнуления Чжана (НСО), а также их значимость для исследования оптимального управления ресурсами в РДВС.

Нейронные сети – это алгоритмы машинного обучения, которые пытаются моделировать функционирование человеческого мозга. Они состоят из множества связанных между собой нейронов, каждый из которых выполняет простую задачу, такую как суммирование входных сигналов и выдачу результата в виде выходного сигнала [125]. Нейронные сети классифицируются по различным критериям, например, по количеству слоев (однослойные, многослойные), по типу соединения нейронов (сверточные, рекуррентные), по типу обучения (наблюдаемое, ненаблюдаемое). Некоторые из наиболее распространенных [126-132]:

Прямое распространение нейронные сети – это наиболее распространенный тип нейронных сетей, в которых информация передается от входных нейронов к выходным нейронам через скрытые слои. Примерами прямого распространения нейронных сетей являются многослойные персептроны и сверточные нейронные сети.

Многослойные персептроны – это класс искусственных нейронных сетей прямого распространения, состоящих как минимум из трех слоёв: входного,

скрытого и выходного. За исключением входных, все нейроны используют нелинейную функцию активации. При их обучении используется обучение с учителем и алгоритм обратного распространения ошибки.

Сверточные нейронные сети – используются для анализа изображений и других типов данных сеток. Они используют операцию свертки для извлечения признаков из входных данных и могут быть эффективно использованы для классификации, детектирования объектов и распознавания образов.

Рекуррентные нейронные сети – это тип нейронных сетей, в которых имеется обратная связь между выходами и входами, что позволяет использовать информацию о предыдущих состояниях для принятия решений. Рекуррентные нейронные сети широко применяются в задачах, связанных с обработкой последовательностей данных, таких как анализ текста или распознавание речи. Тип рекуррентных нейронных сетей представлен такими сетями:

1. Простые рекуррентные нейронные сети (Simple RNN): Простые RNN имеют обратные связи, позволяющие информации передаваться от предыдущих шагов времени к будущим. Однако у них есть проблема исчезающего градиента (vanishing gradient problem), из-за которой они имеют тенденцию забывать долгосрочные зависимости в данных.

2. Долгая краткосрочная память (Long Short-Term Memory, LSTM): LSTM-сети были разработаны для решения проблемы исчезающего градиента в простых RNN. Они содержат специальные структуры памяти, такие как ячейки состояния, ворота (gate), которые помогают им запоминать и управлять информацией на протяжении длительных последовательностей.

3. Сети долгой памяти и краткосрочной памяти (Gated Recurrent Unit, GRU): GRU являются вариантом LSTM, предложенным в ответ на сложность реализации LSTM. Они имеют структуры ворот, позволяющие им сохранять долгосрочные зависимости и управлять потоком информации внутри сети.

4. Двухнаправленные рекуррентные нейронные сети – это тип рекуррентных нейронных сетей, который способен учитывать контекст как с предыдущих, так и с последующих временных шагов. Они имеют

двунаправленную архитектуру, что позволяет им иметь доступ к информации как из прошлого, так и из будущего.

5. Обнуляющие нейронные сети Чжана – предназначены для итеративного вычисления приблизительного решения во времени из нескольких начальных значений. Он делает это прогнозно с течением времени и с высокой точностью. Успех ZNN зависит от идеи Чжана об установлении экспоненциального затухания для связанной функции ошибки [133].

Самоорганизующиеся карты Кохонена – это нейронные сети, которые используются для кластеризации данных и визуализации многомерных данных в двумерном пространстве.

Нейронные сети Хопфилда – это нейронные сети, которые используются для решения задач оптимизации и ассоциативной памяти.

Глубокие вероятностные нейронные сети – это нейронные сети, которые используются для моделирования вероятностных распределений данных и решения задач, таких как классификация изображений или генерация текста.

Генеративно-сопоставительные нейронные сети – используются для генерации новых данных на основе существующих. Они состоят из двух нейронных сетей - генератора и дискриминатора, которые конкурируют друг с другом в процессе обучения.

Сети преобразований – используются для обработки текстов и других типов последовательностей данных. Они используют механизмы внимания для эффективного управления информацией внутри сети и показывают высокую производительность в задачах обработки естественного языка.

Далее было проведено сравнение нейронных сетей на основе изученного материала по критериям динамичности, вычислительной сложности, работы с дискретными данными и работы в режиме реального времени [134]. Результаты представлены в таблице 3.

Таблица 3 – Сравнение нейронных сетей

Название сети	Динамичность	Вычислительная легкость	Работа с дискретными данными	Работа в режим реального времени
Простые рекуррентные нейронные сети	Высокая	Умеренная	Да	Может быть сложно из-за вычислительных требований
Долгая краткосрочная память	Очень высокая	Умеренная	Да	Сложно из-за вычислительных требований
Сети долгой памяти и краткосрочной памяти	Очень высокая	Умеренная	Да	Сложно из-за вычислительных требований
Двунаправленные	Очень высокая	Умеренная	Да	Возможна при оптимизации
Обнуляющие нейронные сети Чжана	Очень высокая	Высокая	Да	Работают в режиме реального времени
Самоорганизующиеся карты Кохонена	Низкая	Высокая	Да	Возможна при оптимизации
Нейронные сети Хопфилда	Низкая	Высокая	Да	Возможна при оптимизации
Глубокие вероятностные нейронные сети	Высокая	Низкая	Да	Сложно из-за вычислительных требований
Генеративно- состязательные нейронные сети	Высокая	Низкая	Да	Сложно из-за вычислительных требований
Сети преобразований	Высокая	Умеренная	Да	Возможна при оптимизации

Выводы:

Обнуляющие нейронные сети Чжана (НСО) – это метод, который позволяет находить неизвестные значения в непрерывно изменяющихся данных на основе

известных равноудаленных данных. Его применение особенно эффективно в случаях, когда информация поступает в режиме реального времени в форме дискретных равноудаленных сигналов.

Основные характеристики и преимущества методов НСО в контексте распределенных динамических вычислительных систем включают в себя:

Работа с данным в реальном времени. Поскольку НСО способен анализировать дискретные равноудаленные сигналы от датчиков, он идеально подходит для работы в реальном времени. Он может предсказывать и оптимизировать значения, основываясь на предыдущих данных, что делает его важным инструментом для систем мониторинга и управления.

Скорость и точность. Методы НСО обладают выдающейся скоростью и точностью в численных матричных вычислениях. Это позволяет им быстро адаптироваться к изменяющимся данным и быстро реагировать на изменения в окружающей среде.

Прогнозирование во времени. НСО способен итеративно прогнозировать значения на следующем временном шаге, используя предыдущие данные. Это прогнозирование происходит с высокой точностью и позволяет системе быть готовой к будущим событиям [135-137].

3.2 Алгоритм гибридного подхода на базе обнуляющих нейронных сетей

Гибридный метод управления ресурсами основан на аппарате обнуляющей нейронной сети по причинам того, что НСО предоставляет преимущества в контексте управления динамическими системами в реальном времени. Важно отметить, что НСО работает на архитектуре, аналогичной рекуррентным нейронным сетям, представлено на рисунке 9. Архитектура рекуррентной нейронной сети способна обрабатывать последовательные данные и сохранять внутреннее состояние для учета контекста. Однако НСО обладает важными дополнениями, такими как возможность обнуления последовательно хранящихся и передаваемых данных, что также позволит минимизировать количество

необходимых вычислений для непосредственного управления ресурсами, а соответственно снизит углеродный след [138].

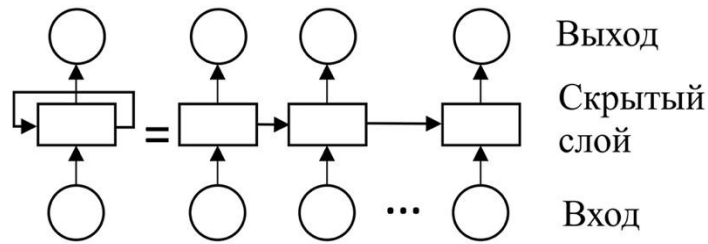


Рисунок 9 – Архитектура рекуррентной нейронной сети

Типичный алгоритм НСО требует семи отдельных шагов для его настройки. Построение алгоритмов НСО начинается с формирования уравнения ошибки, которое необходимо для определения ошибки прогнозирования, допускаемой нейронной сетью на каждом этапе обучения и условия, что функция ошибки уменьшается экспоненциально быстро. Функция ошибок затем соединяется со сходящейся конечно-разностной формулой, чтобы функцию итерации, который надежно предсказывает будущее состояние системы на основе текущих и более ранних данных о состоянии.

Дана непрерывная изменяющаяся во времени векторная модель

$$F(A(t), B(t), x(t), \dots) = g(t, C(t), u(t), \dots) \quad (15)$$

с нестационарным неизвестным вектором или матрицей $x(t)$ и с изменяющимися во времени матрицами $A(t), B(t), C(t), \dots$ и изменяющиеся во времени векторы $u(t), \dots$ которые известны при дискретных равноудаленных временных экземплярах t_i для $i \leq k$ и $k = 1$ из данных работы системы. Равноудаленные данные идеально подходят для дискретизированного НСО. Задача нейросети состоит в том, чтобы найти $x(t_{k+1})$ точно и в режиме реального времени от более раннего $x..$ значения и более ранних матричных и векторных данных. Тогда данные динамически меняющейся матрицы $A(t)$ и другие, возможно, придется дополнить чтобы оставаться совместимыми с развернутым вектором собственных данных $x(t)$, а также с любыми другими векторами или матрицами $u(t)$.

Алгоритм НСО

Шаг 1: Из уравнения модели (1) сформируем функцию ошибки

$$E(t) = F(A(t), B(t), x(t), \dots) - g(t, C(t), u(t), \dots) \quad (16)$$

которая была бы одинаково равна нулю, т.е. $E(t) = 0$ для всех t , если $x(t)$ решает (16).

Шаг 2: Возьмем производную $E(t)$ функции ошибки $E(t)$ и определим ее экспоненциальный распад так, чтобы

$$E(t) = -\eta E(t) \quad (17)$$

для некоторой константы $\eta > 0$ в случае нейронных сетей Чжана (НСО).

Шаг 3: Решим экспоненциально затухающее уравнение ошибки дифференциального уравнения (3) шага 2 алгебраически для $x(t_k)$.

Шаг 4: Выберем перспективную сходящуюся конечно-разностную формулу для требуемого порядка ошибок усечения $Q(\tau^j)$ которая выражает $x(t_k)$ в терминах $x(t_{k+1}), x(t_k), \dots, x(t_k)x(t_{k+1-j})$, т.е. в терминах $j + 1$ известных точек данных из таблицы известных сходящихся конечно-разностных формул.

Шаг 5: Приравняем производные $x(t_k)$ на шагах 3 и 4 и тем самым полностью избавимся от $x(t_k)$.

Шаг 6: Решим производное свободное линейное уравнение, полученное на шаге 5 для $x(t_{k+1})$ и выполним итерацию.

Шаг 7: Увеличим $k + 1$ до $k + 2$ и вверх все данные Шага 6; затем решим обновленную рекурсию для $x(t_{k+2})$. И повторяем до тех пор, пока $t \dots \geq t_f$ [139].

Таким образом, аппарат обнуляющей нейронной сети выполняет вычисления на каждом этапе с учетом параметров системы и задач.

3.2.1 Структура нейронной сети обнуления для гибридного метода

В данном алгоритме НСО выполняет важные функции, учитывая параметры производительности и энергопотребления системы прогнозирует следующий шаг и выдает параметры для частоты, напряжения и необходимости миграции, репликации, задержки. Рассмотрим, каким образом НСО подсчитывает решения на каждом этапе алгоритма:

На этапе сбора и анализа информации о текущем состоянии системы НСО анализирует доступные ресурсы, загрузку узлов и параметры задач, учитывая как внешние, так и внутренние источники задач. НСО использует эту информацию для определения текущей конфигурации системы и оценки загруженности.

При проверке доступности и производительности узлов НСО анализирует уровень доступности и производительности каждого узла в системе. Это позволяет определить, какие ресурсы могут быть использованы для выполнения задач.

При расчете времени выполнения и миграции задач НСО выполняет расчет времени выполнения задач на основе данных, собранных на предыдущих этапах. Он также определяет, есть ли необходимость в миграции задач между узлами системы для балансировки производительности и энергопотребления.

При применении технологии динамического изменения частоты и напряжения НСО оптимизирует эту технологию, чтобы определить оптимальную частоту и напряжения работы процессоров.

Таким образом, адаптивное управление ресурсами распределенных динамических вычислительных систем на базе аппарата обнуляющей нейронной сети выполняет вычисления и оптимизацию на каждом этапе алгоритма с учетом параметров системы и задач, а также обеспечивает баланс между производительностью и энергопотреблением для достижения экологической устойчивости.

На рисунке 10 представлен процесс моделирования разработанного подхода управления ресурсами в РДВС, основанного на нейронных сетях обнуления Чжана (НСО).

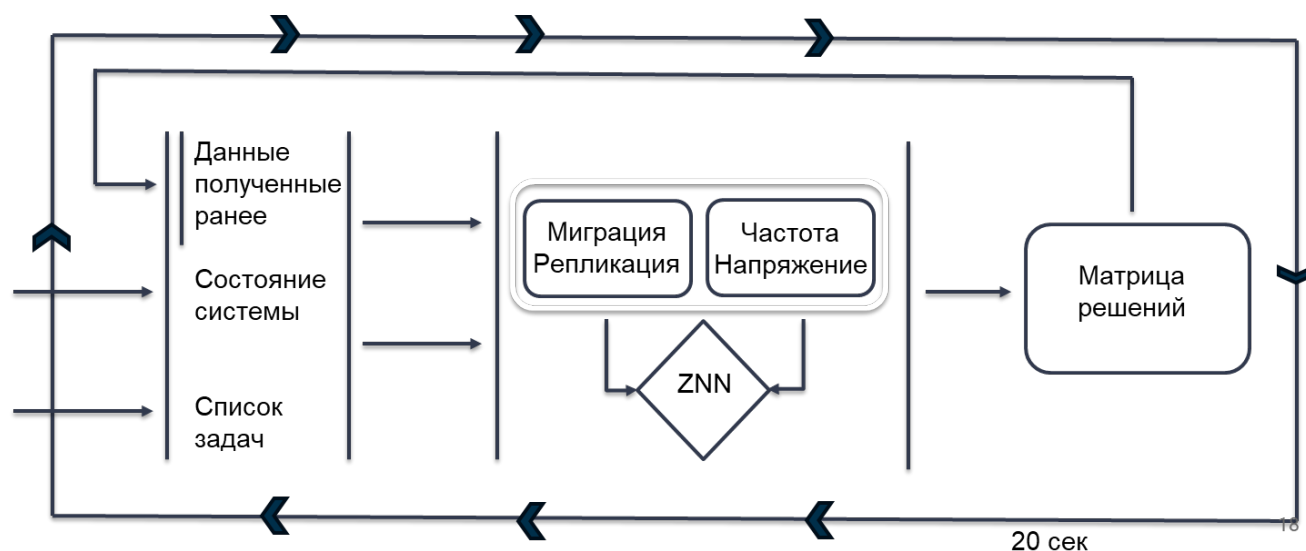


Рисунок 10 – Алгоритм работы НСО

3.2.2 Подготовка данных

Для тестирования эффективности алгоритма мы подготовим набор реальных данных, собранных с РДВС NorduGrid, состоящую из более 2,7 миллионов вычислительных узлов, распределенных в 141 географических точках [140]. Данные о вычислительных узлах были дополнены оценкой выбросов углекислого газа в результате выработки энергии необходимой для работы этих вычислительных узлов. Показатели выбросов углекислого газа в атмосферу были получены из интегрированной базы данных по выбросам и генерирующим ресурсам (eGRID) представляющей из себя комплексный источник данных. Данные включают экологические характеристики, выбросы, уровень выбросов, выработку, тепловложение (например, энергию от сгорания) и многие другие атрибуты для всех подключенных к сети электрогенерирующих установок (ЭГУ) на объектах, паспортная генерирующая мощность которых составляет не менее 1 МВт.

Последовательность сбора и обработки данных включает следующие шаги:

1. Сбор необходимых данных для дальнейшего анализа и использования в экспериментальном исследовании.

Набор данных относится к области планирования и расписания задач. Он предоставляет информацию о различных задачах, которые необходимо выполнить, а также о доступных ресурсах для их выполнения [141].

Типичный набор данных включает в себя следующие основные элементы:

А. Задачи: Каждая задача имеет определенные характеристики, такие как время выполнения, сроки выполнения, приоритеты, зависимости от других задач и требования к ресурсам.

В. Ресурсы: Ресурсы представляют собой вычислительные узлы, которые могут быть использованы для выполнения задач. Они могут быть ограничены по доступности, пропускной способности, энергопотреблению или другими параметрами.

С. Ограничения: Набор данных также может содержать ограничения, связанные с использованием ресурсов или выполнением задач. Это могут быть ограничения на максимальное количество задач, которые можно выполнить одновременно, ограничения на использование определенных ресурсов или другие ограничения, связанные с временем или приоритетами.

Набор данных представлен в форматах отдельных таблиц и баз данных.

2. Очистка данных: Данные, собранные из различных источников, могут содержать ошибки, пропуски или выбросы. Поэтому перед анализом необходимо провести их очистку, чтобы убедиться в правильности данных.

В ходе исследования была проведена очистка данных, собранных из двух различных источников, с целью обеспечить их правильность и надежность. В первую очередь была проведена детальная проверка данных, включая изучение их структуры, типов переменных и связей между ними.

В процессе анализа было выявлено несколько проблем, включая наличие пропущенных значений, выбросов и ошибочных данных. Были применены соответствующие методы для обнаружения этих проблем и их последующей обработки. Пропущенные значения были заполнены с использованием подходящих стратегий, таких как заполнение средним или медианным значением, либо удаление соответствующих наблюдений в случае неприемлемой потери

информации. Выбросы и ошибочные данные были исследованы внимательно, чтобы определить их природу и принять решение о дальнейшей обработке, такой как удаление выбросов или коррекция ошибочных значений. Очистка данных производилась с использованием таких инструментов MatLab как приложение Data Cleaner, Live Editor, Smoothdata и Fillmissing, Signal Analyser, Signal Labeler и Image Labeler.

Кроме того, была проведена нормализация данных, чтобы привести их к единому масштабу и обеспечить сопоставимость между различными переменными [142].

В результате очистки данных были получены пригодные для дальнейшего анализа данные, что позволило приступить к проведению экспериментов и исследованию эффективности метода оптимального управления ресурсами в распределенной динамической системе на базе НСО.

3. После очистки данных можно приступить к их обработке. Здесь применяется метод оптимального управления ресурсами распределенных динамических систем на базе НСО.

Важным шагом в обработке данных была подготовка данных для моделирования и анализа. Это включало разделение данных на обучающую и тестовую выборки, чтобы оценить производительность модели на независимых данных. Также было проведено разбиение данных на признаки и целевую переменную, необходимые для построения моделей и прогнозирования [143].

В результате обработки данных были получены чистые, структурированные и готовые к использованию данные, которые можно применять для построения моделей, проведения анализа и получения выводов в рамках исследования эффективности метода оптимального управления ресурсами в распределенной динамической системе на базе обнуляющих нейронных сетей.

3.2.3 Реализация алгоритма

Для настройки нашего алгоритма в контексте НСО мы используем ограниченную нелинейную оптимизацию для изменяющихся во времени матричных потоков:

Задачи оптимизации при линейных ограничениях, таких как

Найти $\max f(x)$ или $\min f(x) \in R$, чтобы $Ax = b \in R^m$ для $A_{m,n} \in R$ и $x \in R^n$ со статическими элементами может быть решена с помощью множителей Лагранжа, где решается дифференциальное уравнение Лагранжа $\nabla_{x,\lambda} \mathcal{L}(x, \lambda) = 0$ от $\mathcal{L}(x, y) = f(x) \pm \lambda(Ax - b)$ с ∇ обозначает функцию градиента. Здесь градиенты будут использоваться в виде векторных строк соответствующих частных производных. Ограниченные крайние точки f являются нулями дифференциального уравнения Лагранжа выше, а значения λ являются множителями Лагранжа, которые решают задачу.

Здесь мы хотим решить DE Лагранжа для изменяющихся во времени данных $A(t), x(t), b(t)$ и $\lambda(t)$ в прогностическом дискретизированном НСО виде. Т.е. мы хотим найти $x(t_{k+1})$ и $\lambda(t_{k+1})$ из более ранних данных для времен t_j с $j = k, k - 1, \dots$ в режиме реального времени и точно. Для $f : R^n \rightarrow R$, векторы столбцов $x(t) \in R^n$, векторы строк $\lambda(t) \in R^m$ и $A(t) \in R_{m,n}$, выбираем изменяющуюся во времени функцию Лагранжа со знаком $+$

$$\mathcal{L}(x(t), \lambda(t)) = f(x(t)) + \lambda(t)(A(t)x(t) - b(t)) : R^{n+m} \rightarrow R.$$

Нужно оценить DE Лагранжа для \mathcal{L} через соответствующие векторы градиента для x и λ . Наглядно

$$\begin{aligned} \nabla_x \mathcal{L}(x(t), \lambda(t)) &= \nabla_x \left(f(x(t)) + \lambda(t)(A(t)x(t) - b(t)) \right) = \\ &= \nabla_x f(x(t)) + \lambda(t)A(t) \in R^n \end{aligned}$$

$$\text{и } \nabla_\lambda \mathcal{L}(x(t), \lambda(t)) = (A(t)x(t) - b(t))^T \in R^m \quad (1)$$

поскольку производная линейной функции, такой как $\lambda(t)(A(t)x(t) - b(t))$ по отношению к x , является вектор строки $\lambda(t)A(t) \in R^n$, а по отношению к λ она

равна вектор столбца $A(t)x(t) - b(t) \in R^n$. Для объединения двух уравнений (1) задаем $y(t) = [x(t); \lambda(t)^T] \in R^{n+m}$ в нотации Matlab и определяем

$$h(y(t)) = \begin{pmatrix} \nabla_x \mathcal{L}(x(t), \lambda(t)) \\ \nabla_\lambda \mathcal{L}(x(t), \lambda(t)) \end{pmatrix}^T = \begin{pmatrix} \nabla_x f(x(t))^T + A(t)^T \lambda(t) \\ A(t)x(t) - b(t) \end{pmatrix} = \begin{pmatrix} h_1(t) \\ \vdots \\ h_{n+m}(t) \end{pmatrix} \quad (\text{шаг 1})$$

функция ошибки. Чтобы найти производную $\dot{y}(t)$ от $y(t)$, мы используем правило многопеременной цепи, которое дает нам производную $h(y(t))$ как

$$\dot{h}(y(t)) = J(h(y(t))) \dot{y}(t) + \dot{h}_t(y(t)).$$

Здесь

$$J(h(y(t))) = \begin{pmatrix} \frac{\delta h_1(t)}{\delta x_1} & \dots & \frac{\delta h_1(t)}{\delta \lambda_m} \\ \vdots & & \vdots \\ \frac{\delta h_{n+m}(t)}{\delta x_1} & \dots & \frac{\delta h_{n+m}(t)}{\delta \lambda_m} \end{pmatrix}_{n+m, n+m} \quad \text{и} \quad \dot{h}_t(y(t)) = \begin{pmatrix} \frac{\delta h_1(y(t))}{\delta t} \\ \vdots \\ \frac{\delta h_{n+m}(y(t))}{\delta t} \end{pmatrix}$$

это якобианская матрица J из $h(y(t))$, взята относительно места вектора $x(t) = (x_1(t), \dots, x_n(t))$ и вектор коэффициентов Лагранжа $(\lambda_1(t), \dots, \lambda_m(t))$ и производная по времени от $h(y(t))$ соответственно.

Для изменяющихся во времени задач наименьших квадратов и с использованием стандартного «Ansatz» Лагранжа теперь определим $\dot{h}(y(t)) = -\eta h(y(t))$ (шаг 2) что экспоненциально быстро приведет нас к оптимальному решению $y(t)$ для $t_0 \leq t \leq t_f$. Решение для $\dot{y}(t)$ дает

$$\dot{y}(t) = -J(h(y(t)))^{-1} (\eta h(y(t)) + \dot{h}_t(y(t))) \quad (\text{шаг 3})$$

Используем формулу конечных разностей 5-IFD (шаг 4) для \dot{y}_k с дискретизированными данными $y_k = y(t_k)$, получим следующее уравнение без производных для итераций y_j с $j \leq k$, приравняв два выражения для \dot{y}_k следующим образом:

$$18\tau \cdot \dot{y}_k = 8y_{k+1} + y_k - 6y_{k-1} - 5y_{k-2} + 2y_{k-3} = -18\tau \cdot J(h(y_k))^{-1} (\eta h(y_k) + \dot{h}_t(y_k)) \quad (\text{шаг 5})$$

Решение 5 для y_{k+1} дает полную формулу рекурсии НСО, которая завершает шаг 6 разработки предиктивного дискретизированного алгоритма НСО для изменяющимися во времени ограниченных нелинейных оптимизаций:

$$y_{k+1} = -\frac{9}{4}\tau \cdot J(h(y_k))^{-1} + (\eta h(y(t) + \dot{h}_t(y_k)) - \frac{1}{8}y_k + \frac{3}{4}y_{k-1} + \frac{5}{8}y_{k-2} - \frac{1}{4}y_{k-3}) \\ \in \mathbb{C}^{n+m}$$

Алгоритм оптимизации на основе Лагранжа для многомерных функций и ограничений. Для этого конкретного примера оптимальное решение известно. Код может быть модифицирован для решения других задач оптимизации. Он является модульным и принимает все сходящиеся с опережением формулы конечных разностей, которые приведены в Polyksrestcoeff3.m. [144] в формате k_s .

Результаты Главы 3

Глава 3 представляет собой этап исследования, посвященный разработке алгоритма управления ресурсами в распределенных динамических вычислительных системах (РДВС) с использованием нейронных сетей обнуления Чжана (НСО). В этой главе проведена разработка алгоритма, рассмотрена его ключевая концепция и разработан метод применения в контексте РДВС.

Мы начали с обзора классических методов оптимизации и нейронных сетей. Далее был выбран метод нейронных сетей обнуления Чжана (НСО) как инструмента для реализации гибридного подхода управления ресурсами в РДВС. Затем мы рассмотрели адаптацию этого метода к РДВС, учитывая специфику структуры и требований системы. В результате данной разработки был создан алгоритм использования НСО для решения задачи управления ресурсами в РДВС.

Глава 3 является важным шагом в исследовании оптимального управления ресурсами в РДВС на базе нейронных сетей обнуления Чжана (НСО). Разработанный алгоритм и полученные результаты обеспечивают основу для дальнейших экспериментальных исследований с целью подтверждения эффективности его работы.

4 Программная реализация и экспериментальные исследования эффективности подхода к управлению ресурсами распределенной динамической вычислительной системы на базе нейронных сетей обнуления Чжана

Для организации вычислительных экспериментов был разработан программный комплекс подхода к управлению ресурсами распределенных динамических систем на базе обнуляющей нейронной сети, зарегистрированный в государственном реестре программ для ЭВМ [145]. Модель была реализована на высокоуровневом языке программирования MATLAB, который широко используется в научных и инженерных приложениях, в том числе в области моделирования и оптимизации.

В данной главе описаны экспериментальные исследования эффективности предложенного подхода управления ресурсами в распределенных динамических вычислительных системах на базе обнуляющих нейронных сетей Чжана.

4.1 Описание программной реализации подхода

В данном разделе представлено описание программной реализации метода управления ресурсами, основанного на нейронных сетях обнуления Чжана (ZNN) в распределенных динамических вычислительных системах (РДВС). Программная реализация является ключевым шагом для применения метода на практике и его дальнейшего экспериментального исследования.

Далее коротко описана программная реализация:

1. Модуль загрузки и предобработки данных:

Функции для загрузки и чтения исходных данных, включая характеристики ресурсов, задачи и параметры системы.

Функции для предобработки данных, такие как нормализация, фильтрация или обработка выбросов.

2. Модуль обучения обнуляющих нейронных сетей:

Функции для определения архитектуры и параметров обнуляющих нейронных сетей.

Функции для разделения данных на обучающую и тестовую выборки.

Функции для обучения нейронных сетей на основе предварительно подготовленных данных и оценки их производительности.

3. Модуль оптимального управления ресурсами:

Алгоритм, учитывающий минимизацию углеродного следа и сохранение производительности системы.

Использование обученных обнуляющих нейронных сетей для принятия решений о распределении ресурсов частоты и миграций задач.

Оптимизация алгоритма с помощью итераций и анализа результатов.

4. Модуль экспериментов и оценки результатов:

Функции для проведения серии экспериментов с использованием реальных и синтетических данных.

Функции для сравнения производительности и углеродного следа с другими существующими методами управления ресурсами.

Функции для анализа результатов, включая оценку эффективности, точности и стабильности разработанного подхода.

5. Модуль визуализации:

Функции для визуализации результатов экспериментов и процесса управления ресурсами.

Описание кода MATLAB, который выполняет временную оптимизацию с использованием метода Лагранжа и опережающей дискретизации. Задача состоит в минимизации целевой функции $f(x, y, t)$ при наличии ограничения вида $Ax = b$. Код включает следующие шаги:

Входные параметры:

t_o и t_f – временные пределы от t_o до t_f .

j и s – параметры, определяющие тип используемой опережающей дискретизации.

τ_{arr} и ϵ_{arr} – массивы интервалов дискретизации и констант затухания.

$graph$ – флаг, указывающий, нужно ли строить графику.

Символьные переменные:

Задаются символьные переменные x , y , z_l , z_t .

Целевая функция:

Целевая функция $f(x, y, t)$ вычисляется на основе символьных переменных.

Символьные вычисления:

Вычисляются градиент, якобиан и другие символьные выражения, связанные с целевой функцией.

Преобразование символьных функций в числовые функции:

Исходные символьные функции преобразуются в числовые функции для последующих вычислений.

Инициализация:

Начальные значения устанавливаются с использованием метода Эйлера.

Основной цикл:

Внутри цикла выполняется опережающая дискретизация для минимизации целевой функции.

Построение графиков (по желанию):

Если установлен флаг `graph`, код генерирует графику для анализа результатов.

Целевая функция:

Определяется вспомогательная функция `funct(x, y, t)`, представляющая целевую функцию для минимизации.

Проверка окончания времени:

Проверяется, достигнуто ли конечное время `tf`.

Этот код решает задачу временной оптимизации с использованием метода Лагранжа и опережающей дискретизации, обновляя решение на каждом временном шаге. Результаты могут быть визуализированы с использованием графиков, если флаг `graph` установлен в 1 [146].

4.2 Проведение экспериментов для проверки эффективности разработанного подхода управления

Этот раздел посвящен описанию экспериментов, проведенных для проверки эффективности разработанного метода управления ресурсами на базе нейронных сетей обнуления Чжана (НСО) в распределенных динамических вычислительных

системах (РДВС). Целью проведения экспериментов является оценка работы метода в реальных условиях и сравнение его результатов с существующими методами управления ресурсами.

Методика проведения экспериментов:

Сбор данных. В ходе экспериментов собирались данные о нагрузке на систему, энергопотреблении и производительности. Эти данные использовались для оценки работы подхода.

Запуск экспериментов. Эксперименты включали:

Сравнительный анализ. Полученные результаты были сравнены с результатами, полученными при использовании существующих методов управления ресурсами в РДВС. Это позволило провести оценку эффективности разработанного метода.

Используемые метрики:

Для оценки эффективности метода и сравнения его с другими методами использовались следующие метрики:

Производительность системы: измеряется в терминах выполненных задач или обработанных запросов в единицу времени.

Энергопотребление: оценивается в потребленных энергетических ресурсах системы при выполнении задач.

Распределение ресурсов: анализируется равномерность распределения ресурсов между вычислительными узлами.

Отклик системы: Время, необходимое для реакции системы на изменения нагрузки или требований.

Ожидаемые результаты:

Проведение экспериментов позволит оценить эффективность разработанного метода оптимального управления ресурсами в реальных условиях РДВС. Ожидается, что метод НСО сможет обеспечить более эффективное и адаптивное управление ресурсами, что будет выражено в улучшении производительности системы и снижении энергопотребления. Также ожидается,

что метод НСО сможет лучше распределять ресурсы между узлами системы, обеспечивая более равномерное и эффективное использование ресурсов.

Полученные результаты экспериментов будут представлены и проанализированы в следующем разделе главы.

Для проведения экспериментальных исследований была также выбрана задача изменения частоты в зависимости от длины (иначе – вычислительной сложности) задачи для оптимизации управления ресурсами.

Для проведения экспериментов был взят набор данных <https://www.nordugrid.org> [147]. Проект – предоставление надежного, масштабируемого, портативного и полнофункционального решения для глобальной вычислительной системы и системы Grid данных – промежуточного программного обеспечения ARC, бесплатного программного обеспечения с открытым исходным кодом, которые как организуют работу вычислительных кластеров, так и делятся данными об этой работе. И также был взят набор сгенерированных данных в среде Virtual Internet Routing Lab, состоящий из параметров системы и соответствующих им значений функции цели, и представляющих собой распределенную вычислительную систему, состоящую из 300 вычислительных узлов и каналов связи между ними. Период работы сети составил 30 дней. Затем была проведена серия экспериментов, которая представлена в таблице 4.

Таблица 4. Энергопотребление, необходимое для решения задачи в зависимости от использованной частоты процессора

Номер эксперимента	Частота (ГГц)	Время, затраченное на выполнение задачи	Затраченный энергоресурс
1	0.6	0.4629	5.12
2	0.7	0.3966	5.3467
3	0.8	0.3498	6.5333
4	0.9	0.3113	6.88
5	1.0	0.2854	7.8

Окончание таблицы 4. Энергопотребление, необходимое для решения задачи в зависимости от использованной частоты процессора

Номер эксперимента	Частота (ГГц)	Время, затраченное на выполнение задачи	Затраченный энергоресурс
6	1.1	0.2591	7.4533
7	1.2	0.2434	8.1733
8	1.3	0.2297	8.4133
9	1.4	0.2175	8.2
10	1.5	0.1998	8.133

Из полученных значений можно сделать вывод, что энергия, а следовательно, и выброс углекислого газа, затрачиваемая на вычисления прямо пропорциональна скорости выполнения.

Изучение работы метода нейронных сетей обнуления для решения задачи управления ресурсами распределенной динамической вычислительной сети.

Рассмотрим серию экспериментов по исследованию подбора структуры и параметров НСО, при которых ошибка будет минимальна.

Был выбран порядок ошибок усечения $k+2$. Длины начальных векторов указаны в таблице. Данные для проведения экспериментов описаны в таблице 5.

Таблица 5. Описание данных для экспериментов

k, s	second largest root magnitude	polyrest	taucoeff
1, 2	0.4472	[-0.6; -0.2; -0.2]	1.6
2, 2	0.8360	[3.808215205198584e-01; -1.168617033577950; -4.156914832110251e-01; 2.034869962691167e-01]	2.389539011192650
2,3	0.5226	[1.176689655734125e-01; -8.516604383514612e-01; -3.818277128943235e-01; 6.773366868988580e-02; 4.808551698248645e-02]	2.219772790140505
2,4	0.4637	[-7.325911840449030e-02; -6.340690083336632e-01; -3.531153119650761e-01; 4.999504148385766e-03; 4.272929389028893e-02; 1.271464066455492e-02]	2.090810740934728
3,3	0.8290	[7.356287764690319e-01; -1.310212542580311; -9.130279068639379e-01; 4.241789042054587e-01;	2.706690598404911

Продолжение таблицы 5. Описание данных для экспериментов

k, s	second largest root magnitude	polyrest	taucoeff
		1.601227985618904e-01; -9.669002979213243e-02]	
3,4	0.6397	[5.020490554114563e-01; -1.054074297181794; -7.768688386704860e-01; 1.835959023390256e-01; 2.017816085472454e-01; -2.855311728115157e-02; -2.793031316429520e-02]	2.560245298708237
3,5	0.5816	[2.480059771240410e-01; -7.885198366863375e-01; -6.390357837629567e-01; 4.813679468512866e-03; 1.687449854130043e-01; 4.118123638757526e-02; -2.278552457478595e-02; -1.240473336905329e-02]	2.394810523248907
4,4	0.8477	[1.038159703195576; -1.378825739811979; -1.496966727252006; 6.334710791603586e-01; 5.406390647842177e-01; -3.038823278816560e-01; -9.303412726167593e-02; 6.043907506716473e-02]	2.964332575206227
4,5	0.7508	[6.643887844445272e-01; -9.273967222211478e-01; -1.299068589620714; 1.843924712566855e-01; 5.752670764736695e-01; -4.112171440073703e-02; -2.032607397397568e-01; 2.599124381423302e-02; 2.080818999324064e-02]	2.748056965594512
4,6	0.7513	[1.854040656660873e-01; -4.167064710266586e-01; -1.041699445159904; -4.706410159092017e-02; 2.125066146101461e-01; 2.774435481718890e-01; -1.138997269046412e-01; -5.698432164801684e-02; -2.838362875302206e-02; 2.938346663504090e-02]	2.448959910091972
5,5	0.9241	[1.215386987348222; -1.291056554393765; -2.059796135880188; 6.095320901394168e-01; 1.228581408859198; -4.731170818697435e-01; -5.085129829526192e-01; 2.285625671433630e-01; 1.036480848334800e-01; -5.322838322736356e-02]	3.134323027738423
5,6	0.8450	[9.789318117566991e-01; -1.091516557617663; -1.693989102837719; 1.873678566963183e-01; 1.016348476940637e+00; -2.991032426368895e-02; -4.821578816042506e-01; -2.455350179352843e-02; 1.549728822935899e-01; 7.407813151091853e-03; -2.290147272148661e-02]	2.988932059446562
5,7	0.8529	[9.163297735429142e-01; -1.024354167068672; -1.654636187872638; 1.566276337562119e-01; 9.435980453654198e-01; 2.414314319510011e-02; -4.440174739171195e-01; -4.428233248795205e-02; 1.246731646238272e-01; 2.121989726029419e-02; -1.543586247760823e-	2.951233120586878

Окончание таблицы 5. Описание данных для экспериментов

k, s	second largest root magnitude	polyrest	taucoeff
		02; -3.865633919776790e-03]	

Вычислительные эксперименты были проведены на наборе данных работы сети NorduGRID состоящей из более чем 2,7 миллиарда узлов, распределенных в 141 географических точках. Данные были обогащены оценкой выбросов углекислого газа в результате выработки энергии необходимой для работы этих вычислительных узлов. Эти показатели были получены из интегрированной базы данных по выбросам и генерирующим ресурсам (eGRID) представляющей из себя комплексный источник данных Отдела рынков чистого воздуха Агентства по охране окружающей среды и экологических характеристиках практически всей электроэнергии, вырабатываемой в мире.

Полным перебором была построена функция оптимального управления этой системой. При сохранении достаточной производительности (среднее время расчётов не меньше реального) параметры работы узлов, расположение задач на узлах были перераспределены так, чтобы минимизировать выбросы CO². На временном промежутке в 30 дней это позволило снизить выбросы на 9,4%. Необходимо отметить, что расчет оптимального управления системой на компьютере с характеристиками: процессоры core i9 11980HK(2,6-3,3ГГц), GeForce RTX 3090 Ti, оперативная память 128 Гб и 24Гб видеопамяти, 1Тб твердотельный накопитель, составил 47 часов и 12 минут.

Так как функция управления РДВС зависит от параметров настройки вычислительных узлов (x) и миграции (y), будем рассматривать поведение НСО по ним.

На рисунке 11 представлены результаты работы при следующих параметрах: $k = 1, s = 2, \tau = 0,03, \eta = 25$.

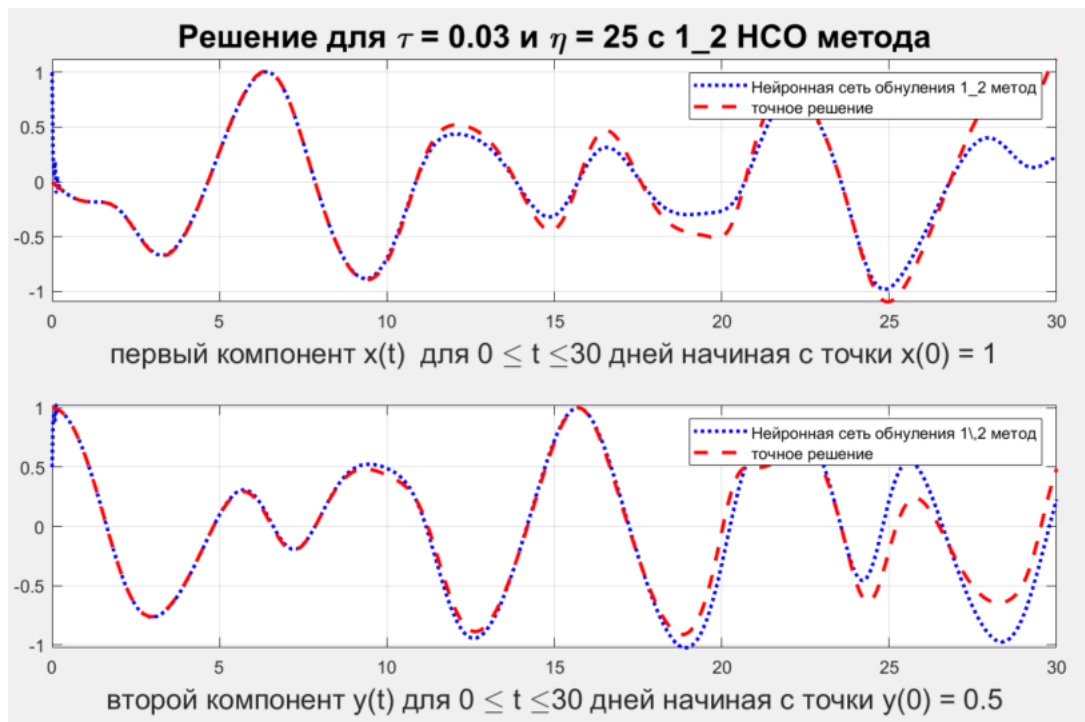


Рисунок 11 – Оценка качества работы НСО. Эксперимент1.

Ошибка работы метода представлена на рисунке 12.

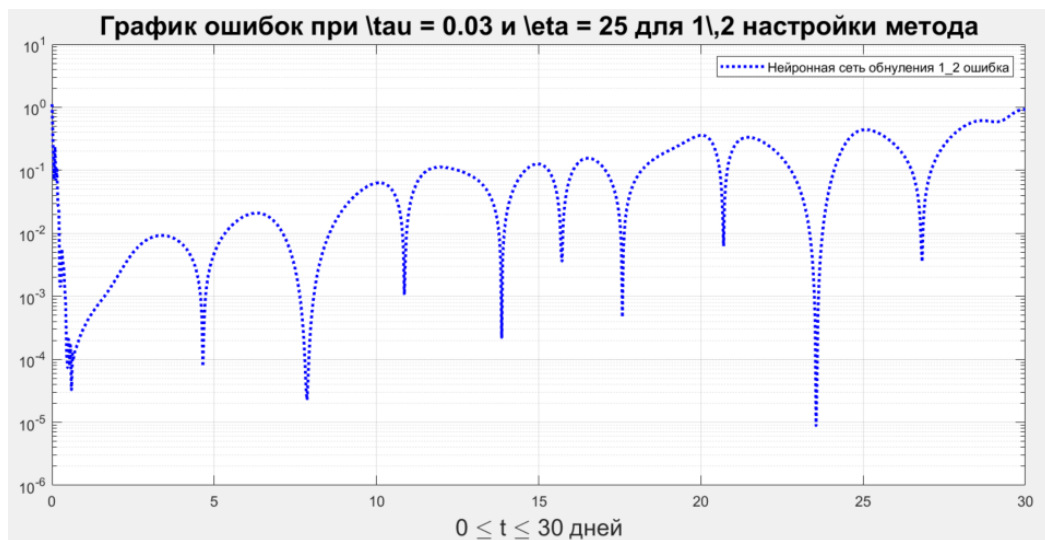


Рисунок 12 – Ошибка работы метода НСО. Эксперимент1.

Видно, что к концу периода ошибка становится критической и не допустимой для управления. При сокращении интервалов наблюдения и увеличении константы

затухания результаты становятся значительно хуже, и ошибка возрастает до 1000 (Рисунок 13).

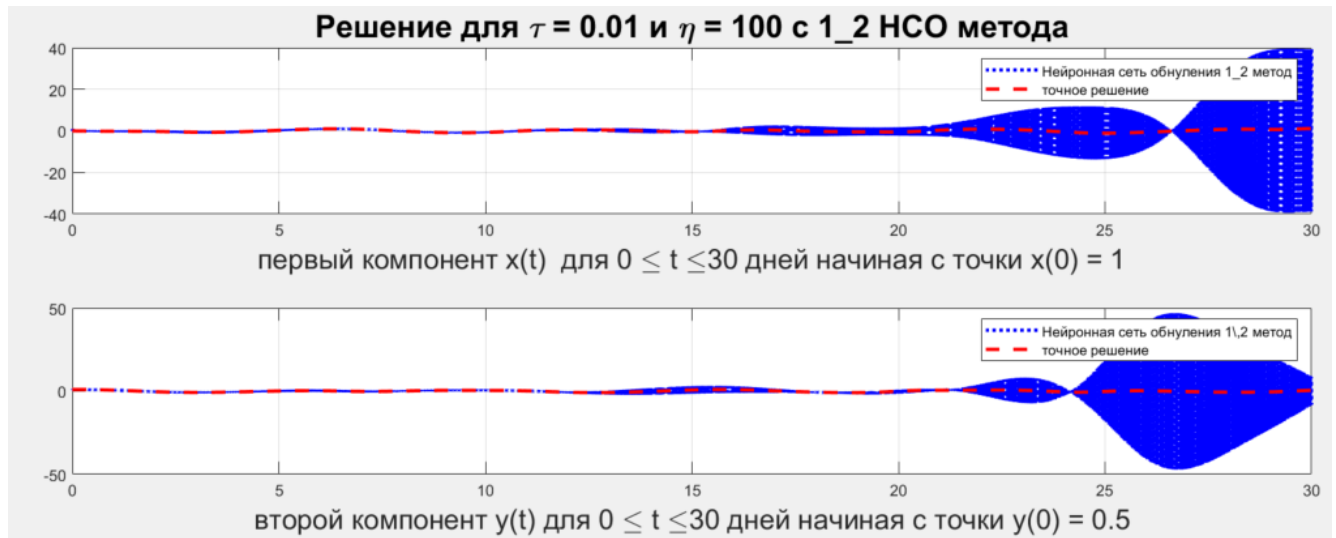


Рисунок 13 – Оценка качества работы НСО. Эксперимент 2.

Изменение параметров начального вектора и существенное увеличение частоты выборки до $\tau = 0,0002$, что в свою очередь приводит к увеличению вычислительной сложности работы, также не приводит к существенному снижению ошибки в интервале от 24 суток и далее.

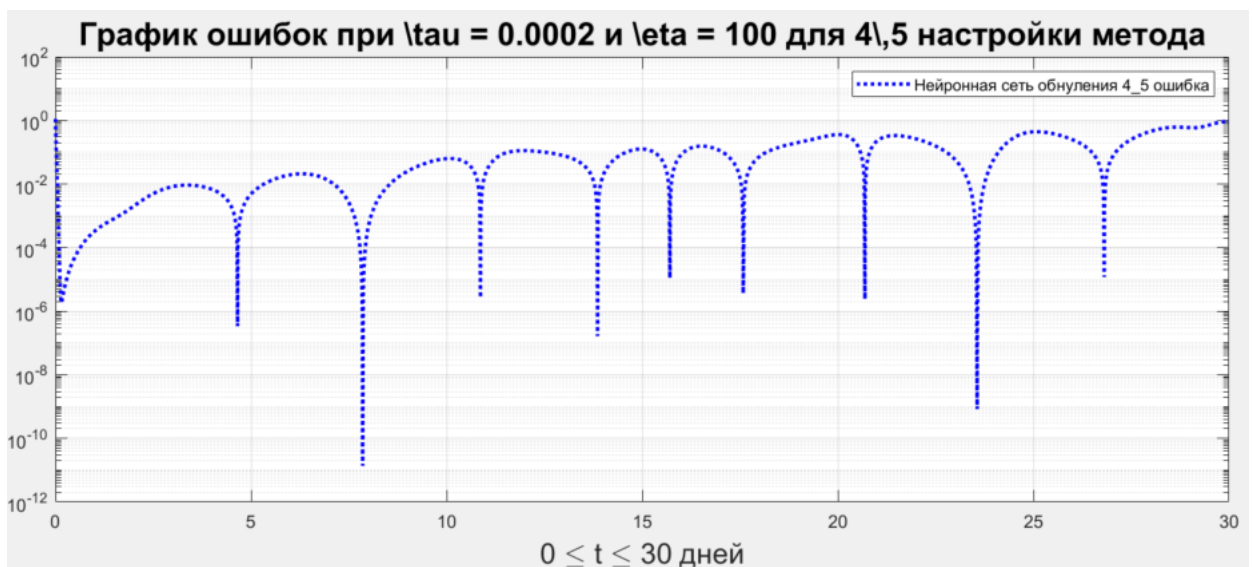


Рисунок 14 – Ошибка работы метода НСО. Эксперимент 3.

Подобная данные были получены и при зашумлении данных работы системы и изменении оптимальных показаний функции.

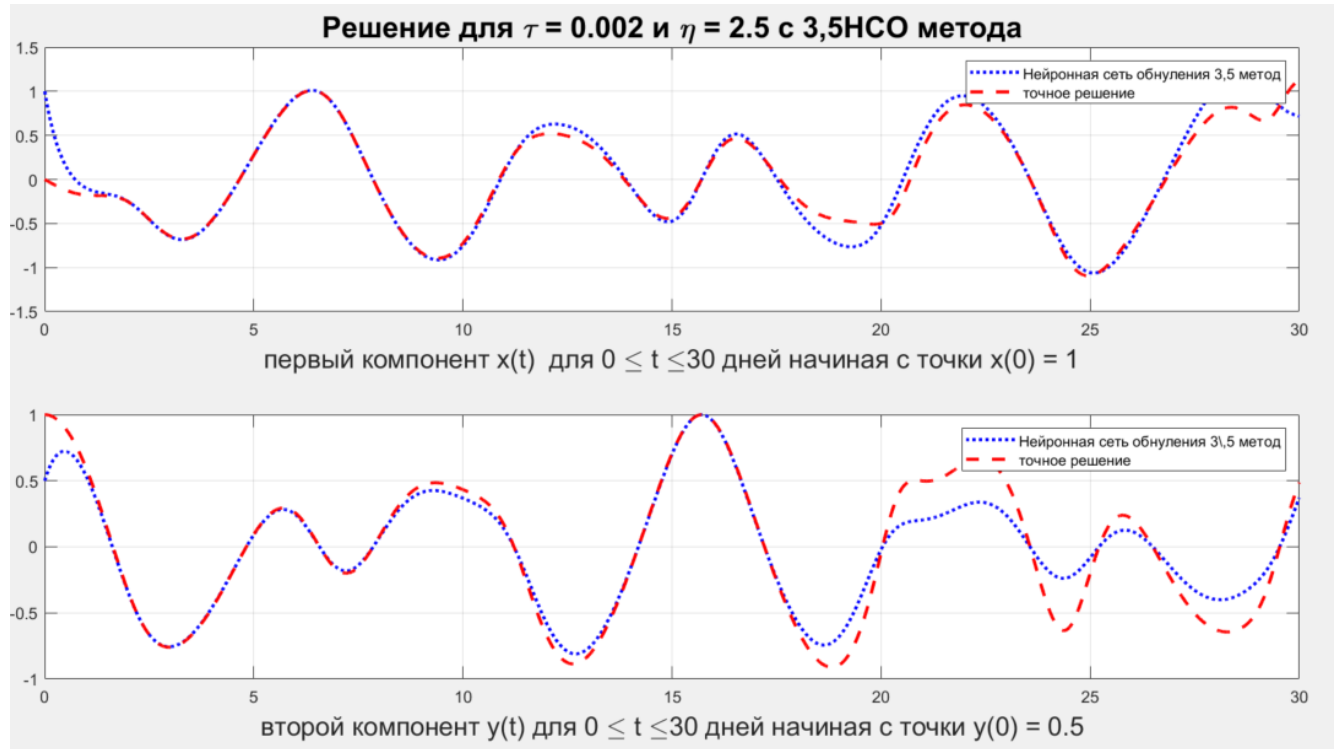


Рисунок 15 – Оценка качества работы НСО. Эксперимент 4.

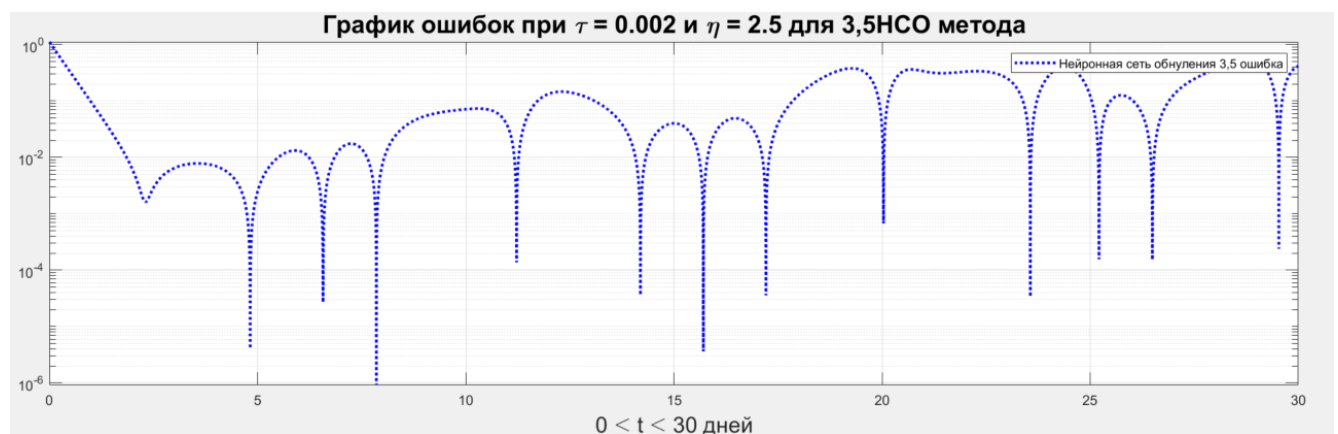


Рисунок 16 – Ошибка работы метода НСО. Эксперимент 4.

Лучшим результатом можно считать параметры: $k = 2, s = 4, \tau = 0,0001, \eta = 100$, при которых уже в первые сутки ошибка достигает показателей

10^{-4} и в течении всего периода не увеличивается выше показателя 10^{-10} . Визуализации этого эксперимента приведены на 17 рисунке.

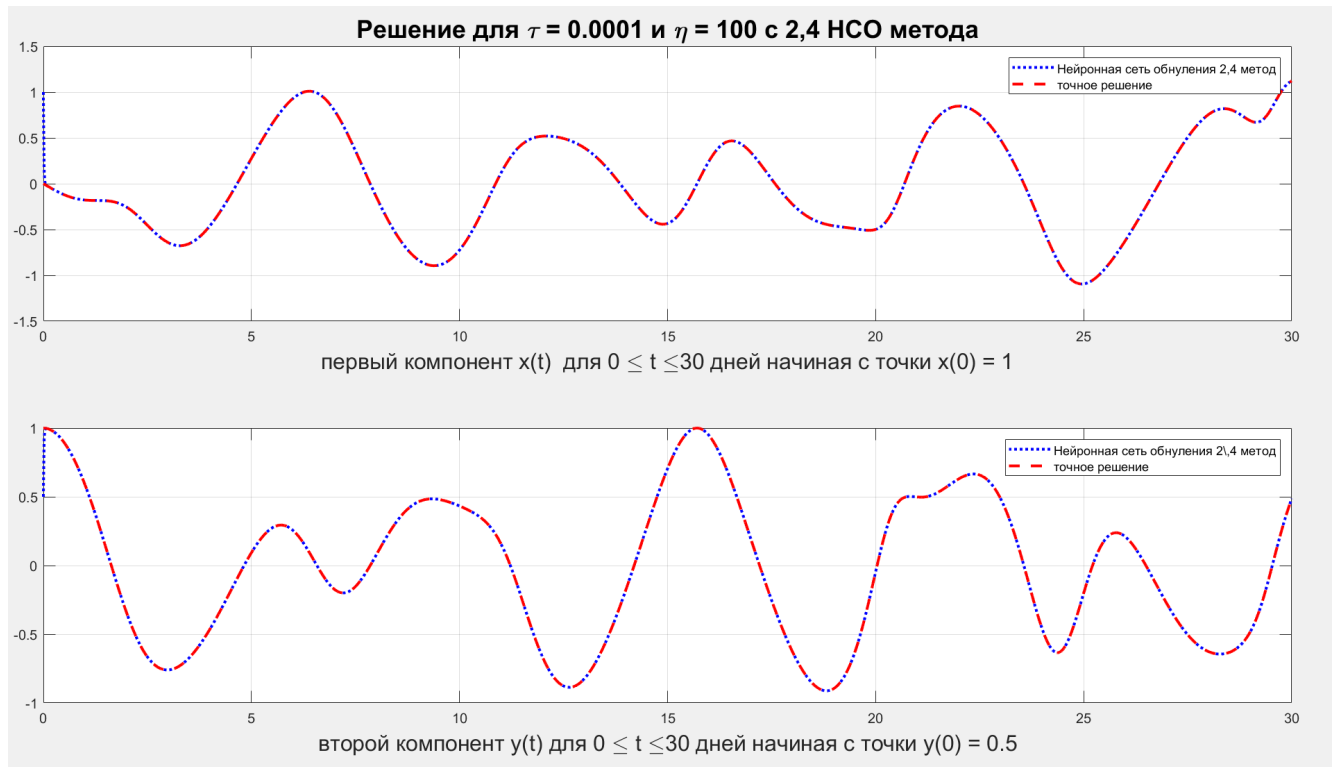


Рисунок 17 – Оценка качества работы НСО. Эксперимент 5.

График ошибки можно видеть на 18 рисунке.

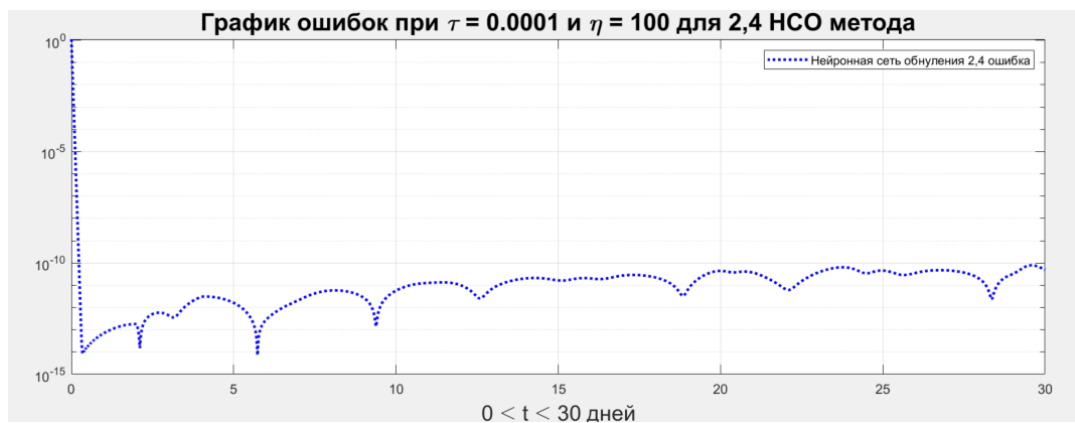


Рисунок 18 – Ошибка работы метода НСО. Эксперимент 5.

На 19 рисунке визуализировано поведение алгоритма по двум компонентам вместе.

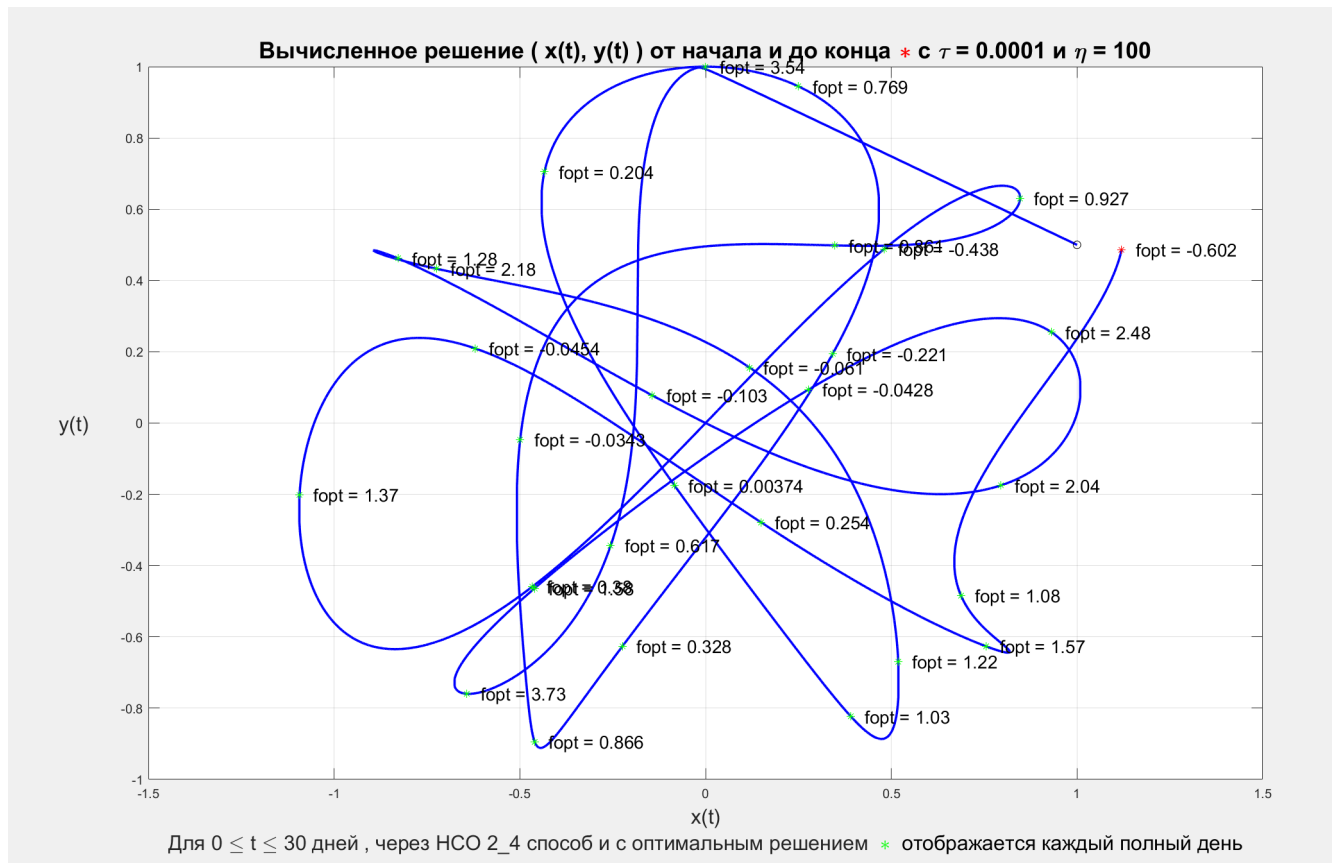


Рисунок 19 – Оценка качества работы НСО по двум компонентам. Эксперимент 5.

В зависимости от энергопотребления мы построили график задержки задачи, который представлен на рисунке 20. Мы видим, что для всех входных последовательностей задержка задачи соответствовала указанному требованию. Когда длина входного сигнала была небольшой (т. е. 200 и 300), задержка такого запроса была очень маленькой, и его можно было легко удовлетворить с использованием самой низкой частоты. Однако по мере увеличения вычислительной сложности задачи использование самой низкой частоты значительно увеличивало время выполнения. Рабочая частота варьировалась соответствующим образом, чтобы соответствовать ограничениям, накладываемым на работу системы.



Рисунок 20 – Задержка задачи с использованием предложенного подхода на пограничных устройствах

4.3 Сравнение результатов с существующими методами и анализ полученных данных

Этот раздел посвящен сравнительному анализу результатов, полученных в результате проведения экспериментов с использованием разработанного подхода управления ресурсами на базе нейронных сетей обнуления Чжана (НСО), с результатами, полученными при применении существующих методов управления ресурсами в распределенных динамических вычислительных системах (РДВС).

Сравнение с существующими методами:

В рамках экспериментов были проведены сравнительные анализы между разработанным методом НСО и следующими существующими методами управления ресурсами:

Далее в таблице 6 и на рисунке 21 представлено сравнение метода НСО к классическим RNN и OSR [148-149].

Таблица 6. Сравнение HCO с RNN и OSR

	HCO	RNN	OSR
1	0,007	0,0072	0,0071
2	0,0054	0,00531	0,00529
3	0,0034	0,0041	0,0043
4	0,002	0,002	0,003
5	0,003	0,003	0,003
6	0,001	0,002	0,002
7	0,001	0,002	0,001
8	0,002	0,002	0,002
9	0,001	0,001	0,001
10	0,002	0,002	0,002
11	0,001	0,002	0,003
12	0,003	0,002	0,002
13	0,002	0,002	0,001
14	0,001	0,001	0,001
15	0,001	0,001	0,001
16	0,002	0,002	0,002
17	0,001	0,002	0,003

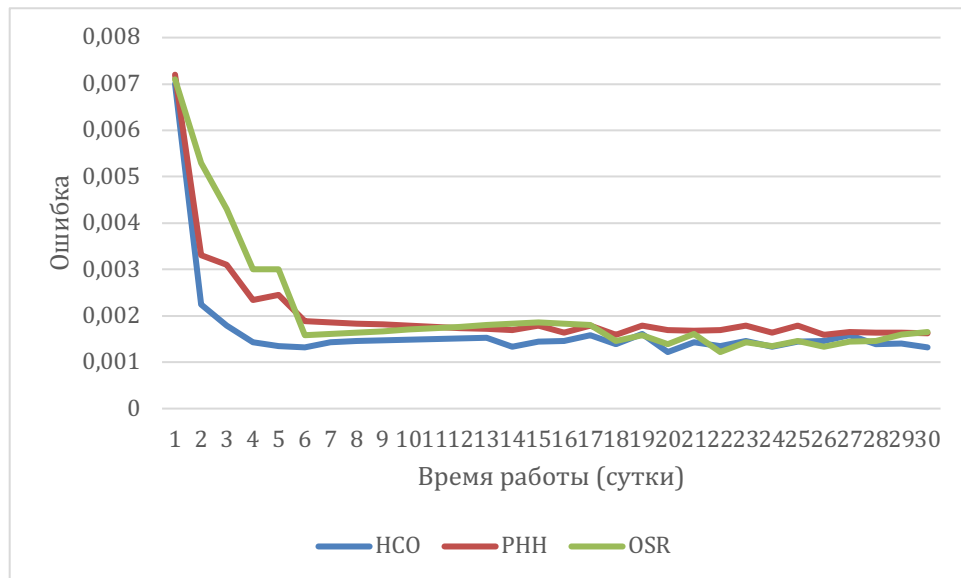


Рисунок 21 – Сравнение решения задач с RNN и OSR

Анализ полученных результатов показал, что метод обнуляющих нейронных сетей демонстрирует хорошую эффективность в управлении ресурсами в распределенных динамических вычислительных системах. Он позволяет достигать значительного уменьшения значения функции цели при соблюдении всех ограничений на значения параметров системы.

Таким образом можно сделать выводы, что разработанный метод советуется таким критериям:

Увеличение производительности. Разработанный метод способен значительно повысить производительность с учетом влияния на окружающую среду системы по сравнению с другими методами.

Снижение энергопотребления. Гибридный метод также обеспечивает снижение энергопотребления системы, что делает его эффективным для энергоэффективных РДВС.

Адаптивность. В отличие от статических методов позволяет адаптироваться к изменяющейся нагрузке, что делает его более гибким и эффективным.

Равномерное распределение ресурсов. Разработанный метод способен более равномерно распределять ресурсы между узлами системы, что приводит к более эффективному использованию ресурсов.

Полученные результаты позволяют сделать вывод о том, что разработанный метод управления ресурсами на базе нейронных сетей обнуления Чжана обладает высокой эффективностью и адаптивностью к изменяющимся условиям в РДВС. Он способен повысить производительность, снизить энергопотребление и обеспечить равномерное распределение ресурсов. Эти результаты подтверждают перспективность и потенциал применения подхода в управлении ресурсами в распределенных динамических вычислительных системах.

Результаты Главы 4

Глава 4 представила результаты программной реализации метода управления ресурсами, основанного на нейронных сетях обнуления Чжана (НСО), а также результаты экспериментальных исследований, направленных на оценку эффективности данного метода в контексте распределенных динамических вычислительных систем (РДВС). В результате проведенных исследований и анализа данных были получены следующие ключевые выводы:

Разработанный гибридный подход оказался успешно реализован в программном виде и демонстрирует высокую производительность при управлении ресурсами в РДВС.

Эксперименты, проведенные с использованием гибридного подхода, показали его способность эффективно адаптироваться к различным сценариям и нагрузкам, что делает его перспективным инструментом для оптимального управления ресурсами в динамичных системах.

Сравнение результатов, полученных с помощью гибридного подхода, с результатами, полученными с применением существующих методов управления ресурсами, выявило преимущества разработанного метода в плане производительности и эффективности использования ресурсов.

Результаты также подтвердили способность гибридного подхода к снижению энергопотребления системы, что актуально в контексте сокращения углеродного следа РДВС.

Эти выводы подчеркивают важность и перспективность гибридный подход управления ресурсами на базе нейронных сетей обнуления Чжана в распределенных динамических вычислительных системах. Разработанный подход представляет собой инновационный подход к управлению ресурсами и управлению производительностью, что может иметь широкое практическое применение в современных вычислительных системах. Полученные результаты позволяют сделать выводы о преимуществах предложенного метода.

ЗАКЛЮЧЕНИЕ

В главе 1 была рассмотрена актуальность проблемы управления ресурсами в распределенных динамических системах, а также были представлены основные существующие методы решения этой проблемы.

В главе 2 были проведены математическое и имитационное моделирования управления ресурсами в распределенных динамических вычислительных системах и поставлена задача оптимизации. Также выделены значимые ресурсы для уменьшения размерности задачи.

В главе 3 был предложен гибридный метод управления ресурсами на базе НСО. Была представлена архитектура НСО, а также алгоритм ее обучения. Был представлен алгоритм управления ресурсами распределенной динамической вычислительной системы, в которой была реализована НСО.

В главе 4 были проведены эксперименты для оценки эффективности метода. Были сравнены результаты работы НСО с результатами работы других методов управления ресурсами. Были проанализированы полученные результаты и сделаны выводы.

Полученный метод, включая разработанные алгоритмы лег в основу программного комплекса, зарегистрированного в Федеральной службе по интеллектуальной собственности (Приложение А). Результаты диссертационного исследования прошли всестороннюю апробацию и внедрены в практику Экологического научно-образовательного центра мирового уровня «Енисейская Сибирь» (Приложение Б).

Таким образом разработана обобщенная математическая модель распределенной динамической вычислительной системы, которая позволяет количественно оценить углекислого газа выбрасываемое в атмосферу при решении любой задачи на любом из узлов. Предложена имитационная модель распределенной динамической вычислительной системы, которая позволяет изучать поведение системы во времени при различной структуре системы. Создан гибридный подход адаптивного управления ресурсами в распределенных динамических вычислительных системах включающий алгоритм планирования

задач миграции, репликации, задержки и вычислений, технологию динамического управления частотой и напряжением работы процессоров, а также обнуляющий нейросетевой алгоритм управления ресурсами РДВС позволяет, за счет комплексности и точности решения задачи, снизить выбросы углекислого газа в атмосферу до 7% в системе состоящей из 2700209 узлов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Vadym Kolumbet, Olga Svynchuk. Multiagent methods of management of distributed in hybrid clusters. *Advanced Information Systems* 6(1):32-36. 2022. DOI:10.20998/2522-9052.2022.1.05
2. А. С. Андрэ и Т. Эдлер, «О глобальном использовании телекоммуникационных технологий: тенденции до 2030 года», *Вызовы*, том 6, No 1, стр. 117–157, 2015.
3. Н. Джонс, «Как остановить центры обработки данных от поглощения мирового электричества», журнал *Nature*, том 561, No 7722, стр. 163–167, 2018 г.
4. Ф. Янг и А. А. Чиен, «Крупномасштабные и экстремальные вычисления с застрявшей зеленой энергией: возможности и затраты», *IEEE Transactions on Parallel and Distributed Systems*, том 29, No 5, стр. 1103–1116, 2018.
5. Apple, «Отчет о воздействии продукта на окружающую среду iphone 11», 2019 г.
6. Ф. Янг и А. А. Чиен, «Крупномасштабные и экстремальные вычисления с застрявшей зеленой энергией: возможности и затраты», *IEEE Transactions on Parallel and Distributed Systems*, том 29, No 5, стр. 1103–1116, 2018.
7. Перспективы реализации лесоклиматических проектов: потенциал регионов Енисейской Сибири : флагманский аналитический доклад / Е. А. Ваганов, А. И. Пыжев, М. В. Курбатова [и др.] ; науч. ред. Е. А. Ваганов, А. И. Пыжев, М. В. Курбатова ; Сиб. фед. ун-т. — Красноярск : Сиб. федер. ун-т, 2023. — 160 с.
8. Yaru Fu, Yue Shan. A Distributed Microservice-aware Paradigm for 6G: Challenges, Principles, and Research Opportunities. *IEEE Network* PP(99):1-1. 2023. DOI:10.1109/MNET.2023.3321528.
9. Romain Demangeon, Nobuko Yoshida. Causal Computational Complexity of Distributed Processes. *Information and Computation* 290:104998. 2022. DOI:10.1016/j.ic.2022.104998.

10. Cohen P.R., Morgan J.L., Pollack M. Intentions in Communication Bradford Books. MITPress. – 1990. – 508 p.
11. Антамошкин О.А. Технология утилизации ресурсов распределенных вычислительных // Системы управления и информационные технологии. – 2012. – Т. 48. – № 2.2. – С. 220–224.
12. Антамошкин О.А. Система поддержки принятия решений на основе многоатрибутивных методов // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2009. – № 4 (25). – С. 69–71.
13. Antamoshkin O.A., Antamoshkina O.A. Multi-agent automation system for monitoring, forecasting and managing emergency situations // IOP Conf. Series: Materials Science and Engineering. – 2016. – Vol. 120. Article ID 012016/ – 9P. DOI:10.1088/1757-899X/94/1/012016.
14. Швецов А.Н. Агентно-ориентированные системы: основные модели: монография. – Вологда: ВоГТУ, 2012. – 190 с.
15. Гаврилова Т.А., Червинская К.Р. Извлечение и структурирование знаний для экспертных систем. – М.: Радио и связь, 1992. – 200 с.
16. Guide to PAS 2050: How to assess the carbon footprint of goods and services. ISBN 978-0-580-71382-8. 2008.
17. Аббасов, М. Э. Методы оптимизации: учебное пособие / М. Э. Аббасов. – Санкт-Петербург : Издательство «ВВМ», 2014. – 64 с.
18. Некрасов С.И., Некрасова Н.А. Философия науки и техники: тематический словарь справочник. Учебное пособие. – Орёл: ОГУ, 2010. – 289 с.
19. Воробьёв, В. А. Теория систем и системный анализ. Стохастические системы: учебное пособие / В. А. Воробьёв, Ю. В. Березовская; Сев. (Арктич.) федер. ун-тим. М. В. Ломоносова. – Архангельск: ИПЦ САФУ, 2012. – 147 с. – ISBN 978-5-261-00616-9. – Текст: непосредственный.
20. ГОСТ 15971-90. Системы обработки информации. Термины и определения. Взамен ГОСТ 15971-84. Введ. с: 01.01.1992 – М.: Изд-во стандартов, 1991. 12 с.

21. Медведев, А. В. Оптимизационная система поддержки принятия решений в бизнес-планировании / А. В. Медведев // Успехи современного естествознания. – 2015. – No 1-4. – С. 679-683.
22. Бриллюэн, Л. Наука и теория информации / Л. Бриллюэн. – Москва: Наука, 1960. – 392 с. – Текст: непосредственный.
23. Ростовцев, В. С. Искусственные нейронные сети : учебник для вузов / В. С. Ростовцев. – 2-е изд., стер. – Санкт-Петербург: Лань, 2021. – 216 с. – ISBN 978-5-8114-7462-2. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/160142> (дата обращения: 22.10.2023). – Режим доступа: для авториз. пользователей.
24. Frank Uhlig. Zeroing Neural Networks, an Introduction to, a Survey of, and Predictive Computations for Time-varying Matrix Problems. 2020
25. Борисова, Л. Ф. Моделирование оптимальных сетевых структур на базе графов кодовых пересечений при объединении произвольных сетей / Л. Ф. Борисова, Н. М. Путинцев. – Текст: непосредственный// Вестник МГТУ. – 2011. – Том 14. – No 4. – С. 789–798.
26. Основы математического моделирования: учебное пособие / С. В. Звонарев. — Екатеринбург: Изд-во Урал. ун-та, 2019. — 112 с.
27. Моделирование технологических процессов: конспект лекций / В. А. Штерензон. Екатеринбург: Изд-во Рос. гос. проф.-пед. ун-та, 2010. 66 с.
28. Распределенные системы; Автор: Таненбаум Э., Стин М.; Перевод: Яроцкий В.; 2020 года; Объем, стр: 584; ISBN: 978-5-97060-708-4
29. А. Каток, Б. Хасселблат. Введение в современную теорию динамических систем. М.: Факториал, 1999.
30. Нестеров С. А. Адаптивные системы управления: Конспект лекций. – СПб.: Факультет технической кибернетики СПбГПУ, 2005. – 90 с.
31. Северцев Н. А. Динамические системы: безопасность и отказоустойчивость: учебное пособие для вузов / Н. А. Северцев. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2023. – 415 с. – (Высшее

образование). – ISBN 978-5-534-05711-9. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/515633>

32. Отказоустойчивые и масштабируемые вычислительные системы: учеб.-методич. пособие / А.Н. Земцов; ВолгГТУ. – Волгоград, 2021. – 32 с.

33. Системный анализ и информационные технологии в организациях [Текст] : учебное пособие / И. М. Бескровный. – М.: РУДН, 2012. – 392 с.

34. ГОСТ Р 59926-2021 Информационные технологии (ИТ). Эталонная архитектура больших данных. Часть 2. Варианты использования и производные требования.

35. Математическое моделирование: учебное пособие / С.В. Каштаева; Министерство сельского хозяйства Российской Федерации, федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский аграрно-технологический университет имени академика Д.Н. Прянишникова». – Пермь: ИПЦ «Прокрость», 2020. – 112 с; 21 см – Библиогр.: – с. 111-112. – 50 экз. – ISBN 978-5-94279-487-3 – Текст: непосредственный.

36. Симушкин С.В. «Методы теории вероятностей» Издательство: Лань ISBN: 978-5-8114-3442-8 Автор: Симушкин С.В. Год издания: 2020.

37. Будылина, Е. А. Основные принципы проектирования сложных технических систем в приложениях / Е. А. Будылина. – Текст: непосредственный // Молодой ученый. – 2013. – No 5. – С. 42–45.

38. И. А. Чубукова Data Mining. Учебное пособие. – М.: Интернет-Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2006. – 382 с.: ил., табл. – (Серия «Основы информационных технологий»).

39. Н. Паклин. «Кластеризация категориальных данных: масштабируемый алгоритм CLOPE». Электронное издание. URL: <http://www.basegroup.ru/clusterization/clope.htm>

40. Sudipto Guha, Rajeev Rastogi, Kyuseok Shim «CURE: An Efficient Clustering Algorithm for Large Databases». Электронное издание.

41. Tian Zhang, Raghu Ramakrishnan, Miron Livny «BIRCH: An Efficient Data Clustering Method for Very Large Databases». Электронное издание.

42. Технические средства автоматизации. Учебное пособие / ФГБОУ ВО РХТУ им. Д.И. Менделеева, Новомосковский институт (филиал). Новомосковск, 2018. – 102 с. ISBN 978-5-7237-1487-8.
43. Коган Д.И. Динамическое программирование и дискретная многокритериальная оптимизация: Учеб. пособие. Нижний Новгород: Изд-во ННГУ, 2004. XXX с.
44. Вьюгин В.В. Математические основы машинного обучения и прогнозирования. Москва, издательство МЦНМО, 2018. 384 с.
45. БУРКОВ В.Н. и др. Механизмы управления. – М: Ленанд, 2011. – 192 с.
46. В. С. Симанков, Е. В. Луценко, В. Н. Лаптев, Системный анализ в адаптивном управлении. – 2001. ISBN 5-8344-0007-8.
47. Андросик, А. Б. Неопределенность информации в задаче прогнозирования / А. Б. Андросик, В. И. Хрусталева, А. С. Дулесов [и др.]. – Текст: непосредственный // Информационные технологии: приоритетные направления развития: [монография]. – Книга 7. – Новосибирск: СИБПРИНТ, 2012. – С. 97–112.
48. Л. С. Понтрягин, К теории дифференциальных игр, УМН, 1966, том 21, выпуск 4, 219–274.
49. Lakra A.V., Yadav D.K. Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization. Procedia Computer Science. 2015;48:107–113. DOI:10.1016/j.procs.2015.04.158.
50. Лазарева Т. Я., Мартемьянов Ю. Ф. Основы теории автоматического управления: Учебное пособие. 2-е изд., перераб. и доп. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2004. 352 с. ISBN 5-8265-0149-9.
51. Авдеева, З., Коврига, С., Макаренко, Д., Максимов, В. (2007). Когнитивный Подход В Управлении. URL: <http://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edsbas&AN=edsbas.D98B485A>

52. Mondal R., Nandi E., Sarddar D. Load Balancing Scheduling with Shortest Load First. *International Journal of Grid and Distributed Computing*. 2015;8: 171–178. DOI:10.14257/ijgdc.2015.8.4.17.
53. Lakra A.V., Yadav D.K. Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization. *Procedia Computer Science*. 2015;48:107–113. DOI: 10.1016/j.procs.2015.04.158.
54. Wang H., Wang, F., Liu, J., Wang, D., Groen, J. Enabling customer-provided resources for cloud computing: Potentials, challenges, and implementation. *IEEE Transactions on Parallel and Distributed Systems*. 2015;26:1874–1886.
55. Gill S.S., Chana I., Singh M., Buyya R. CHOPPER: An intelligent QoS-aware autonomic resource management approach for cloud computing. *Cluster Computing*. 2018;21:1203–1241. DOI:10.1007/s10586-017-1040-z.
56. Thomas A., Krishnalal G., Raj P.V. Credit Based Scheduling Algorithm in Cloud Computing Environment. *Procedia Computer Science*. 2015;46:913–920. DOI:10.1016/j.procs.2015.02.162.
57. Sajid M., Raza, Z. Turnaround Time Minimization-Based Static Scheduling Model Using Task Duplication for Fine-Grained Parallel Applications onto Hybrid Cloud Environment. *IETE Journal of Research*. 2015;62(3):1–13. DOI:10.1080/03772063.2015.1075911.
58. Hadji M., Zeghlache D. Minimum Cost Maximum Flow Algorithm for Dynamic Resource Allocation in Clouds. *Cloud Computing (CLOUD)*, 2012 IEEE 5th International Conference on. 2012;876–882. DOI:10.1109/CLOUD.2012.36.
59. Elzeki O., Reshad M., Abu Elsoud, M. Improved Max-Min Algorithm in Cloud Computing. *International Journal of Computer Applications*. 2012; 50(12):22-27. DOI:10.5120/7823-1009.
60. Fernández Cerero, D., Fernández-Montes, A., Jakóbič, A., Kołodziej, J., Toro, M. SCORE: Simulator for cloud optimization of resources and energy consumption. *Simulation Modelling Practice and Theory*. 2018;82:160-173. DOI:10.1016/j.simpat.2018.01.004.

61. Ma T., Chu Y., Zhao L., Otgonbayar A. Resource Allocation and Scheduling in Cloud Computing: Policy and Algorithm. IETE Technical Review. 2014;31(1):4-16. DOI:10.1080/02564602.2014.890837.
62. Carrasco, R.; Iyengar, G.; Stein, C. Resource Cost Aware Scheduling. European Journal of Operational Research. 2018;269(2):621-632. DOI:10.1016/j.ejor.2018.02.059.
63. Wang H., Wang, F., Liu, J., Wang, D., Groen, J. Enabling customer-provided resources for cloud computing: Potentials, challenges, and implementation. IEEE Transactions on Parallel and Distributed Systems. 2015;26:1874–1886.
64. В. Н. Козлов Системный анализ, оптимизация и принятие решений. Численные алгоритмы с С, с CD-ROM, Springer, 1996, 596 p. (MR 97i:65001) (Zbl 857.65003).
65. Т. К. Руткаускас [и др.]; под общ. ред. д-ра экон. наук, проф. Т. К. Руткаускас. – 2-е изд., перераб. и доп. – Екатеринбург: Изд-во УМЦ УПИ, 2018. – 260 с. ISBN 978-5-8295-0563-9.
66. Вильчинская, О. О. Определение количества информации в структуре технической системы / О.О. Вильчинская, И. Н. Гатауллин, С. О. Головинов [и др.]. – Текст: непосредственный // Информационные технологии: приоритетные направления развития. Кн. 5 : [монография]. – Новосибирск: Сибпринт, 2010. – 261 с.
67. Карандеев, Д. Ю. Проблематика осуществления имитационного моделирования распределительной сети / Д. Ю. Карандеев. – Текст: непосредственный // Прикладная математика и информатика : современные исследования в области естественных и технических наук : сборник научных статей IV научно-практической международной конференции (школы-семинара) молодых ученых : 23–25 апреля 2018 г. В двух частях. – Тольятти, 2018. – Ч. 1. – С. 352–358.
68. Karen Shanton, Alvin Goldman, Simulation theory, 2010, <https://doi.org/10.1002/wcs.33>

69. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. – 352 с.
70. Тарасов В.Б. Управление знаниями в сетях предприятий // Новости искусственного интеллекта. – 2003. – № 3. – С. 32–35.
71. Matlab Simulink <https://www.mathworks.com/products/simulink.html>
72. Борисова, Л. Ф. Моделирование оптимальных сетевых структур на базе графов кодовых пересечений при объединении произвольных сетей / Л. Ф. Борисова, Н. М. Путинцев. – Текст : непосредственный // Вестник МГТУ. – 2011. – Том 14. – № 4. – С. 789–798.
73. Глобальный энергетический обзор 2021 // Международное энергетическое агентство (IEA) [Электронный ресурс]. Режим доступа: <https://www.iea.org/reports/global-energy-review-2021> (In Eng.).
74. Barika, M.; Garg, S.; Zomaya, A.Y.; Wang, L.; Moorsel, A.V.; Ranjan, R. Orchestrating Big Data Analysis Workflows in the Cloud: Research Challenges, Survey, and Future Directions. ACM Comput. Surv. 2019, 52, 1–41.
75. Rjoub, G.; Bentahar, J.; Wahab, O.A. BigTrustScheduling: Trust-aware big data task scheduling approach in cloud computing environments. Future Gener. Comput. Syst. 2020, 110, 1079–1097.
76. Cao, K.; Liu, Y.; Meng, G.; Sun, Q. An Overview on Edge Computing Research. IEEE Access 2020, 8, 85714–85728.
77. Fedushko, S.; Ustyianovych, T.; Syerov, Y.; Peracek, T. User-Engagement Score and SLIs/SLOs/SLAs Measurements Correlation of E-Business Projects Through Big Data Analysis. Appl. Sci. 2020, 10, 9112.
78. Zhang, C.; Li, M.; Wu, D. Federated Multidomain Learning With Graph Ensemble Autoencoder GMM for Emotion Recognition. IEEE Trans. Intell. Transp. Syst. 2022, 1–11.
79. Luo, X.; Zhang, C.; Bai, L. A fixed clustering protocol based on random relay strategy for EHWSN. Digit. Commun. Netw. 2023, 9, 90–100.

80. Chen, H.; Wen, J.; Pedrycz, W.; Wu, G. Big Data Processing Workflows Oriented Real-Time Scheduling Algorithm using Task-Duplication in Geo-Distributed Clouds. *IEEE Trans. Big Data* 2020, 6, 131–144.
81. Mishra, S.K.; Puthal, D.; Sahoo, B.; Jena, S.K.; Obaidat, M.S. An adaptive task allocation technique for green cloud computing. *J. Supercomput.* 2017, 74, 370–385.
82. Stavrinides, G.L.; Karatza, H.D. Scheduling Data-Intensive Workloads in Large-Scale Distributed Systems: Trends and Challenges. In *Modeling and Simulation in HPC and Cloud Systems*; Kołodziej, J., Pop, F., Dobre, C., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 19–43.
83. Yang, C.; Huang, Q.; Li, Z.; Liu, K.; Hu, F. Big Data and cloud computing: Innovation opportunities and challenges. *Int. J. Digit. Earth* 2017, 10, 13–53.
84. Hashem, I.A.T.; Yaqoob, I.; Anuar, N.B.; Mokhtar, S.; Gani, A.; Ullah Khan, S. The rise of «big data» on cloud computing: Review and open research issues. *Inf. Syst.* 2015, 47, 98–115.
85. Mazumdar, S.; Seybold, D.; Kritikos, K.; Verginadis, Y. A survey on data storage and placement methodologies for Cloud-Big Data ecosystem. *J. Big Data* 2019, 6, 1–37.
86. Natesan, G.; Chokkalingam, A. Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. *ICT Express* 2019, 5, 110–114.
87. Jafarnejad Ghomi, E.; Masoud Rahmani, A.; Nasih Qader, N. Load-balancing algorithms in cloud computing: A survey. *J. Netw. Comput. Appl.* 2017, 88, 50–71.
88. Alami Milani, B.; Jafari Navimipour, N. A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions. *J. Netw. Comput. Appl.* 2016, 64, 229–238.
89. Ahmad, N.; Che Fauzi, A.A.; Sidek, R.; Zin, N.; Beg, A. Lowest Data Replication Storage of Binary Vote Assignment Data Grid. *Commun. Comput. Inf. Sci.* 2010, 88, 466–473.

90. Mohammadi, B.; Navimipour, N.J. Data replication mechanisms in the peer-to-peer networks. *Int. J. Commun. Syst.* 2019, 32, e3996.
91. Campêlo, R.A.; Casanova, M.A.; Guedes, D.O.; Laender, A.H.F. A Brief Survey on Replica Consistency in Cloud Environments. *J. Internet Serv. Appl.* 2020, 11, 1.
92. Long, S.Q.; Zhao, Y.L.; Chen, W. MORM: A Multi-objective Optimized Replication Management strategy for cloud storage cluster. *J. Syst. Archit.* 2014, 60, 234–244.
93. Mokadem, R.; Hameurlain, A. A data replication strategy with tenant performance and provider economic profit guarantees in Cloud data centers. *J. Syst. Softw.* 2020, 159, 110447.
94. Wang, D.; Chen, J.; Zhao, W. A Task Scheduling Algorithm for Hadoop Platform. *J. Comput.* 2013, 8, 929–936.
95. Yi, P.; Ding, H.; Ramamurthy, B. Budget-Minimized Resource Allocation and Task Scheduling in Distributed Grid/Clouds. In *Proceedings of the 2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, Nassau, Bahamas, 30 July–2 August 2013; pp. 1–8.
96. Reddy, G. A Deadline and Budget Constrained Cost and Time Optimization Algorithm for Cloud Computing. *Commun. Comput. Inf. Sci.* 2011, 193, 455–462.
97. Xin, Y.; Xie, Z.Q.; Yang, J. A load balance oriented cost efficient scheduling method for parallel tasks. *J. Netw. Comput. Appl.* 2017, 81, 37–46.
98. Yang, S.J.; Chen, Y.R. Design adaptive task allocation scheduler to improve MapReduce performance in heterogeneous Clouds. *J. Netw. Comput. Appl.* 2015, 57, 61–70.
99. Smara, M.; Aliouat, M.; Pathan, A.S.; Aliouat, Z. Acceptance Test for Fault Detection in Component-based Cloud Computing and Systems. *Future Gener. Comput. Syst.* 2016, 70, 74–93.
100. R. Buyya, C. S. Yeo, and S. Venugopal, «Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities», in *Proc. IEEE Int. Conf. High Perform. Comput. Commun.*, Sep. 2008, pp. 5–13.

102. Z. Li, C. Wang and R. Xu, «Computation offloading to save energy on handheld devices: A partition scheme», in Proc. Int. Conf. Compilers, Arch. Synthesis Embedded Syst., 2001, pp. 238–246.

103. L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, «A framework for partitioning and execution of data stream applications in mobile cloud computing», ACM SIGMETRICS Perform. Eval. Rev., vol. 40, no. 4, pp. 23–32, Mar. 2013.

104. Брюханова Е.Р. Обобщенная экологическая модель динамической распределенной вычислительной системы / Антамошкин О.А., Брюханова Е.Р. // Моделирование, оптимизация и информационные технологии. 2023;11(4).

105. Винер, Н. Кибернетика, или управление и связь в животном и машине / Н. Винер. – Москва : Наука, 1983. – 344 с.

106. Карандеев, Д. Ю. Анализ показателей эффективности в задачах построения структур технических систем / Д. Ю. Карандеев. – Текст : непосредственный // Информационные технологии в моделировании и управлении : подходы, методы, решения : сборник научных статей I Всероссийской научной конференции : 12–14 декабря 2017 г. – В двух частях. – Тольятти, 2017. – Ч. 2. – С. 119–124.

107. Дулесов, А. С. Выбор структуры системы управления на основе меры неопределенности информации / А. С. Дулесов [и др.]. – Текст : непосредственный // Moderní vymoženosti vedy – 2013 : materialy IX mezinárodní vědecko-praktické conference. – Díl 71. Moderní informační technologie : Praha. – Р. 37–40.

108. Основы математического моделирования: учебное пособие / С. В. Звонарев. – Екатеринбург : Изд-во Урал. ун-та, 2019. – 112 с.

109. Антамошкин, А. Н. Метод ветвей и границ для задач условной псевдобулевой оптимизации с алгоритмически заданными функциями / А. Н. Антамошкин, И. С. Масич. – Текст : непосредственный // Решетневские чтения. – 2014. – Т. 2. – No 18. – С. 13–14.

110. Вильсон, А. Дж. Энтропийные методы моделирования сложных систем / А. Дж. Вильсон. – Москва: Наука, 1978. – 248 с. – Текст: непосредственный.

111. Воробьёв, В. А. Теория систем и системный анализ. Стохастические системы: учебное пособие / В. А. Воробьёв, Ю. В. Березовская; Сев. (Арктич.) федер. ун-тим. М. В. Ломоносова. – Архангельск: ИПЦ САФУ, 2012. – 147 с. – ISBN 978-5-261-00616-9. – Текст: непосредственный.
112. Вяткин, В. Б. Хаос и порядок дискретных систем в свете синергетической теории информации / В. Б. Вяткин. – Текст : непосредственный // Научный журнал КубГАУ. – Краснодар : КубГАУ, 2009. – No 47 (13).
113. Halpern J.Y., Fagin R. Modeling knowledge and action in distributed systems. – Distributed Computing 3, 1989. – P. 159–177.
114. Halpern J.Y., Moses Y.O. Knowledge and common knowledge in a distributed environment. – ACM 37(3). – 1990. – P. 549–587.
115. Artificial life / Ed. by C. Langton. – Redwood City: Addison-Wesley, 1988. – 655 p.
116. Пантелеев М.Г., Натей-Голенко М.А. Планирование действий интеллектуального робота в реальном времени / под ред. Е.И. Юревича // Материалы IX науч.-техн. конф. «Экстремальная робототехника». – СПб.: Изд-во СПбГТУ, 1998. – С. 232–237.
117. Денисов В.В., Пантелеев М.Г. Методы и алгоритмы оптимизации продукционных баз знаний для интеллектуальных роботов: Материалы IX науч.-техн. конф. «Экстремальная робототехника» / под ред. Е.И. Юревича. – СПб.: Изд-во СПбГТУ, 1998. – С. 70–76.
118. Pesin P., Tahon C., Yarassov V. Multi-Agent Architecture for the Activity Control of Complex Industrial Systems // Proc: of the International Workshop «Distributed Artificial Intelligence and Multi-Agent Systems» DAIMAS'97. – June 15–18, 1997. – St. Petersburg, Russia, 1997. – P. 182–191.
119. Bonasso R.P., Kortenkamp D., Muller J.P., Slack M. Experiences with an Architecture for Intelligent Reactive Agents. Ed. by M. Wooldridge.
120. J.P. Muller, M. Tambe // Proc. of the Intelligent Agents IL Agent Theories, Architectures, and Languages. IJCAF95 Workshop (ATAL). – August 19–20, 1995. Montreal, Canada. – Berlin: Springer, 1996. – P. 187–202.

121. Fisher K., Muller J.P., Pischel M. A Pragmatic BDFArchitecture. Ed. by M. Wooldindge, J.P. Muller, M. Tambe // Proc, of the Intelligent Agents IL Agent Theories Architectures, and Languages, IJCAF 95: Workshop (ATAL). – August 19–20, 1995. Montreal, Canada–Berlin: Springer, 1996. – P. 203–218.
122. Rao A.S. Georgeff M.P. Modelling Agents Within A BDI Architecture / Ed. by R. Fikes and E. Sandewall // Proc. of the 2nd International Conference on Principles of Knowledge. Representation, and Reasoning (KR-91). –Cambridge, Mass.: Morgan Kaufmann, 1991. – P. 473–484.
123. Pinson S.D., Louca J.A., Moractis P. A distributed decision support system for strategic planning // Decision Support Systems. The International Journal. – 1997. – Vol 20. – № 1. – P. 35–51.
124. Васильев А.Н. Принципы и техника нейросетевого моделирования / А.Н. Васильев, Д.А. Тархов. Москва: Высшая школа, 2014. – 218 с.
125. Галушкин А.И. Нейронные сети: основы теории. / А.И. Галушкин. М.: РИС, 2015. 496 с.
126. Yang H. et al. A deep convolutional neural network inspired by auditory perception for underwater acoustic target recognition // Sensors. 2019. Т. 19. No. 5. С. 1104.
127. Gorman R.P., Sejnowski T.J. Analysis of hidden units in a layered network trained to classify sonar targets // Neural networks. 1988. Т. 1. No. 1. С. 75-89.
128. Gimse H. Classification of Marine Vessels Using Sonar Data and a Neural Network :дис. NTNU, 2017.
129. Hu G., Wang K., Liu L. Underwater acoustic target recognition based on depthwise separable convolution neural networks //Sensors. 2021. Т. 21. No. 4. – С. 1429.
130. Real Doppler RAD-DAR database [Электронный ресурс]. – URL: <https://www.kaggle.com/datasets/iroldan/real-doppler-raddar-database>, свободный. Яз. англ. (дата обращения 25.02.2023).

131. CroplandMapping [Электронный ресурс]. – URL: <https://www.kaggle.com/datasets/pcbreviglieri/cropland-mapping>, свободный. Яз. англ. (дата обращения 25.02.2023).
132. Иванько А.Ф., Иванько М.А., Сизова Ю.А. Нейронные сети: Общие технологические характеристики // Научное обозрение. Технические науки. – 2019. – № 2. – С. 17-23; URL: <https://science-engineering.ru/ru/article/view?id=1236>
133. FRANK UHLIG, коды MATLAB для изменяющихся во времени матричных вычислений собственных значений через ZNN доступны по адресу http://www.auburn.edu/~uhligfd/m_files/T-VMatrixEigenv/
134. Тарасов В.Б. Современные направления искусственного интеллекта // Моделирование управления движениями человека / Под ред. М.П. Шестакова и А.Н. Аверкина. – М.: СпортАкадемПресс, 2003. – С. 9–43.
135. Predrag S. Stanimirovic Xue-z Hong Wang и Haifeng Ma, Комплекс ZNN для вычисления взвешенных псевдоинверсов, Применимый анализ. Дискрет. Мат., 13 (2019), с. 131-164.
136. Feng Xu, Zexin Li, Zhuoyun Nie, Hui Shao And Dongsheng Guo, Zeroing Neural Network для решения изменяющихся во времени линейных уравнений и систем неравенства, IEEE Trans. Невр. Нетв. Учиться. Сист., 30 (2019), с. 2346-2357.
137. Frank Uhlig, Построение поля значений разлагаемых и общих матриц, представлено, 8 с.
138. Min Sun и Jing Liu, новая шумоустойчивая нейронная сеть Чжана для изменяющегося во времени уравнения Ляпунова, Adv. Дифф. Экват., в печати (2020), 15 с.
139. NorduGrid, <https://www.nordugrid.org>
140. Frank Uhlig, Список перспективных конвергентных конечных разностных формул на http://www.auburn.edu/~uhligfd/m_files/ZNNSurveyExamples/Polyksrestcoeff3.m.

141. Брюханова Е.Р. Программный комплекс на основе подхода управления ресурсами распределенных динамических систем на базе обнуляющей нейронной сети № 2023660732 от 24.05.2023.

Приложение А. Свидетельство о государственной регистрации программы
для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2023663036

Программный комплекс на основе подхода управления
ресурсами распределенных динамических систем на базе
обнуляющей нейронной сети

Правообладатель: *Брюханова Евгения Романовна (RU)*

Автор(ы): *Брюханова Евгения Романовна (RU)*



Заявка № 2023660732

Дата поступления 24 мая 2023 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 19 июня 2023 г.

Руководитель Федеральной службы
по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 429b6a0fe3853164baf96f83b73b4aa7
Владелец **Зубов Юлий Сергеевич**
Действителен с 10.05.2023 по 02.08.2024

Ю.С. Зубов

Приложение Б. Акты об использовании результатов

Научно-образовательный центр
мирового уровня
ЕНИСЕЙСКАЯ СИБИРЬ
e-mail: yeniseysiberia@yandex.ru

Акт
о внедрении результатов диссертационного исследования
Брюхановой Евгении Романовны

Настоящим актом подтверждается, что разработанный Брюхановой Е.Р. метод управления ресурсами в распределенных динамических вычислительных системах и их программная реализация используются в деятельности научно-образовательного центра (НОЦ) «Енисейская Сибирь» при моделировании управления ресурсами распределенных динамических вычислительных систем.

Применение нового метода управления ресурсами на основе нейронных сетей обнуления, предложенного в рамках диссертационного исследования на соискание ученой степени кандидата технических наук Брюхановой Евгенией Романовной, позволили оценить возможность снижения выбросы углекислого газа при работе распределенных динамических вычислительных систем. Полученные результаты подчеркивают значимость экологически устойчивых подходов в управлении ресурсами и поднимают важный вопрос об ответственном использовании вычислительных мощностей с учетом их воздействия на окружающую природную среду.

Директор Проектного офиса
НОЦ «Енисейская Сибирь»,
доцент, канд. с.-х. н,
Верховец Сергей Владимирович



Подпись, печать

