

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

**АВТОРЕФЕРАТ**  
диссертации на соискание ученой степени  
кандидата технических наук

Красноярск – 2023

Работа выполнена в Федеральном государственном бюджетном образовательном учреждении высшего образования «Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева», г. Красноярск.

Научный руководитель: доктор технических наук, доцент  
**Антамошкин Олеслав Александрович**

Официальные оппоненты: **Дулесов Александр Сергеевич**  
доктор технических наук, доцент,  
ФГБОУ ВО «Хакасский государственный университет им. Н. Ф. Катанова», профессор  
кафедры информационных технологий и систем

**Царев Роман Юрьевич**  
кандидат технических наук, доцент,  
ФГБОУ ВО «МИРЭА – Российский технологический университет»,  
доцент кафедры прикладной математики

Ведущая организация: ФГБОУ ВО «Воронежский государственный технический университет»

Защита состоится «29» декабря 2023 года в 15:00 на заседании диссертационного совета Д 24.2.403.01, созданного на базе ФГБОУ ВО «Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева» по адресу: 660037, г. Красноярск, проспект имени газеты Красноярский рабочий, 31.

С диссертацией можно ознакомиться в библиотеке ФГБОУ ВО «Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева» и на сайте: <https://www.sibsau.ru>.

Автореферат разослан «\_\_\_» \_\_\_\_\_ 2023 г.

Ученый секретарь  
диссертационного совета

Панфилов Илья Александрович

## Общая характеристика работы

**Актуальность темы исследования.** В настоящее время эффективное использование вычислительных и сетевых ресурсов имеет прямое влияние на производительность пространственно-распределенных компьютерных систем, экономическую эффективность (минимизацию потребления различного рода ресурсов) и удовлетворение возможных ограничений, накладываемых на работу необходимостью решения задач, размещаемых для вычисления на узлах таких систем. Управление ресурсами в распределенных динамических системах является задачей с высокой вычислительной сложностью, требующей эффективных алгоритмов и методов для ее решения. Существующие методы управления ресурсами не всегда способны оперативно и адекватно реагировать на динамические изменения и обеспечивать оптимальное использование ресурсов при различной и постоянно меняющейся конфигурации сети.

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. Актуальные системы планирования распределенных вычислений оперируют такими критериями как производительность, надежность и экономичность, но не учитывают экологический фактор.

Исследование и учет экологических аспектов в динамических распределенных системах сталкивается с рядом вызовов. 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. 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 важных критериев оценки такого воздействия является эмиссия углекислого газа, как наиболее значимый по воздействию на климат. Это обуславливает необходимость создания модельно-алгоритмического обеспечения для управления распределенными динамическими вычислительными системами, важным критерием эффективности которых будет являться снижение выбросов углекислого газа в ходе их функционирования.

В настоящее время предложено и в достаточной мере исследовано большое количество различных подходов к управлению распределенными системами, основанных как на точных, так и на эвристических алгоритмах. Эффективность их работы различна и зависит от уровня, на котором они используются, а также от модели и структуры системы, стохастичности и устойчивости системы и прочих факторов. Очевидным решением проблемы повышения эффективности управления распределенными динамическими вычислительными системами является применение гибридного метода, объединяющего наиболее эффективные методы и алгоритмы, тем самым позволяющего добиться максимальной полезности на соответствующих уровнях управления. Кроме того, применение гибридного метода 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). Расширение класса решаемых задач инструментарием нейронных сетей обнуления является актуальной задачей в области системного анализа и управления.

Несмотря на заметный прогресс как в области оптимального управления различными системами на базе нейронных сетей обнуления, 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. Повышение осведомленности о расходуемой системой энергии, с получением которой связана 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 работе в режиме реального времени.

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

4. Необходимо повышать точность и адаптивность методов и алгоритмов, которые позволят более эффективно управлять ресурсами в сложных и динамических системах.

Таким образом, актуальной является задача разработки гибридного метода управления ресурсами распределенных динамических вычислительных систем, а применение нейронных сетей обнуления (НСО) в составе такого метода представляется перспективным благодаря таким свойствам НСО, как способность работать с дискретными равноудаленными данными, высокая скорость работы в системах реального времени, описанных в виде динамически изменяющихся матриц, способность с высокой точностью итеративно прогнозировать значения на следующем шаге времени, используя данные о состоянии системы на предыдущих шагах.

**Степень разработанности темы.** В область исследований оптимального управления ресурсами распределенных динамических вычислительных систем внесли вклад множество ученых и исследовательских коллективов из разных стран. Некоторые из известных имен в этой области включают Калмана Р., Беллмана Р., Курца Х., Понсе Р., Рекхенбаха С., Бойда С. В России исследованиями в области управления динамическими распределенными вычислительными системами занимаются Гавриш Д.А., Саранча С.Н., Клепиков А.К., Лемешко А.В., Вавенко Т.В., Литвинов К.А., Пасечников И.И., Миков А.И., Замятина Е.Б., Козлов А.А., Минухин С.В., Пархоменко С.С., Леденёва Т.М., Покусин Н.В., Фраленко В.П., Агроник А.Ю., Хританков, О.Я. Кравец, А.С. Дулесов и многие другие. Также исследования этой области представлены коллективами таких организаций: Исследовательский центр «Нелинейные и адаптивные системы», Лаборатория 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 Чжан Ю. и Ван Ц., профессор Обернского университета штата Улик Ф., профессор Нишского университета Республики Сербия Станимирович П., профессор Цзинаньского университета в Китае Джин Д., профессор Стайнер К. из Колумбийского университета в Нью-Йорке, Веллер А. из Университета Калифорнии в Лос-Анджелесе.

**Объектом исследования** являются ресурсы распределенных динамических вычислительных систем.

**Предметом исследования** являются инструменты управления ресурсами в распределенных динамических вычислительных системах.

**Целью исследования** является повышение эффективности управления ресурсами распределенных динамических вычислительных систем (РДВС) для снижения влияния их работы на окружающую среду.

Для достижения данной цели были поставлены и решены следующие задачи:

1. Анализ существующих методов управления ресурсами распределенных динамических вычислительных систем.
2. Разработка обобщенной математической и имитационной моделей 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 управления ресурсами в РДВС с учетом влияния их работы на окружающую среду.
5. Программная реализация полученного метода и алгоритма.
6. Проведение численных экспериментов по оценке эффективности применения разработанного метода и алгоритма управления 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; 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.

**Теоретическая значимость** данной диссертационной работы заключается в развитии методов оптимального управления ресурсами в распределенных динамических 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 метода управления ресурсами на основе нейронных сетей Чжана открывают 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.

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

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

**Основные положения, выносимые на защиту:**

1. Предложена обобщенная математическая модель распределенной динамической вычислительной системы, отличающаяся от известных учётом экологических последствий через расход затрачиваемой на вычисление 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», на конференциях «Ключевые тренды развития ИИ: наука и технологии», МГТУ им. Баумана, 2023, III Всероссийской научной конференции с международным участием «Наука, технологии, общество: Экологический инжиниринг в интересах устойчивого развития территорий» (НТО-III) 2022, Конференции студентов, аспирантов и молодых ученых «ПРОСПЕКТ СВОБОДНЫЙ-2022», Красноярск: СФУ, 2022, III Международной конференции «MIST: Aerospace-III 2019: Передовые технологии в аэрокосмической отрасли, машиностроении и автоматизации». На I Международном семинаре «MIST: Aerospace-I 2018: Передовые технологии в аэрокосмической отрасли, машиностроении и автоматизации».

Диссертация была обсуждалась на расширенных семинарах кафедры системного анализа и исследования операций СибГУ имени академика М.Ф. Решетнева.

**Публикации.** По теме диссертационной работы опубликовано 14 научных работ, из которых 3 в рецензируемых научных журналах рекомендованных ВАК России, 5 в изданиях индексируемых в наукометрических базах Scopus и Web of Science. 2 программные разработки зарегистрированы как программы для ЭВМ.

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

### **Основное содержание работы**

**Во введении** обоснована актуальность темы исследований, проводимых в рамках данной диссертационной работы, дана характеристика современного состояния исследований по изучаемой проблеме, сформулированы цель и задачи исследования, излагаются научная новизна, методы исследования, практическая и теоретическая значимости работы, приводится структура и содержание.

**Первая глава** посвящена обзору литературы. Рассматриваются источники, позволяющие сформулировать постановку задачи и производится анализ результатов работы различных методов решения задачи управления ресурсами.

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.

Работа распределенных динамических вычислительных систем требует больших вычислительных ресурсов. Алгоритмы управления ресурсами – это набор политик, процедур и правил, реализованных для назначения наилучшего ресурса для выполнения задач с целью достижения целей поставщика услуг и пользователя вычислительных мощностей. Существующие методы управления ресурсами учитывают множество показателей производительности подобных систем. Наиболее распространенные из них: пропускная способность каналов передачи данных, время выполнения вычислительных задач, время отклика, стоимость выполнения вычислений, бюджетные ограничения вычислений, сбалансированность вычислительной системы, отказоустойчивость, энергопотребление, объем передаваемых данных.

В данной работе рассматривается случай, когда стоит задача не только сохранить или увеличить производительность, но при этом минимизировать вредные выбросы, обусловленные работой вычислительных систем.

Следует отметить, что в современных распределенных динамических вычислительных системах возникает потребность в разработке обобщенной концептуальной модели, способной учесть экологические параметры в процессе ее функционирования. Такие системы могут включать в себя сети облачных вычислений, распределенные вычислительные платформы, интернет вещей и другие, где эффективность, производительность и воздействие на окружающую среду становятся ключевыми аспектами.

**Вторая глава** посвящена постановке задачи управления ресурсами в распределенных динамических вычислительных системах и исследованию взаимосвязей в системе путем имитационного моделирования и факторного анализа для дальнейшего определения и построения структуры подхода к минимизации углеродного следа (энергоэффективности) с помощью различных известных алгоритмов влияния на процессы при работе распределенных динамических вычислительных сетей и отдельных узлов.

Для общего представления задачи опишем концептуальную модель в общем виде, на ней представлен вход, ресурсы системы на рисунке 1.



Рисунок 1 – Концептуальная модель системы РДВС

Для разработки стратегии управления ресурсами РДВС создадим ее математическую модель. Для начала опишем характеристики отдельных узлов РДВС 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).

$\beta_{ij}$  – 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 [ $u_i$ ] – целочисленное значение, представляющее вычислительную мощность узла  $u_i$  (в миллионах инструкций в секунду – MIPS), которое вычисляется следующим образом:

$$PP[u_i] = N\_CPU[u_i] \times P\_CPU[u_j] \quad , \quad (2)$$

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. Предполагается, что распределенная вычислительная система включают в себя множество вычислительных узлов ( $CU$ ), 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  $CNi$  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 оценкой сложности  $k$ -ой задачи, измеряемой 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 выполнения каждой задачи в этой модели можно рассчитать следующим образом:

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

для каждого  $T_i$  предшествующего  $T_k$ , (5)

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:

$$FT(T_k, CU_i(P_j)) = \begin{cases} deadline(w_k), & \text{если } T_k \text{ является выходной задачей } EStime(CU_i(P_j)), \\ Ave\ Execution\ Time(CU_i(P_j)), & \text{в противном случае,} \end{cases} \quad (6)$$

где *Ave Execution Time* – среднее время выполнения, *deadline*( $w_k$ ) – крайний срок выполнения  $k$ -й задачи, *CT* – 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:

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

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

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  $k$ -ой задачи в момент времени  $t$ ,

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

$T(C_k)$  – строго регламентированное предельное время решения  $k$ -ой 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 ограничения и  $f(x, y)$  стремилось к минимуму.

Предложена модель процессов управления ресурсами в распределенных динамических вычислительных системах: создание математической модели, которая

описывает поведение ресурсов в системе и позволяет оценить их эффективное использование.

С использованием **имитационной модели** выполняется анализ производительности РДВС. Это включает в себя изучение характеристик системы в различных сценариях работы, выявление узких мест и оценку эффективности использования ресурсов. Для создания имитационной модели распределенной динамической вычислительной системы с использованием Simulink Matlab, необходимо определить следующие компоненты:

**Узлы.** Моделируются как сущности с характеристиками, зависящими от параметров ресурсов. Эти характеристики включают емкость хранилища, объем памяти, скорость чтения, скорость записи и производительность процессора.

**Сеть.** Моделируется как ресурс с характеристиками, такими как пропускная способность и вероятность передачи данных между узлами.

**Задачи.** Моделируются как процессы, которые выполняются на узлах. Задачи потребляют процессорное время, выполняют чтение и запись данных, и могут мигрировать между узлами.

**Ограничение и критерий оптимизации.** Включают в себя сохранение производительности и минимизацию энергопотребления (энергоэффективность). Ограничения на использование процессорного времени: каждый узел имеет ограниченные ресурсы, поэтому система должна учитывать потребление ресурсов и сроки выполнения: система должна выполнять задачи в срок.

Для имитационной модели были созданы 35 сценариев работы РДВС, которые были запущены на реальном наборе данных работы онлайн-платформы для решения вычислительных задач для сбора данных о системе и её ресурсах в разных состояниях.

На основе данных, полученных с имитационной модели, был выполнен факторный анализ с целью выявления наиболее значимых параметров и факторов, влияющих на производительность и минимизацию выбросов углекислого газа при работе РДВС, который показал, что наибольшее влияние на энергопотребление оказывают частота и напряжение работы процессоров. Репликация и миграция оказывают существенное влияние на время выполнения задач в системе. Оптимизация процессов репликации и миграции приводит к улучшению производительности даже при снижении энергопотребления.

Изучение современных научных публикаций определило необходимость использования гибридного управления 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 шаги 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.

Одна из проблем оптимизации управления в распределенных динамических вычислительных системах заключается в распределении входящих задач между вычислительными узлами и сокращении времени отклика на задачи, избегая при этом перегруженных или недогруженных узлов. Миграция данных требует времени, следовательно данные для решения задач необходимо как можно больше локализовывать, что в итоге приведет к сокращению 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.

Изучая эту проблему, было обнаружено, что задачи в очереди, которые должны быть выбраны, это задачи, которые будут выполняться на узле с минимальным 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 сохранении заданной производительности.

Чтобы лучше проиллюстрировать технологию, на рисунке 2 приведем пример схемы назначения наборов данных вычислительным узлам.

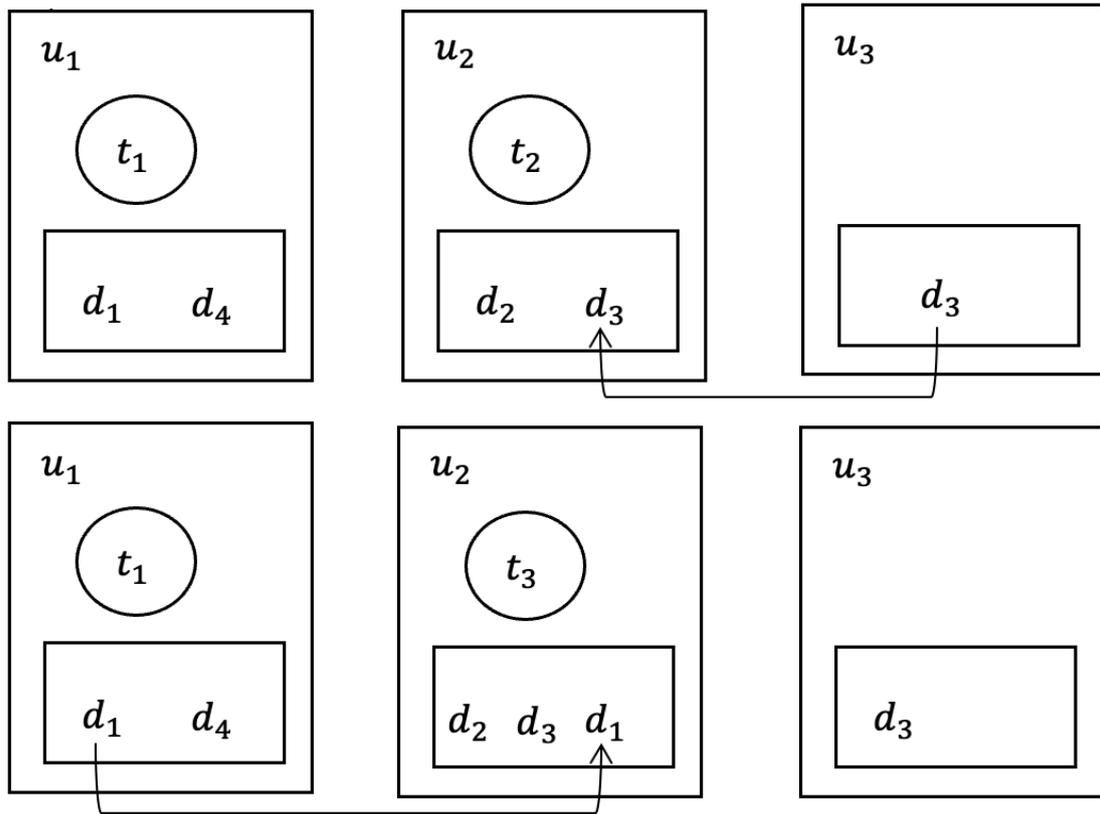


Рисунок 2 – Пример работы алгоритма планирования задач миграции, репликации, задержки и вычислений

Определим конфигурацию системы как  $Q$  – очередь онлайн-заданий,  $F$  – матрица назначения наборов данных задаче, и  $\psi$  – матрица назначения наборов данных узлам  $u$ .

$$Q = \{t_1, t_2, t_3\} \quad F = \begin{matrix} & t_1 & t_2 & t_3 \\ d_1 & \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \\ d_2 & \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \\ d_3 & \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \\ d_4 & \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{matrix} \quad \psi = \begin{matrix} & u_1 & u_2 & u_3 \\ d_1 & \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\ d_2 & \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \\ d_3 & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \\ d_4 & \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{matrix} .$$

Согласно алгоритму, узел  $u_1$  определен как оптимальный выбор для задачи  $t_1$ , как показано на рисунке 2, 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  $u_2$  в качестве наиболее эффективного решения. Таким образом, выполнение  $t_2$  на  $u_2$  приведет к минимальному 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  $u_2$ .

Далее 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  $\Delta$ . Несмотря на задержку, выполнение  $t_3$  на  $u_2$ , приведет

к меньшему 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 ( $\Delta$ ): время, в течение которого задача может ожидать доступности данного узла;
- время ожидания ( $WT$ ): сумма времени планирования ( $ST$ ) и времени задержки ( $\Delta$ );
- 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:

$$\min RT_{ij} = \min(WT_{ij} + TET_{ij}) = \min(ST_{ij} + \Delta_{ij} + DMT_{ij} + DAT_{ij} + ET_{ij}).$$

(10)

Ограничения, связанные с нашей целевой функцией, показаны в уравнениях (11) и (12):

$$RAM[t_i] \leq RAM[m_j] - \sum_{t_k \in TP_j} RAM[t_k], \quad (11)$$

$$Load_{min} \leq Load_j \leq Load_{max}. \quad (12)$$

Независимые пользователи отправляют задания на выполнение. Предполагается, 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;

$l_n$  – длина n-й задачи (в миллионе инструкций –  $MI$ );

$CPU [t_n]$  – 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  $t_n$ ;

$\alpha_n$  – индекс конечного назначения узла ( $m_{\alpha_n}$ ) задачи  $t_n$ ;

$\omega_n$  – 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:

$$load[u_i] = \frac{\sum_{t_j \in TP_i} l[t_j]}{PP[u_i]}, \quad (13)$$

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  $PP[u_i]$ .

Предполагается, 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 (14)

описывает вычисление матрицы  $\psi$ .

$$\psi = \begin{cases} 1 & \text{если } d_k \text{ входит в } u_i \\ 0 & \text{в других случаях} \end{cases}. \quad (14)$$

$F = \{f_{ki}\}$  – присвоение наборов данных матрице задач. 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).

$$f_{kn} = \begin{cases} 1 & \text{если } d_k \text{ требует } t_n \\ 0 & \text{в других случаях} \end{cases}. \quad (15)$$

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 потребителю 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  $t_n$ ).

6. Расчет необходимой частоты и напряжения. Динамическое управление частотой и напряжением процессора позволяет оптимизировать энергопотребление при выполнении задач. Рассчитываются необходимые частота и напряжение для выполнения задачи с минимальным энергопотреблением.

7. 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.



Рисунок 3 – Концептуальная схема управления ресурсами РДВС

Выявлены взаимосвязи, на их основе определены инструменты управления ресурсами в РДВС. Построенная 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 на рисунке 3.

**В третьей главе** проводится исследование возможности и эффективности решения, задачи управления ресурсами РДВС с учетом влияния ее работы на окружающую среду с помощью НСО. Было проведено сравнение эффективности применения НСО в сравнении как с классическими методами оптимизации, так и с различными нейросетевыми подходами. Разработано решение управления ресурсами в РДВС.

НСО позволяет находить неизвестные значения в непрерывно изменяющихся данных на основе известных равноудаленных данных. Его применение особенно эффективно в случаях, когда информация поступает от датчиков в форме дискретных равноудаленных сигналов.

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

**Работа с данным в реальном времени.** Поскольку НСО способен анализировать дискретные равноудаленные данные о состоянии системы, 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.

Для организации вычислительных экспериментов был разработан программный комплекс на основе подхода к управлению ресурсами распределенных динамических систем на базе НСО, зарегистрированный в государственном реестре программ для ЭВМ. Гибридное управление ресурсами основано на аппарате НСО по причинам того, что НСО предоставляет преимущества в контексте управления динамическими системами в реальном времени. Стоит отметить, что НСО работает на архитектуре, аналогичной рекуррентным нейронным сетям (РНН), представлено на рисунке 4. Архитектура РНН 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 позволяет минимизировать энергопотребление и, как следствие, выделение углекислого газа.

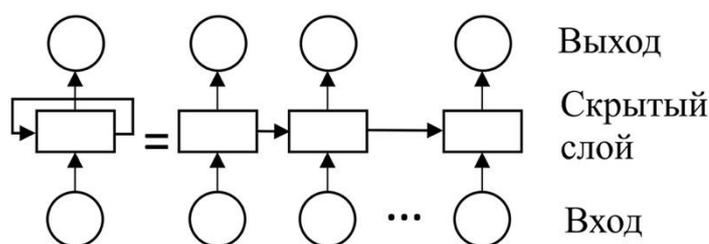


Рисунок 4 – Архитектура рекуррентной нейронной сети

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 сходящаяся 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

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

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 (16) сформируем функцию ошибки

$$E(t) = F(A(t), B(t), x(t), \dots) - g(t, C(t), u(t), \dots), \quad (17)$$

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

$$E(t) = -\eta E(t) \quad (18)$$

для некоторой константы  $\eta > 0$ .

**Шаг 3.** Решим экспоненциально затухающее уравнение ошибки дифференциального уравнения (18) шага 2 алгебраически для  $x(t_k)$ .

**Шаг 4.** Выберем перспективную сходящаяся конечно-разностная формула 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 линейное уравнение, полученное на шаге 5 для  $x(t_{k+1})$ .

**Шаг 7.** Увеличим  $k + 1$  до  $k + 2$  и увеличим все данные шага 6, затем решим обновленную рекурсию для  $x(t_{k+2})$ . Далее повторяем до тех пор, пока  $t \dots \geq t_f$ .

Таким образом, адаптивное управление ресурсами распределенных динамических вычислительных систем на базе аппарата НСО выполняет вычисления на каждом этапе алгоритма с учетом текущих значений параметров системы и задач.

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

Для проведения экспериментальных исследований были выбраны задачи минимизации времени отклика и изменения частоты в зависимости от длины (иначе – вычислительной сложности) задачи для оптимизации управления ресурсами в распределенной динамической вычислительной системе с использованием метода НСО. Таким образом были определены параметры оптимизации, а также введены ограничения на значения этих параметров.

Также была разработана математическая модель распределенной динамической системы и реализован оптимизационный подход на основе метода обнуления нейронных сетей. Модель была реализована на высокоуровневом языке программирования MATLAB, который широко используется в научных и инженерных приложениях, в том числе в области моделирования и оптимизации.

Для проведения экспериментов был выбран набор данных, полученных из статистики работы GRID сети NorduGrid состоящей из 139 территориально распределенных по всему миру объектов. Общее количество одновременно работающих узлов этой РДВС составляет порядка 2700000. Набор данных включает работу сети в течении 30 суток. Количество поступивших и решенных за это время задач превысило 1 миллиард.

Дополнительно в среде Virtual Internet Routing Lab был сгенерирован синтетический набор данных, состоящий из параметров системы и соответствующих им значений функции цели, и представляющих собой распределенную вычислительную

систему, состоящую из 3000000 вычислительных узлов и каналов связи между ними. Период работы сети составил 30 дней.

Затем была проведена серия экспериментов, в которых метод НСО был использован для оптимизации управления ресурсами в распределенной динамической системе. В результате экспериментов были получены значения оптимизируемых параметров и соответствующие значения функции цели.

**Таблица 1.** 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
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.

На Рисунке 5 представлены результаты работы метода НСО при решении задачи управления ресурсами. Видно, что по обоим критериям алгоритм быстро адаптируется и дальнейшее отклонение от оптимального значения составляет не более десятой доли процента.

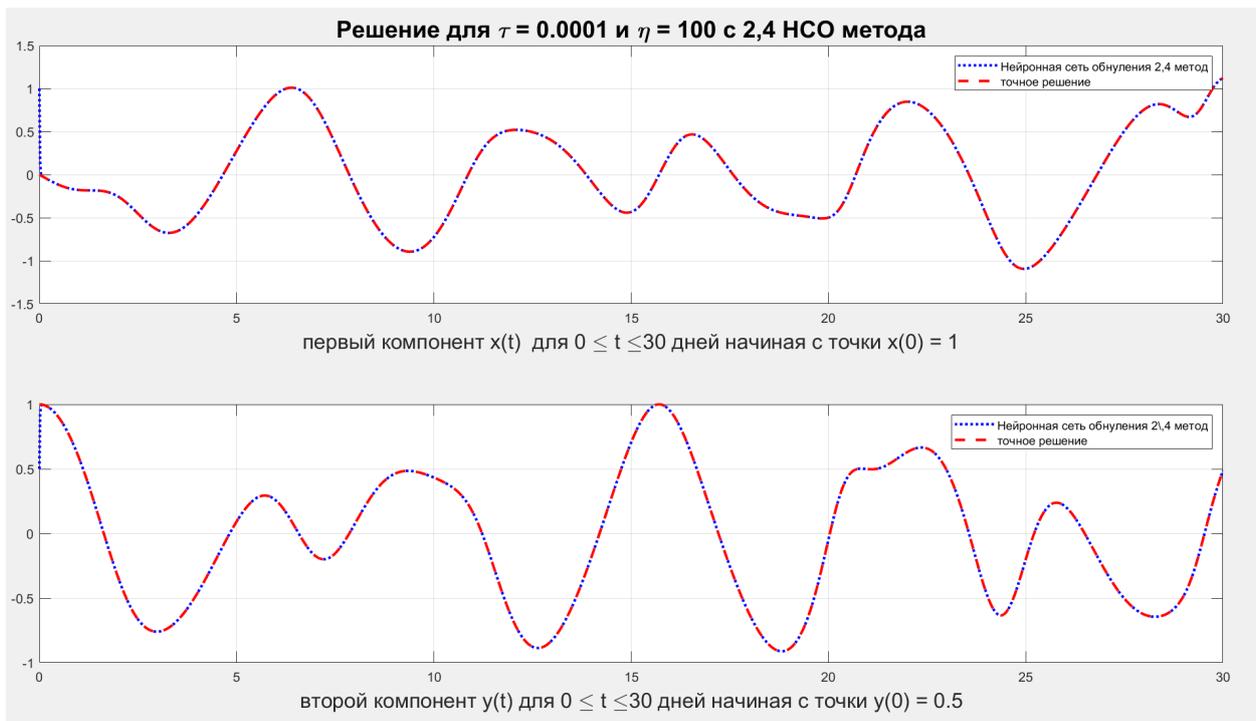


Рисунок 5 – Оценка качества работы НСО по критериям оптимальности – выбросы углекислого газа и времени решения задач

В зависимости от выбросов  $CO_2$  мы построили график задержки задачи, который представлен на рисунке 6. Видим, что для всех входных последовательностей задержка задачи соответствовала указанному требованию. Когда длина задач 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 6 – Задержка задачи с использованием предложенного подхода

Далее приведем сравнения работы различных алгоритмов при управлении ресурсами РДВС в целом. На основании набора данных, собранного в результате работы системы NorduGrid полным перебором вариантов, были получены значения распределения задач по узлам сети оптимальные, с точки зрения минимального выброса углекислого газа в атмосферу. Далее эту задачу решили с использованием алгоритмов НСО, рекуррентной нейронной сети и OSR (Online Switch Resist). На рисунке 7 представлена ошибка работы различных алгоритмов в течении 30 суток. Несмотря на то, что НСО показал лучшие результаты не на всем изучаемом промежутке, в среднем его ошибка заметно ниже конкурентов. Стоит так же отметить, что НСО значительно быстрее обучается и адаптируется к конфигурации РДВС.

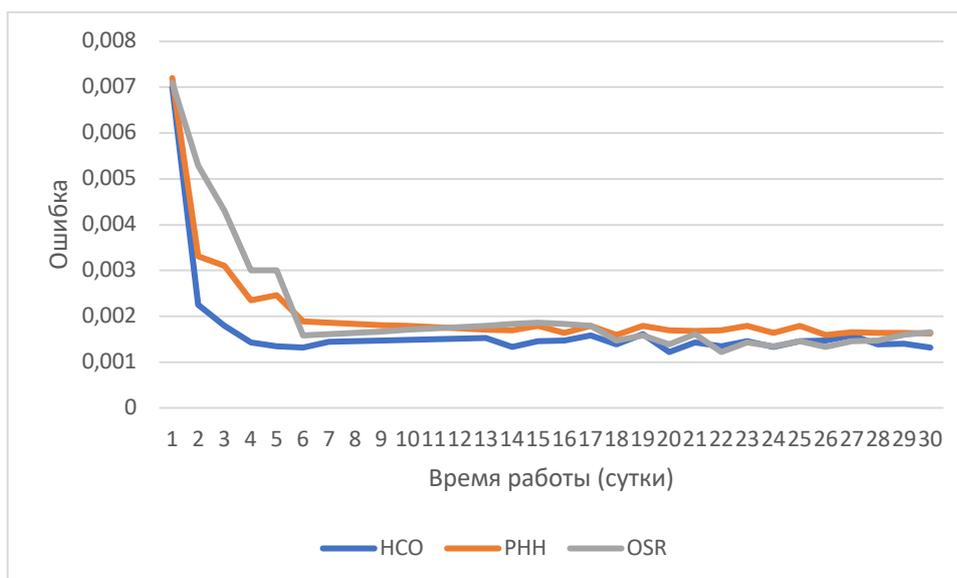


Рисунок 7 – Сравнение решения задач НСО с РНН и OSR

Анализ полученных результатов показал, что метод НСО 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.

**В заключении** приведены основные результаты работы.

1. В результате анализа существующих подходов и методов оптимального управления ресурсами распределенных динамических вычислительных систем были выявлены наиболее перспективные в рамках решения задачи управления ресурсами РДВС.

2. Разработаны обобщенная математическая и имитационная модели распределенной динамической вычислительной системы, описывающие ее ключевые ресурсы и параметры.

3. Факторный анализ обобщенной модели РДВС на основе данных имитационного моделирования позволил выявить информативные признаки РДВС и значительно сократить размерность задачи.

4. Предложен подход к динамическому управлению ресурсами распределенной динамической вычислительной системы, позволяющий учитывать требования к производительности, энергоэффективности и экологичности её работы.

5. Разработан алгоритм нейронных сетей обнуления для управления ресурсами РДВС.

6. Обнуляющий нейросетевой алгоритм управления ресурсами распределенной динамической вычислительной системы программно реализован.

7. Проведенные вычислительные эксперименты показали эффективность работы подхода и алгоритма на тестовых и реальных задачах.

Таким образом, разработана обобщенная математическая модель распределенной динамической вычислительной системы, позволяющая количественно оценить выбросы в атмосферу углекислого газа при решении любой задачи на любом из узлов РДВС. Предложена обобщенная 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: который позволяет, за счет комплексности и точности решения задачи, снизить выбросы углекислого газа в атмосферу до 7% в системе, состоящей из нескольких миллионов узлов.

### **Основные публикации по теме научно-квалификационной работы**

1. Брюханова Е.Р. Адаптивное управление ресурсами распределенных динамических систем на базе алгоритма обнуляющей нейронной сети / Антамошкин О.А., Брюханова Е.Р. // Системы управления и информационные технологии. 2023. № 1 (92)

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

3. Брюханова, Е. Р. Подход к использованию нейронных сетей обнуления для решения задачи управления ресурсами распределенной динамической вычислительной сети. Современные инновации, системы и технологии - Modern Innovations, Systems and Technologies, 3(4), 2023, 0301–0310. <https://doi.org/10.47813/2782-2818-2023-3-4-0301-0310>

4. Брюханова, Е. Р. Алгоритм адаптивного управлению ресурсами в распределенных динамических вычислительных системах на базе динамического управления частотой и напряжением. Информатика. Экономика. Управление - Informatics. Economics. Management, 2(4), 0201–0208, 2023, <https://doi.org/10.47813/2782-5280-2023-2-4-0201-0208>

5. Брюханова, Е.Р. Минимизация углеродного следа с применением обнуляющих нейронных сетей / Брюханова Е.Р. // Наука, технологии, общество: экологический инжиниринг в интересах устойчивого развития территорий Выпуск 6, 2022, DOI 10.47813/nto.3.2022.6.382-389

6. Bryukhanova, E.R. Minimizing the carbon footprint with the use of zeroing neural networks / E R Bryukhanova, O A Antamoshkin // The European Proceedings of Computers and Technology (EpTC), 2022 <https://doi.org/10.15405/epct.23021.20>

7. Bryukhanova, E.R. Methodology for automated classification of farmland based on Earth remote sensing data / O A Antamoshkin, E R Bryukhanova O A Stupin and N V Kamenskaya // IOP Conf. Series: Earth and Environmental Science, 2022, 981.

8. Bryukhanova, E.R. Simulation modelling of the heterogeneous distributed information processing systems / G.A. Ontuzheva, E.R. Bryukhanova, O.A. Antamoshkin, I.N. Rudov, N.O. Pikov // MISTAerospace 2018. IOP Conf. Series: Materials Science and Engineering 450 (2018) 052018: electronic edition, Krasnoyarsk

9. Брюханова Е.Р. Концептуальное проектирование интеллектуальной системы удаленной кардиореабилитации / А.С. Михалев, О.А. Антамошкин, Брюханова Е.Р., Д.С. Русин, С.Е. Головенкин // Научно-технический журнал «Системы управления и информационные технологии» Выпуск 1 (91), 2023, 59-64

10. Bryukhanova, E.R. Implementation of the Wölfflin formal statistical analysis method using fuzzy logic / O.A. Antamoshkin, E.R. Bryukhanova, V.O. Antamoshkina, N.O. Pikov, V.V. Kukartsev and V.V. Tynchenko// MISTAerospace 2019. IOP Journal of Physics: Conference Series 1399 (2019) 033103: electronic edition, Krasnoyarsk.

11. Bryukhanova, E.R. Evaluating possible classifications of websites by design type in electronic commerce / V V Kukartsev, E S Volneikina, E R Bryukhanova, S E Zinner, A I Stokan, and N O Pikov // J. Phys.: Conf. Ser. 2032 012126, 2021.

12. Bryukhanova, E.R. Modeling of the aggregator-platform for storage, analysis and processing of historical and cultural data "Siberiana» / E R Bryukhanova, O A Antamoshkin // The European Proceedings of Computers and Technology (EpTC), 2022, <https://doi.org/10.15405/epct.23021.10>

#### **Зарегистрированные программные системы:**

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

14. Брюханова Е.Р. Программа для ЭВМ «Система прогнозирования осложнений при заболеваниях сердечно-сосудистой системы» / Е.Р. Брюханова, Д.С. Русин, А.С. Михалев, О.А. Антамошкин, С.Е. Головенкин, Е.А. Сопов - М.: РОСПАТЕНТ. Свидетельство о государственной регистрации программы для ЭВМ № 2022683274 от 02.12.2022