На правах рукописи

ЛУЦЕНКО ВЛАДИСЛАВ ВЯЧЕСЛАВОВИЧ

РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ, МЕТОДОВ И АЛГОРИТМОВ ДЛЯ ПОВЫШЕНИЯ СКОРОСТИ ОБРАБОТКИ ДАННЫХ В ТУМАННЫХ ВЫЧИСЛЕНИЯХ С ИСПОЛЬЗОВАНИЕМ МОДУЛЯРНОЙ АРИФМЕТИКИ

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

Автореферат
диссертации на соискание ученой степени
кандидата физико-математических наук

Москва – 2025

Работа выполнена на кафедре вычислительной математики и кибернетики факультета математики и компьютерных наук имени профессора Н.И. Червякова федерального государственного автономного образовательного учреждения высшего образования «Северо-Кавказский федеральный университет» (ФГАОУ ВО СКФУ).

Научный руководитель: доктор физико-математических наук, доцент, **Бабенко Михаил Григорьевич**

Официальные оппоненты: **Феоктистов Александр Геннадьевич**, доктор технических наук, доцент, заведующий лабораторией параллельных и распределенных вычислительных систем, Федеральное государственное бюджетное учреждение науки Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук

Морозов Александр Юрьевич, доктор физико-математических наук, старший научный сотрудник отдела математического моделирования гетерогенных систем, Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН)

Ведущая организация: Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет»

Защита состоится «27» ноября 2025 года в 15 ч. на заседании диссертационного совета 24.1.120.01 при Федеральном государственном бюджетном учреждении науки Институте системного программирования им. В.П. Иванникова Российской Академии Наук по адресу: 115035, г. Москва, ул. Садовническая, д. 41, ст. 2.

С диссертацией можно ознакомиться в библиотеке и на сайте Федерального государственного бюджетного учреждения науки «Институт системного программирования им. В.П. Иванникова Российской Академии Наук».

Автореферат разослан «__» _____ 2025 г.

Ученый секретарь
диссертационного совета 24.1.120.01,
кандидат физико-математических наук Турдаков Д.Ю.

Общая характеристика работы

Актуальность работы. Развитие туманных вычислений (Fog Computing, FC) как перспективной технологии обработки данных в условиях роста количества устройств Интернета вещей (Internet of Things, IoT) и необходимости обработки информации в реальном времени привело к значительному увеличению требований к скорости и безопасности обработки и хранения информации [5, 12]. Туманные вычисления, обеспечивая обработку данных ближе к источнику их генерации, позволяют снизить задержки и уменьшить нагрузку на облачные серверы [11]. Однако такие системы сталкиваются с рядом задач, связанных с ограниченными вычислительными ресурсами граничных устройств и необходимостью обеспечения высокой скорости обработки данных при сохранении их конфиденциальности. Традиционные алгоритмы шифрования, зачастую оказываются слишком ресурсоемкими для устройств с ограниченной производительностью, что делает их применение в туманных вычислениях недостаточно эффективным [10].

Одним из направлений для решения этой задачи является применение системы остаточных классов (СОК). СОК позволяет выполнять параллельные вычисления над остатками числа, что значительно ускоряет выполнение арифметических операций в алгоритмах шифрования. Это особенно важно для туманных вычислений, где критически важны низкие задержки и высокая производительность. Кроме того, использование СОК может снизить энергопотребление, что актуально для IoT-устройств, работающих от батарей. Однако, несмотря на потенциальные преимущества, применение СОК в криптографических алгоритмах для туманных вычислений остается недостаточно изученным, что делает данное направление актуальным.

Для успешного внедрения СОК в туманные вычисления необходимо решить ряд задач, таких как оптимизация модульных операций СОК, а также разработка эффективных методов выполнения немодульных операций СОК, таких как обратное преобразование, определение знака числа и сравнение чисел. Решение этих задач открывает новые возможности для создания высокопроизводительных и безопасных систем обработки данных в туманных вычислениях.

Значительный вклад в развитие методов вычислений в СОК оказали российские ученые И.Я. Акушский, Д.И. Юдицкий, В.М. Амербаев, Н.И. Червяков, В.С. Князьков, А.А. Коляда, Ш.А. Оцоков, за рубежом – H.L. Garner, A. Omondi, N. S. Szabo D. Schoinianakis, G.C. Cardarilli, J.C. Bajard, G. Pirlo и другие.

Объектом диссертационного исследования является теория обработки данных в распределенных вычислительных системах.

Предмет исследования – методы и алгоритмы модулярной арифметики, направленные на ускорение обработки данных в туманных вычислениях.

Целью диссертационного исследования является повышение скорости работы узлов туманных вычислений в алгоритмах шифрования за счет оптимизации вычислительно сложных процедур модулярного кода и разработки программного обеспечения для их проектирования.

Научная задача диссертационной работы состоит в исследовании и разработке математических методов и алгоритмов выполнения вычислительно сложных операций СОК на основе функции ядра Акушского, способных повысить скорость при выполнении алгоритмов шифрования вычислительными узлами туманной среды, работающими в системе остаточных классов.

Для решения поставленной общей научной задачи была произведена ее декомпозиция на ряд **частных задач**:

1. Разработка алгоритмов для выбора оптимальных параметров СОК, таких как построение компактных базисов специального вида и поиск оптимальных весов для функции ядра Акушского.
2. Разработка и модификация методов и алгоритмов выполнения вычислительно сложных операций в СОК, таких как обратное преобразование из СОК в позиционную систему счисления (ПСС), деление чисел, определение знака числа и сравнение чисел для выполнения обработки данных.
3. Разработка программного комплекса выполнения модульных и немодульных операций в системе остаточных классов для распределенной обработки данных в туманных вычислениях.

Методология и методы исследования включают использование математического аппарата высшей алгебры, теории чисел, модулярной арифметики, теории алгоритмов, численных методов, линейной алгебры, теории вероятностей, а также методов математического моделирования и теории параллельных вычислений.

Соответствие паспорту научной специальности. Область исследования соответствует паспорту специальности 2.3.5 – «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей» по следующим пунктам:

8. Модели и методы создания программ и программных систем для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования.

9. Модели, методы, алгоритмы, облачные технологии и программная инфраструктура организации глобально распределенной обработки данных.

Научная новизна диссертационной работы:

1. Разработаны алгоритмы выбора оптимальных параметров функции ядра Акушского для вычисленных компактных базисов СОК.

2. Модифицированы методы и алгоритмы перевода из СОК в позиционную систему счисления, деления, определения знака и сравнения чисел в СОК, отличающиеся от известных меньшей размерностью операндов и эффективной реализацией операций без необходимости нахождения остатка по большому модулю, за счет выбора оптимальных параметров функции ядра Акушского.
3. Разработан программный комплекс для выполнения немодульных операций в туманных узлах, позволяющий повысить скорость распределенной обработки данных в туманных вычислениях.

На защиту выносятся следующие положения:

1. Метод перевода из системы остаточных классов в позиционную систему счисления на основе Китайской теоремы об остатках (КТО) и ранга числа функции ядра Акушского.
2. Алгоритмы определения знака числа для наборов модулей специального вида с использованием минимальной функции ядра Акушского.
3. Метод построения функций ядра Акушского для операции сравнения чисел.
4. Алгоритм построения компактных базисов специального вида для системы остаточных классов.
5. Алгоритмы поиска оптимальных весов функции ядра Акушского.

Достоверность полученных в диссертационной работе результатов обеспечивается строгостью математических доказательств и корректным использованием методологического аппарата исследований. Справедливость выводов относительно эффективности и корректности предложенных подходов проверена посредством компьютерного моделирования и экспериментальных исследований.

Теоретическая и практическая значимость. Теоретическая значимость диссертационной работы заключается в разработке математической модели, методов и алгоритмов, которые позволяют повысить скорость обработки данных в туманных вычислениях за счёт применения модулярной арифметики. Практическая ценность результатов заключается в их применимости для облачных и туманных структур с ограниченными вычислительными ресурсами, а также в таких направлениях, как криптография и искусственные нейронные сети. Разработанные методы и программные средства могут быть применены в других областях, требующих высокопроизводительной параллельной обработки данных.

Авторский вклад соискателя. Все изложенные в диссертационной работе результаты получены при непосредственном участии автора. Из результатов работ, выполненных коллективно, в диссертацию включены только полученные непосредственно автором. В работе [6] автором рассмотрена проблема критических ядер функции ядра Акушского и предложены методы их определения, необходимые для реализации операций сравнения чисел и определения их знака в системе

остаточных классов. В работе [10] исследованы методы обратного преобразования из СОК в ПСС. В работе [4] автором предложен метод обратного преобразования из системы остаточных классов в позиционную систему счисления с использование КТО и ранга числа, который вычисляется на основе функции ядра Акушского. В работе [3] автором предложена модификация итерационного деления в СОК с использованием функции ядра Акушского. В работе [2] автором исследована эффективность специальных наборов модулей системы остаточных классов. В работе [7] автором предложен метод определения знака числа в СОК, основанный на использовании приближенного ранга числа, вычисляемого с помощью функции ядра Акушского. В работе [1] автором предложено использование генетического алгоритма для поиска оптимальных весов функции ядра. Разработан программный комплекс вычислительных модулей туманной среды для оптимизации модульных операций СОК, а также выполнения модифицированных методов и алгоритмов вычисления немодульных операций в СОК, на который получены свидетельства о государственной регистрации программ для ЭВМ [17-24].

Внедрение. Результаты диссертационной работы были использованы при выполнении проектов: гранта РНФ № 19-71-10033 «Эффективная, безопасная и отказоустойчивая система распределенного хранения и обработки конфиденциальных данных с регулируемой избыточностью для проектирования мобильных облаков на маломощных вычислительных устройствах», гранта РНФ № 22-71-10046 «Разработка новых методов и алгоритмов для повышения надежности и безопасности хранения, передачи и обработки данных в туманных вычислениях», гранта РНФ № 24-21-00149 «Разработка модульных искусственных нейронных сетей ориентированных на туманные вычисления», гранта Северо-Кавказского федерального университета «Интеллектуальный блок управления распределенной системой хранения данных в гетерогенных средах с регулируемой избыточностью и безопасностью», гранта РНФ № 25-71-30007 «Новые технологии для проектирования облачных сервисов машинного обучения, сохраняющих конфиденциальность» (глава 2, глава 3). Кроме того, ряд результатов работы использован в Северо-Кавказском центре математических исследований в рамках соглашения № 075-02-2024-1451 с Министерством науки и высшего образования Российской Федерации.

Апробация работы. Основные результаты диссертационного исследования докладывались на международных и всероссийских конференциях, среди которых «Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2025)» (г. Пятигорск, Россия), «Открытая конференция ИСП РАН им. В.П. Иванникова (ISPRASOPEN 2024)» (г. Москва, Россия), «CPAMCS-2024: Current Problems in Applied Mathematics and Computer Systems» (г. Ставрополь, Россия), «Spring/Summer Young Researchers' Colloquium on Software Engineering

(SYRCoSE 2024)» (г. Ставрополь, Россия), «International Workshop on Advanced in Information Security Management and Applications (AISMA-2024)» (г. Алигарх, Индия, г. Ставрополь, г. Красноярск, Россия), «International Conference on Communication and Computational Technologies (ICCCT 2024)» (г. Джайпур, Индия), «Открытая конференция ИСП РАН им. В.П. Иванникова (ISPRASOPEN 2023)» (г. Москва, Россия), «Национальный Суперкомпьютерный Форум (НСКФ-2023)» (г. Переславль-Залесский, Россия), «Всероссийская научно-практическая конференция имени Жореса Алфёрова» (г. Санкт-Петербург, Россия), «CPAMCS-2023: Current Problems in Applied Mathematics and Computer Systems» (г. Ставрополь, Россия), «International Workshop on Advanced in Information Security Management and Applications (AISMA-2023)» (г. Алигарх, Индия, г. Ставрополь, г. Красноярск, Россия), «VII Всемирный Конгресс Математиков тюркского мира (TWMS Congress 2023)» (г. Туркестан, Казахстан), «International Conference on Mathematics and its Applications in New Computer Systems (MANCS 2021)» (г. Ставрополь, Россия).

Публикации по теме диссертации. Основные результаты по теме диссертационного исследования изложены в 24 публикациях, 5 из которых изданы в журналах, рекомендованных ВАК [1-5], 9 – в тезисах докладов конференций [8-16], 2 – в журналах, входящих в международные базы цитирования Web of Science и Scopus [6-7]. Получено 8 свидетельств о государственной регистрации программ для ЭВМ [17-24].

Моделирование и вычислительный эксперимент проведены на компьютере, оснащенным процессором Intel Core i7-7700HQ с тактовой частотой 2.80 ГГц, 8 ГБ оперативной памяти DDR4 с частотой 1196 МГц и твердотельным накопителем емкостью 512 ГБ, работающем под управлением Windows 10 Home Edition, с использованием языков программирования высокого уровня C++ и Python.

Структура диссертации. Диссертация состоит из введения, трех глав, заключения и семи приложений. Полный объем диссертации составляет 167 страниц, включая 18 рисунков и 42 таблицу. Список литературы содержит 119 наименований.

Основное содержание работы

Во введении обоснована актуальность темы диссертации, сформулированы цель и задачи работы, выбраны объект и предмет исследования, показана научная новизна, практическая и теоретическая ценность полученных результатов, приведены основные положения, выносимые на защиту.

В первой главе рассмотрена архитектура распределенной обработки данных. Несмотря на прогресс в области защиты данных в распределенных системах, таких как туманные, граничные и облачные вычисления, обеспечение безопасности в динамически изменяющихся условиях остается сложной задачей. Повышение

уровня безопасности данных IoT требует повышения скорости криптографических алгоритмов. Существующие подходы к оптимизации производительности криптографических алгоритмов можно условно разделить на три основные категории: аппаратные ускорители, программная оптимизация и гибридные методы. Однако эти методы либо требуют дорогостоящего и энергозатратного аппаратного обеспечения, либо обеспечивают лишь умеренное ускорение. Эта задача может быть решена за счет применения иного математического аппарата – СОК, которая, благодаря своим свойствам параллелизма, позволяет ускорить криптографические вычисления без значительного увеличения аппаратных затрат, что соответствует распределенной природе туманных вычислений.

Во второй главе приведены разработанные математические методы и алгоритмы выполнения немодульных операций в СОК.

Несмотря на преимущества СОК, ряд операций, называемых немодульными, является вычислительно сложным и требует тщательной оптимизации. Для выполнения немодульных операций требуется определение позиционной характеристики (ПХ) числа в СОК. Одним из инструментов нахождение ПХ является функция ядра Акушского [9].

В СОК, с набором модулей $\{p_1, p_2, \ldots, p_n\}$ и динамическим диапазоном $P = \prod_{i=1}^{n} p_i$, для числа X, такого, что $0 \leq X < P$, функция ядра Акушского задается следующим образом:

$$C(X) = \sum_{i=1}^{n} w_i \cdot \left\lfloor \frac{X}{p_i} \right\rfloor, \tag{1}$$

где целые числа w_i – постоянные, определяемые выбором точки интерполяции. Веса w_i задают вес каждого из частных $\left\lfloor \frac{X}{p_i} \right\rfloor$ в формуле (1), тем самым задавая функцию ядра и придавая ей различные свойства.

Используя функцию ядра можно получить информацию о ПХ числа как показано на рисунке 1. Функция ядра позволяет отобразить число в СОК на координатную прямую, где C_{min} и C_{max} минимальное и максимальное значение функции ядра для заданного набора весов.

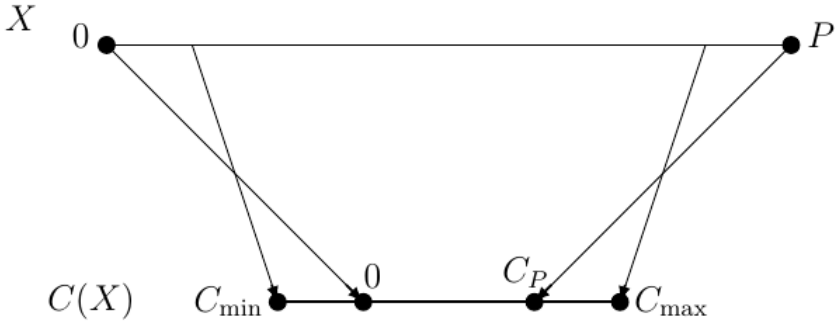

Рис 1. Отображение $0 \leq X < P$ на $C_{min} \leq C(X) \leq C_{max}$

Формула (1) имеет ограниченную применимость в практических вычислениях, поскольку требует знания позиционного представления числа X. Данную проблему можно решить путем применения Китайской теоремы об остатках к формуле (1). Тогда значение $C(X)$ при условии $0 \le C(X) < C_P$ можно вычислить с использованием формулы

$$C(X) \equiv \left| \sum_{i=1}^{n} k_i \cdot x_i \right|_{C_P}, \quad (2)$$

где $k_i = C(B_i) = \frac{B_i C_P}{P} - \frac{w_i}{p_i}, B_i = P_i \cdot \left| P_i^{-1} \right|_{p_i}, P_i = \frac{P}{p_i}$ и $C_P = C(P)$.

Рассмотрена проблема критических ядер функции ядра Акушского. В общем случае при определенных w_i, $C(X)$ может не удовлетворять условию $0 \le C(X) < C(P)$ для $X \in [0, P)$, что ограничивает возможность использования формулы (2). Для решения этой проблемы предложены алгоритмы определения критических ядер функции ядра Акушского [6].

Определять критические ядра можно с помощью усеченного перебора, реализация которого представлена в виде Алгоритма 1. Данный алгоритм универсален, однако обладает высокой вычислительной сложностью при большом количестве модулей в базисе СОК.

Алгоритм 1. Обнаружение критических ядер с помощью усеченного перебора.

Input: $\{p_1, p_2, \dots, p_n\}, \{w_1, w_2, \dots, w_n\}, P = \prod_{i=1}^{n} p_i$ для $i = \overline{1,n}$

Output: $lower\_critical\_cores, upper\_critical\_cores$

1. $lower\_critical\_cores, upper\_critical\_cores = 0$
2. **for** $X = 0, X < p_n, X++$ **do**
   - 2.1. $X \xrightarrow{\text{СОК}} (x_1, x_2, \dots, x_n)$
   - 2.2. $C(X) = X \cdot \sum_{i=1}^{n} \frac{w_i}{p_i} - \sum_{i=1}^{n} \frac{x_i \cdot w_i}{p_i}$
   - 2.3. **if** $C(X) < 0$ **then**
     - 2.3.1. $lower\_critical\_cores = 1$
     - 2.3.2. **break**
3. **for** $X = P - p_n, X < P, X++$ **do**
   - 3.1. $X \xrightarrow{\text{СОК}} (x_1, x_2, \dots, x_n)$
   - 3.2. $C(X) = X \cdot \sum_{i=1}^{n} \frac{w_i}{p_i} - \sum_{i=1}^{n} \frac{x_i \cdot w_i}{p_i}$
   - 3.3. **if** $C(X) \ge C(P)$ **then**
     - 3.3.1. $upper\_critical\_cores = 1$
     - 3.3.2. **break**
4. **return** $lower\_critical\_cores, upper\_critical\_cores$

В Теореме 1 сформулированы условия, которые позволят определить имеет ли функция ядра критические ядра.

Теорема 1. Для того чтобы функция ядра $C(X)$, определяемая весами $w_1, w_2, \dots, w_n$, не содержала критических ядер, необходимо и достаточно выполнение следующих условий для всех $k = 1,2, \dots, n$:

1. $\sum_{i=1,}^{n} w_i > 0$.
2. Отсутствие нижних критических ядер: $C(p_k) = \sum_{i=1}^{n} w_i \left\lfloor \frac{p_k}{p_i} \right\rfloor \geq 0$.
3. Отсутствие верхних критических ядер: $\sum_{i=1}^{n} \left(\left\lfloor \frac{p_k}{p_i} \right\rfloor + 1 \right) \cdot w_i - w_k > 0$.

На основе Теоремы 1, был предложен Алгоритм 2. Предложенный алгоритм в среднем на 99% быстрее предложенного алгоритма усеченного перебора.

Алгоритм 2. Обнаружение критических ядер на основе Теоремы 1.

Input: $\{p_1, p_2, \dots, p_n\}, \{w_1, w_2, \dots, w_n\}$, для $i = \overline{1, n}$

Output: $lower\_critical\_cores, upper\_critical\_cores$

1. $lower\_critical\_cores, upper\_critical\_cores = 0$
2. **for** $k = 1, k \leq n, k++$ **do**
   2.1. $C(p_k) = \sum_{i=1}^{n} w_i \left\lfloor \frac{p_k}{p_i} \right\rfloor$
   2.2. **if** $C(X) < 0$ **then**
      2.2.1. $lower\_critical\_cores = 1$
      2.2.2. **break**
3. **for** $k = 1, k \leq n, k++$ **do**
   3.1. $sum = 0$
   3.2. **for** $i = 1, i \leq n, i++$ **do**
      3.2.1. $sum = sum + \left(\left\lfloor \frac{p_k}{p_i} \right\rfloor + 1 \right) \cdot w_i$
   3.3. $sum = sum - w_k$
   3.4. **if** $sum \leq 0$ **then**
      3.4.1. $upper\_critical\_cores = 1$
      3.4.2. **break**
4. **return** $lower\_critical\_cores, upper\_critical\_cores$

Исследованы методы перевода из СОК в ПСС на основе КТО, обобщенной позиционной системе счисления, приближенной КТО, диагональной функции и функции ядра [15].

Согласно КТО число $X \xrightarrow{\text{СОК}} (x_1, x_2, \dots, x_n)$ может быть приведено в ПСС, с использованием формулы

$$X = \left| \sum_{i=1}^{n} B_i \cdot x_i \right|_P = \sum_{i=1}^{n} B_i \cdot x_i - r(X) \cdot P,$$

где $r(X) = \left\lfloor \frac{\sum_{i=1}^{n} x_i \cdot B_i}{P} \right\rfloor$ – ранг числа, отражающий, сколько раз сколько раз нужно вычесть диапазон P из суммы, чтобы вернуть ее в динамический диапазон.

С использование функции ядра Акушского число X в ПСС может быть вычислено следующим образом:

$$X = \left| \frac{P}{C(P)} \cdot \left(C(X) + \sum_{i=1}^{n} \frac{w_i}{p_i} \cdot x_i \right) \right|_P.$$

На основе КТО и ранга функции ядра Акушского предложен метод обратного преобразования из СОК в ПСС [4]. Согласно предложенному методу число X может быть вычислено следующим образом:

$$X = \sum_{i=1}^{n} B_i \cdot x_i - \check{r}(X) \cdot P, \tag{3}$$

где $\check{r}(X) = \left\lfloor \frac{\sum_{i=1}^{n} k_i \cdot x_i}{C(P)} \right\rfloor$.

Для использования данного метода доказана связь между рангами, вычисляемыми с помощью Китайской теоремы об остатках и функции ядра Акушского.

Теорема 2. Для функции ядра Акушского, у которой отсутствуют критические ядра, выполняется $\check{r}(X) = r(X)$.

Предложенный метод позволяет уйти от вычислительно сложных операций деления и нахождения остатка от деления по модулю, присутствующих в методе обратного преобразования с использованием функции ядра Акушского.

Исследован алгоритм итерационного деления в системе остаточных классов [3]. На каждом шаге итерационного деления возникает необходимость вычисления функции ядра от половины числа. Доказана Теорема 3, на основе которой возможна оптимизация итерационного деления.

Теорема 3. Функцию ядра от числа, деленного пополам, можно вычислить следующим образом:

$$C\left(\frac{X}{2}\right) = \frac{C(X) - \sum^{n}_{\substack{i=1 \\ |x_i|_2 = 1}} w_i}{2}.$$

Если в базисе СОК отсутствуют четные модули, то функцию ядра от числа, деленного пополам можно вычислить в соответствии с Теоремой 4.

Теорема 4. Для СОК $\{p_1, p_2, \ldots, p_n\}$, где p_i – нечетные числа и $Y = \frac{X}{2}$, значение функции ядра от Y равно

$$C(Y) = \left| \frac{1}{2} C(X) + \frac{1}{2} \sum_{|x_i|_2 = 1} k_i x_i \right|_{C_P}.$$

Рассмотрены алгоритмы определения знака числа в системе остаточных классов [7]. Построена функция ядра для набора модулей $\{2^n - 1, 2^{n+a}, 2^n + 1\}$ с заданными свойствами представленными Теоремой 5.

Теорема 5. В СОК $\{2^n - 1, 2^{n+a}, 2^n + 1\}$ $\forall X \in [0, P): 0 \le C(X) \le C_P < P$ и $C_P = 2^b$, то $b \ge 2n + a$ и $w_1 = 2^{|b-a-1|_n}, w_2 = 0, w_3 = -2^{|b+n-a-1|_n}$.

Согласно Теореме 1 представленная функция ядра будет иметь верхние критические ядра. Для того чтобы использовать формулу (2) доказана Теорема 6.

Теорема 6. В СОК с основаниями $\{2^n - 1, 2^{n+a}, 2^n + 1\}$ $\forall X \in [0, P): 0 \le C(X) \le C_P$ с весами $w_1 = 1, w_2 = 0, w_3 = -1$, если при вычислении ядра числа $X = (x_1, x_2, x_3)$ по формуле (2) получится значение $C(X) = 0$, то в случае $x_1 \ge x_3$ имеем $C(X) = 0$, а в противном случае $C(X) = C_P$.

На основе предложенной функции ядра построен Алгоритм 3, с помощью которого можно определить знак числа в СОК.

Алгоритм 3. Определение знака числа с использованием функции ядра для набора модулей $\{2^n - 1, 2^{n+a}, 2^n + 1\}$.

Input: (x_1, x_2, x_3)

Data: $\{p_1, p_2, p_3\}, P = \prod_{i=1}^{n} p_i, C_P = 2^{n+a+1}, C\left(\frac{P}{2}\right) = n + a, N = n + a + 1, \{k_1, k_2, k_3\}$

Output: $sign$

1. $C(X) = (k_1 x_1 + k_2 x_2 + k_3 x_3) \wedge (2^N - 1)$
2. **if** $C(X) = 0$ **and** $x_1 < x_3$ **then**
   2.1. $C(X) = C_P$
3. **else**
   3.1. $C(X) = 0$
4. **if** $C(X) < C\left(\frac{P}{2}\right)$ **then**
   4.1. $sign = 0$
5. **else**
   5.1. $sign = 1$
6. **return** $sign$

Построена функция ядра для набора модулей $\{2^n - 1, 2^{n+1} - 1, 2^{n+a}\}$ с заданными свойствами, представленными Теоремой 7 и Теоремой 8.

Теорема 7. Если для функции $C(X) = w_1 \left\lfloor \frac{X}{2^n - 1} \right\rfloor + w_2 \left\lfloor \frac{X}{2^{n+1} - 1} \right\rfloor + w_3 \left\lfloor \frac{X}{2^{n+a}} \right\rfloor$ выполняются следующие условия $\forall X \in [0, P): 0 \le C(X) \le 2^b$ и $C(P) = 2^b$, то $b \ge 2n + a$ и $w_1 = 2^{|b-a|_n}$, $w_2 = -2^{|b-a|_n}$ и $w_3 = 0$.

Теорема 8. В СОК с основаниями $\{2^n-1, 2^{n+1}-1, 2^{n+a}\}$ $\forall X \in [0, P)$ выполняются следующие условия:

1. Если $C(X) > 2^{2n+a-1}$, то $X > (2^n-1)(2^{n+1}-1)2^{n+a-1}$.
2. Если $C(X) < 2^{2n+a-1}$, то $X < (2^n-1)(2^{n+1}-1)2^{n+a-1}$.

Сформулирована Теорема 9, с использованием которой возможно применение формулы (2) для построенной функции ядра.

Теорема 9. В СОК с основаниями $\{2^n-1, 2^{n+1}-1, 2^{n+a}\}\forall X \in [0, P): 0 \leq C(X) \leq C_P$ с весами $w_1 = 1, w_2 = -1, w_3 = 0$, если при вычислении ядра числа $X = (x_1, x_2, x_3)$ по формуле (2) получится значение 0, то в случае $x_1 \geq x_2$ имеем $C(X) = 0$, а в противном случае $C(X) = C_P$.

Алгоритм определения знака числа с использованием предложенной функции ядра аналогичен Алгоритму 3.

Предложенные функции ядра со значением $C_P = 2^N$ позволяют сократить размеры операндов, а также снизить сложность операции нахождения остатка от деления с $O(n^2)$ до $O(n)$.

Рассмотрена проблема монотонности функции ядра для сравнения чисел в СОК. Доказаны свойства СОК, которые представлены в Теореме 10 и Теореме 11.

Теорема 10. Для того чтобы выполнялось неравенство $\sum_{i=1}^{n} \left| |2^N|_{p_i} \cdot \left|P_i^{-1}\right|_{p_i} \right|_{p_i} P_i < P$ необходимо и достаточно, чтобы $r(2^N) = 0$.

Теорема 11. $r(2^N) = 2^N r(1) - \sum_{i=1}^{n} \left\lfloor \frac{2^N}{p_i} \left|P_i^{-1}\right|_{p_i} \right\rfloor$.

С использованием предложенных свойств, разработан метод построения минимальных функций ядра для операции сравнения чисел, у которых выполняется условие $C_P = C(P) = 2^{N_{min}}$. Данный метод основан на Теореме 12.

Теорема 12. Если $r(2^N) = 0$, то $N_{min} = N - \log_2 \text{НОД}(2^N, w_1, w_2, \dots, w_n)$, $w_i^* = \frac{w_i}{\text{НОД}(2^N, w_1, w_2, \dots, w_n)}$ и $2^{N_{min}} = \sum_{i=1}^{n} w_i^* P_i$, где $N \geq \lceil \log_2 \frac{P}{p_n} \rceil$.

В таблице 1 приведены некоторые наборы модулей и веса, найденные на основании Теоремы 12 с целью построения функций ядра для операции сравнения чисел.

Таблица 1 – Веса функций ядра для сравнения чисел

№	Набор модулей	w	N_{min}	$\lfloor \log_2 P \rfloor$
1	$\{2^n-1, 2^{n+a}, 2^n+1\}$	$\{1,0,1\}$	$2n+a+1$	$3n+a-1$
2	$\{2^{n+1}-1, 2^n, 2^n-1\}$	$\{2^n-1,0,1\}$	$3n$	$3n$
3	$\{2^{2n}, 2^n-1, 2^{n-1}-1\}$	$\{0, 2^{n-1}-1, 1\}$	$4n-2$	$4n-2$

Предложенный метод позволяет найти минимальную функцию ядра для сравнения чисел в СОК, что в свою очередь уменьшает размер операндов и делает возможным выполнение операций по модулю $2^{N_{min}}$.

Таким образом, реализованы все немодульные операции, необходимые для построения высокоскоростной системы обработки данных, работающей в системе остаточных классов.

В третьей главе рассмотрена реализация разработанных во второй главе алгоритмов в виде модулей программного комплекса для вычислительных узлов туманной среды. Модули разработаны с целью выполнения арифметических, в том числе немодульных, операций в СОК. Главное преимущество СОК заключается в возможности параллельной реализации операций сложения, вычитания и умножения по независимым модулям.

Работа вычислительной системы, работающей на основе модулярной арифметики, в виде структурной схемы показана на рисунке 2.

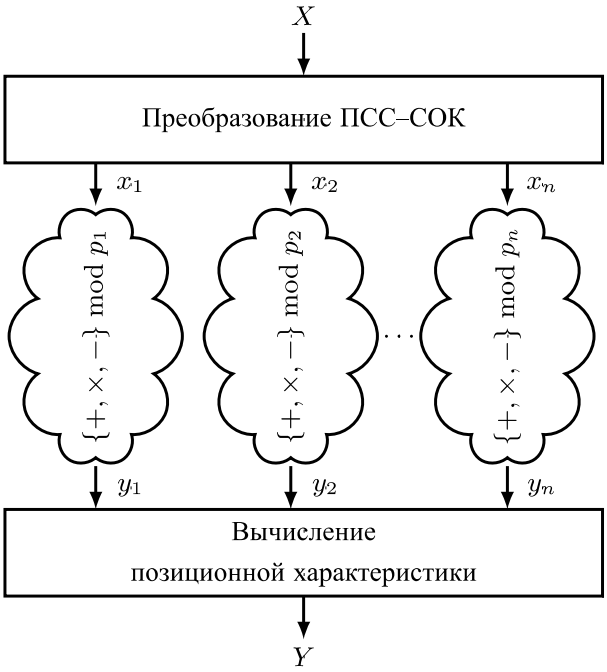

Рис 2. Структурная схема вычислительной системы на основе модулярной арифметики

Вычислительная система, работающая с использованием модулярной арифметики, включает несколько блоков обработки данных. Входной блок выполняет преобразование чисел из ПСС в модулярное представление. Основные вычислительные блоки системы реализуют модульные операции сложения, вычитания и умножения в соответствии с свойствами СОК. Особое значение имеет блок определения позиционных характеристик, который обеспечивает выполнение немодульных операций.

Первым этапом алгоритма проектирования вычислительной системы, работающей на основе модулярной арифметики, является выбор базиса СОК.

Вторым этапом алгоритма проектирования вычислительной системы является выбор оптимальных весов для функции ядра Акушского.

Третий этап предполагает реализацию модульных арифметических операций (сложения, вычитания, умножения) в распределенной среде, включая туманные вычисления. Благодаря независимости вычислений по каждому модулю обеспечивается высокая скорость операций.

Заключительный этап включает выбор алгоритма для обратного преобразования в ПСС, деления, определения знака числа, а также сравнения чисел в СОК.

Для реализации высокоскоростной системы на основе методов и алгоритмов, представленных во второй главе, был создан программный комплекс для работы в системе остаточных классов. Его архитектура представлена на рисунке 3. В открытом доступе есть библиотека NTL для языка программирования C++, предназначенная для работы с различными областями теории чисел. Доступные методы этой библиотеки, поддерживающие вычисления в СОК, были использованы в качестве аналогов для сравнения, отсутствующие методы были реализованы самостоятельно.

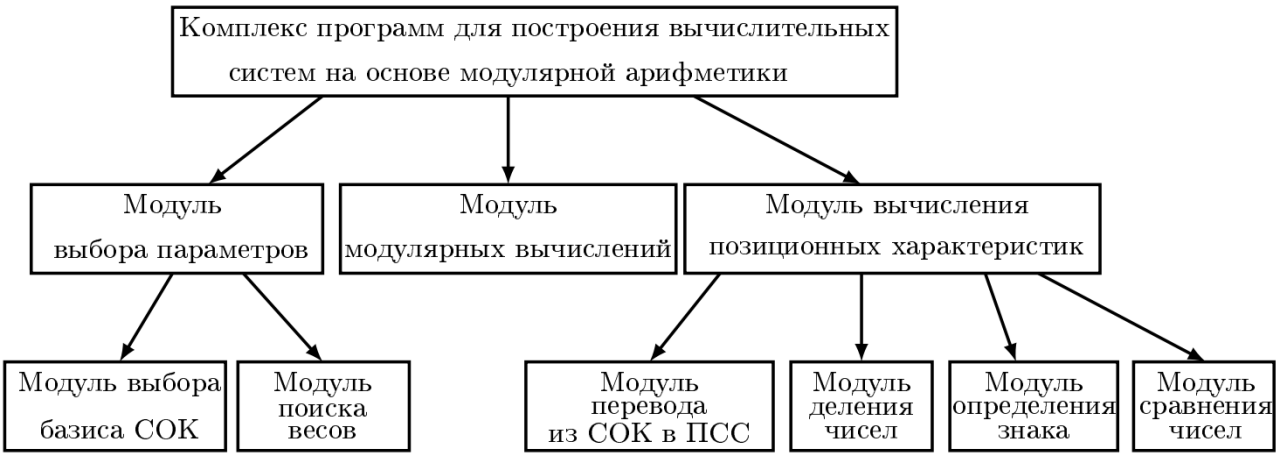

Рис 3. Структура программного комплекса для построения вычислительных систем на основе модулярной арифметики

Модуль выбора параметров представлен программами [19, 22]. Программа [22] позволяет находить оптимальные веса для функции ядра с использованием генетического алгоритма. В программе [19] реализованы алгоритмы определения критических ядер при построении функции ядра Акушского.

Модуль модулярных вычислений представлен программой [21]. В программе [21] реализовано определения переполнения при сложении чисел в системе остаточных классов с использованием функции ядра Акушского.

Модуль вычисления позиционных характеристик содержит модуль перевода из СОК в ПСС, модуль деления чисел, выраженный в программах [20, 23, 24], а также модули определения знака числа и сравнения чисел, представленные комплексом программ [17, 18].

Для оценки эффективности разработанных методов и алгоритмов проведено моделирование вычислительных модулей туманной среды. Для моделирования использовалась следующая тестовая инфраструктура: Intel Core i7-7700HQ с тактовой частотой 2.80 ГГц, 8 ГБ оперативной памяти DDR4 с частотой 1196 МГц, твердотельный накопитель емкостью 512 ГБ, Windows 10 Home Edition. Для моделирования использованы языки программирования высокого уровня C++ и Python.

Проведено исследование эффективности наборов модулей СОК специального вида для арифметических операций сложения, вычитания и умножения [2]. Для исследуемых динамических диапазонов от 8 до 32 бит набор $\{2^n - 1, 2^n, 2^n + 1\}$ демонстрирует лучшие результаты во всех операциях.

Для программной реализации СОК возможно использование наборов модулей общего вида. За счет большего количества модулей, можно достичь большей степени параллелизма. Однако для эффективной реализации наборов модулей общего вида необходимо выполнение условия компактности. Компактным набор модулей является, если $p_n < 2p_1$. Для построения компактных наборов модулей был использован метод, базирующийся на теореме Диемитко, который представлен в виде Алгоритма 4.

Алгоритм 4. Нахождение простых чисел с использованием теоремы Диемитко (PrimeNumbers).

Input: t – требуемая размерность простого числа, q – простое число

Output: p – простое число

1. $\xi = random(0, 1)$
2. $R = \left\lceil \frac{2^{t-1}}{q} \right\rceil + \left\lceil \frac{2^{t-1}\xi}{q} \right\rceil$
3. **if** $R \neq 0 (mod\ 2)$ **then**
   3.1. $R = R + 1$
4. $u = 0$
5. $n = (R + u)q + 1$
6. **if** $n > 2t$ **then**
   6.1. Возврат на шаг 1
7. **if** $2^{(n-1)} \equiv 1 (mod\ n)$ **and** $2^{(R+u)} \not\equiv 1 (mod\ n)$ **then**
   7.1. $p = n$
   7.2. **break**
8. **else**
   8.1. $u = u + 2$
   8.2. Возврат на шаг 5
9. **return** p

Алгоритм 5 позволяет вычислить компактные базисы с модулями вида $p_i = Rq_i + 1$.

Алгоритм 5. Построение компактных базисов СОК.

Input: $\{q_1, q_2, \dots, q_n\}, t$ – требуемая размерность простого числа

Output: $b = \{p_1, p_2, \dots, p_k\}$

1. $p = PrimeNumbers(q_1, t)$
2. p добавить в b
3. **for** $i = 2, i \leq n, i++$ **do**
   3.1. $p = PrimeNumbers(q_i, t)$
   3.2.**if** $p < 2p_1$ **then**
      3.2.1. p добавить в b
4. **return** b

Предложенный алгоритм позволяет снизить время выбора набора модулей СОК в среднем на 17% в сравнении с методом на основе чисел Мерсенна и на 73% – с методом общей фильтрации. Также алгоритм позволил получить ускорение при выполнении модульных операций в среднем на 12% по сравнению с использованием модулей специального вида $\{2^n - 1, 2^n, 2^n + 1\}$.

Оптимальными весами для функции ядра Акушского являются веса, при которых $C(P) = 2^N$. Для поиска оптимальных весов можно использовать метод, основанный на Теореме 12, однако при большом количестве модулей в базисе СОК, поиск оптимальных весов становиться вычислительно сложной задачей. Предложено использование метода Монте-Карло или генетического алгоритма для поиска оптимальных весов [1].

Поиск оптимальных весов с использованием метода Монте-Карло представлен Алгоритмом 6.

Алгоритм 6. Метод Монте-Карло для поиска оптимальных весов функции ядра Акушского.

Input: $\{p_1, p_2, \dots, p_n\}, max\_iterations, weight\_limit$

Data: $P = \prod_{i=1}^{n} p_i$, $P_i = \frac{P}{p_i}$ для $i = \overline{1, n}$

Output: $\{w_1, w_2, \dots, w_n\}, N$ – если найдены оптимальные веса

1. **for** $i = 1, i \leq max\_iterations, i++$ **do**
   1.1.$w_i = random(0, weight\_limit)$
   1.2.$C_P = \sum_{i=1}^{n} w_i P_i$
   1.3.**if** $C_P > 0$ **and** $\left(C_P \wedge (C_P - 1)\right) = 0$ **then**
      1.3.1. $N = \log_2 C_P$
      1.3.2. **return** $\{w_1, w_2, \dots, w_n\}, N$

Поиск оптимальных весов с использованием генетического алгоритма представлен Алгоритмом 7.

Алгоритм 7. Генетический алгоритм для поиска оптимальных весов функции ядра.

Input: $\{p_1, p_2, \dots, p_n\}, population\_size, weight\_limit$

Data: $P = \prod_{i=1}^{n} p_i$, $P_i = \frac{P}{p_i}$ для $i = \overline{1,n}$

Output: $\{w_1, w_2, \dots, w_n\}, N$ – если найдены оптимальные веса

1. **for** $i = 1, i \leq population\_size, i++$ **do**

   1.1. $w_i = random(0, weight\_limit)$

Оценка приспособленности:

2. $C_P = \sum_{i=1}^{n} w_i P_i$
3. **if** $C_P > 0$ **and** $\left(C_P \wedge (C_P - 1)\right) = 0$ **then**

   3.1.1. $N = \log_2 C_P$

   3.1.2. **return** $\{w_1, w_2, \dots, w_n\}, N$

Селекция:

4. $n = population\_size$
5. $f_{w_i} = \frac{C_P}{\sum_{i=1}^{n} C_P}$ – вероятность выбора w_i веса
6. $crossover\_point = random\left(0, len(p_1, p_2, \dots, p_n)\right)$
7. $new\_population = new\_population + w_i$

Мутация:

8. **for** $i = 1, i \leq new\_population, i++$ **do**

   8.1. Замена случайного веса w_i в диапазоне $weigh\_limit$

   8.2. $population = population + new\_population$
9. Возврат на шаг 2

Для оценки эффективности предложенных алгоритмов проведен замер времени поиска оптимальных весов, с использованием языка программирования Python. В среднем, использование генетического алгоритма позволяет сократить время поиска оптимальных весов на 71.2% по сравнению с алгоритмом, основанным на методе Монте-Карло.

Для подтверждения статистической значимости исследования всех немодульных операций СОК для каждого набора модулей проводится 10^6 циклов, в каждом из которых выполняется по 10^4 реализаций. Результатом является среднее время выполнения.

Проведено моделирование перевода из СОК в ПСС для метода на основе КТО и ранга функции ядра Акушского, который представлен формулой (3). Проведено два этапа моделирования. На первом изучается производительность семи наборов модулей специального вида $\{2^n - 1, 2^n, 2^n + 1\}$ с размерностью динамического диапазона от 16 до 64 бит. На втором этапе проводится анализ семи наборов моделей общего вида, варьируя от 3 до 9 модулей, где каждый модуль имеет 8-

битную длину. На рисунках 4-5 представлены результаты моделирования обратного преобразования из СОК в ПСС.

Алгоритм на основе КТО и ранга функции ядра имеет лучшие результаты в диапазоне от 16 до 64 бит. Он на 10.5% в среднем быстрее КТО и на 19.9% быстрее приближенной КТО. В случае динамического изменения числа восьми битных модулей, алгоритм на основе КТО и ранга функции ядра Акушского также показывает лучшие результаты. Он быстрее КТО на 7.2% и быстрее приближенной КТО на 15.8%.

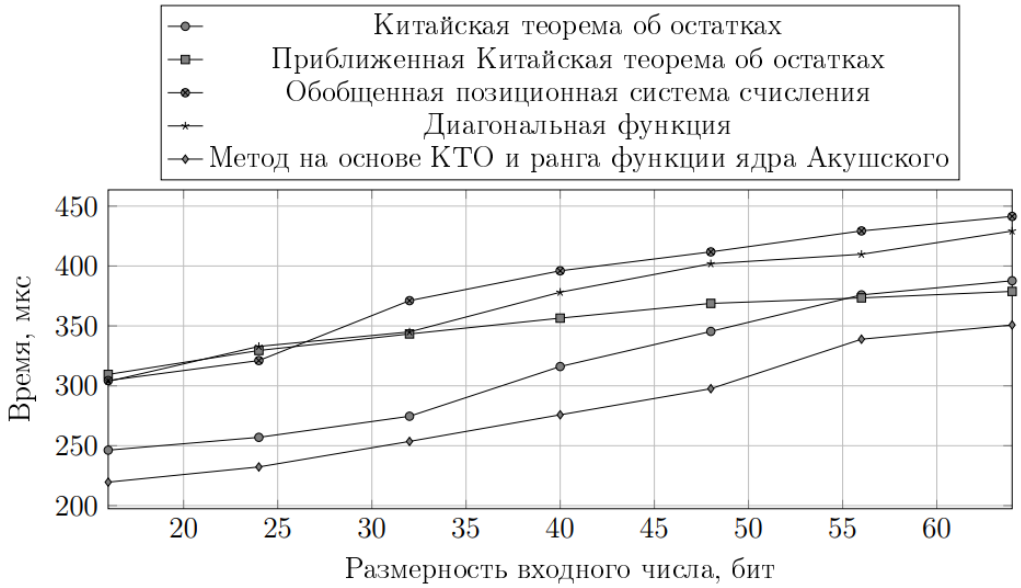

Рис 4. Сравнение времени для методов обратного преобразования из СОК в ПСС, первый этап

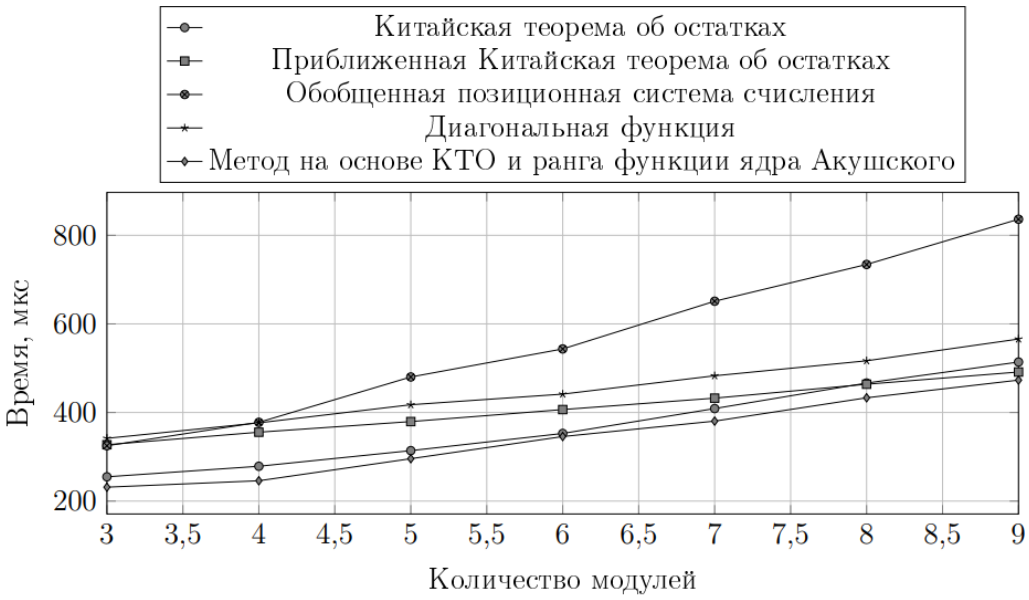

Рис 5. Сравнение времени для методов обратного преобразования из СОК в ПСС, второй этап

Проведено моделирование итерационного деления в СОК. Модификация алгоритма итерационного деления с использованием Теоремы 4 в среднем быстрее классического итерационного деления на 12.2% и быстрее итерационного деления с

Теоремой 3 на 6.2%. Однако, использование Теоремы 4 возможно только при использовании нечетных модулей в базисе СОК.

Проведено моделирование алгоритмов определения знака и сравнения чисел. Результаты моделирования представлены на рисунках 6-7.

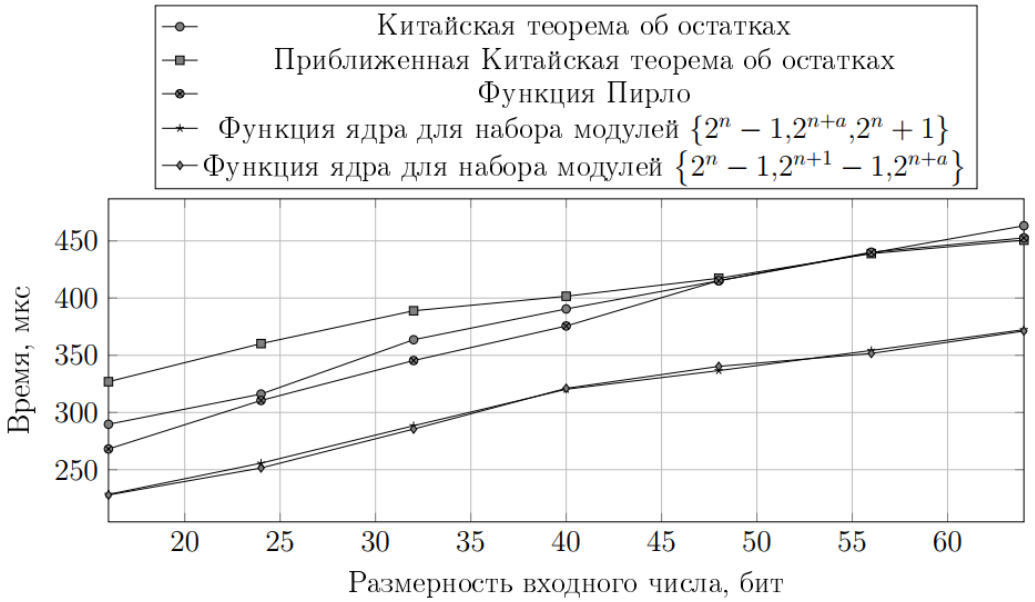

Рис 6. Сравнение времени для методов определения знака числа в СОК

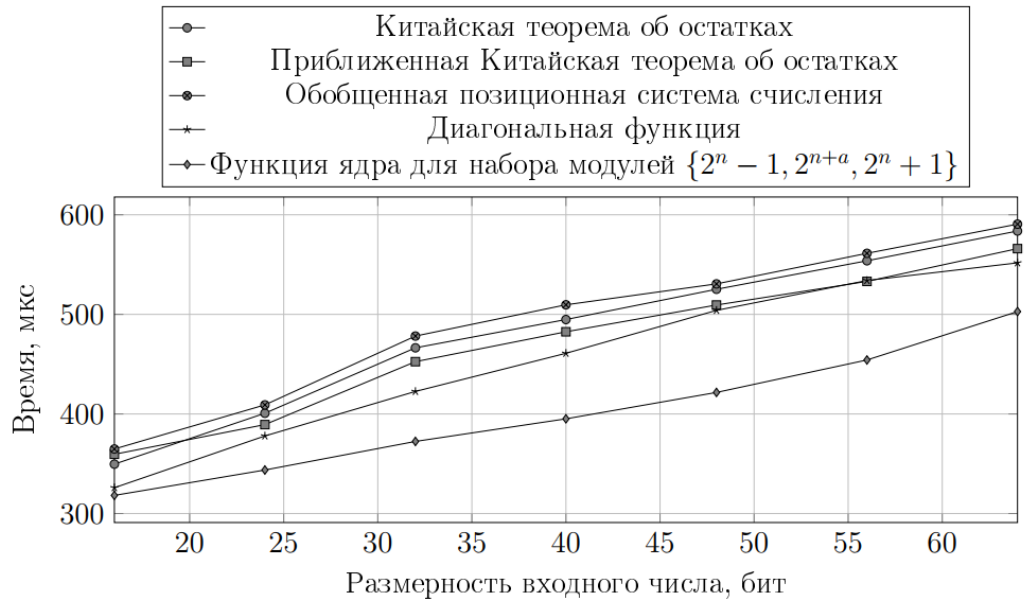

Рис 7. Сравнение времени для методов сравнения чисел в СОК

Алгоритм на основе использования минимальной функции ядра для набора модулей $\{2^n-1,2^{n+1}-1,2^{n+a}\}$ в среднем на 20% быстрее КТО, на 23.4% быстрее приближенной КТО и на 17.6% быстрее функции Пирло в операции определения знака числа.

Разработанный метод построения функций ядра Акушского для операции сравнения чисел в СОК позволил увеличить скорость сравнения в среднем на 15.1% для набора модулей $\{2^n-1,2^{n+a},2^n+1\}$ по сравнению с классическими методами.

Проведено моделирование операций модулярного сложения и умножения в СОК с использованием языка программирования C++ с использованием OpenMP для параллельной обработки разрядов. Для сравнения производительности выбраны библиотеки NTL и MIRACL, работающие в двоичной системе.

Для моделирования и сравнения был взят диапазон от 128 до 1024 бит. В системе остаточных классов наборы модулей выбирались так, чтобы для динамического диапазона P выполнялось условие $\lfloor \log_2 P \rfloor = k$ бит. Наборы построены с помощью Алгоритма 5. Данные о наборах модулей представлены в таблице 2.

Таблица 2 – Параметры наборов модулей системы остаточных классов, использованных для моделирования модуля арифметических операций

Размер набора модулей, n	Длина динамического диапазона, бит
8	128
12	256
16	512
20	768
32	1024

Диапазон случайных чисел для входных данных ограничивался в зависимости от типа операции, чтобы обеспечить соответствие динамическому диапазону СОК и избежать переполнения. Для каждого динамического диапазона и для каждой проверяемой операции было выполнено по 10^5 измерений. Результаты моделирования представлены в таблицах 3-4.

Таблица 3 – Результаты моделирования операции модулярного сложения, мкс

Длина динамического диапазона, бит	СОК	NTL	MIRACL
128	1904	2306	2453
256	2374	3431	4895
512	2449	4227	5963
768	2810	5109	7007
1024	3113	5910	8438

Таблица 4 – Результаты моделирования операции модулярного умножения, мкс

Длина динамического диапазона, бит	СОК	NTL	MIRACL
128	1967	2411	2629
256	2384	3646	5130
512	2571	4520	6678
768	2892	5311	7535
1024	3254	6337	9762

Реализация в СОК позволяет сократить время модулярного сложения на 36.5% по сравнению с NTL, и на 51.2% по сравнению с MIRACL. Модулярное умножение в СОК на 38.1% быстрее NTL и 53.7% быстрее MIRACL.

Основные результаты и выводы по работе

Проведенное исследование демонстрирует, что применение системы остаточных классов способно снизить вычислительную сложность алгоритмов обработки данных за счет параллельного выполнения арифметических операций сложения, вычитания и умножения. Однако при этом возникают дополнительные проблемы, связанные с оптимизацией таких процедур, как: обратное преобразование числа, общее деление, определение знака числа, сравнение чисел и др. Основные результаты исследования могут быть сформулированы следующим образом:

1. Разработан метод для перевода из СОК в ПСС на основе КТО и ранга числа функции ядра Акушского, который за счет ухода от операции нахождения остатка от деления позволяет сократить время вычислений по сравнению с Китайской теоремой об остатках и с приближенным методом на основе Китайской теоремой об остатках.
2. Разработаны модифицированные методы итерационного деления в СОК. Данные методы позволяют повысить скорость вычислений по сравнению с классическим итерационным делением за счет вычисления функции ядра на основе ее значения для предыдущего частного.
3. Разработаны алгоритмы определения знака на основе минимальной функции ядра Акушского с заданными свойствами для специальных наборов модулей. Эти алгоритмы позволяют за счет уменьшения размера операндов и снижения вычислительной сложности операции нахождения остатка от деления с $O(n^2)$ до $O(n)$ повысить скорость определения знака по сравнению с классическими методами.
4. Разработан метод построения функций ядра Акушского для операции сравнения чисел. Сравнение чисел в СОК на основе построенных минимальных функций ядра Акушского, у которых $C(P) = 2^N$, позволяет повысить скорость вычислений за счет снижения размера операндов, а также замены операции нахождения остатка от деления взятием N младших бит числа.
5. Разработан алгоритм построения компактных базисов системы остаточных классов. Наборы базисов, полученные данным алгоритмом, позволяют снизить время выполнения модульных операций в СОК по сравнению с использованием модулей специального вида за счет большей степени параллелизма и соответствия критерию компактности.
6. Разработаны алгоритмы поиска оптимальных весов функции ядра Акушского. Использование генетического алгоритма для решения задачи поиска оптимальных весов функции ядра позволило сократить время поиска на 71.2% по сравнению с методом Монте-Карло. Для задач с большим числом модулей в

системе остаточных классов целесообразнее использовать генетический алгоритм, который демонстрирует лучшее время выполнения, несмотря на возрастающую сложность задачи.

7. Разработан программный комплекс для выполнения модульных и немодульных операций в СОК на вычислительных узлах туманных сред. Комплекс показал высокую эффективность при выполнении указанных операций. Представленные вычислительные модули могут быть применены в облачных и туманных системах, работающих в условиях ограниченных вычислительных ресурсов. К направлениям внедрения результатов диссертационного исследования можно отнести криптографию и искусственные нейронные сети. Кроме того, разработанные методы, алгоритмы и программные средства могут быть применены в других областях, где требуется параллельная обработка больших объемов данных.

Публикации по теме диссертации

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

1. Поиск оптимальных весов для функции ядра Акушского / В.В. Луценко, Д.Е. Горлачев, Н.М. Мирный [и др.] // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. – 2025. – Т. 14, № 2. – С. 26-41.
2. Исследование специальных наборов модулей системы остаточных классов / В.В. Луценко, М.Д. Кравцов, Д.Е. Горлачев [и др.] // Труды Института системного программирования РАН. – 2025. – Т. 37, № 3. – С. 107-120.
3. Оптимизация алгоритма деления чисел в системе остаточных классов на основе функции ядра Акушского / В.В Луценко, М.Г. Бабенко, А.Н. Черных [и др.] // Труды Института системного программирования РАН. – 2023. – Т. 35, № 5. – С. 157-168.
4. Lutsenko, V. High speed method of conversion numbers from residue number system to positional notation / V. Lutsenko, M. Babenko, M. Khamidov // Proceedings of the Institute for System Programming of the RAS. – 2024. – Т. 36, № 4. – С. 117-132.
5. Creating distributed artificial neural networks based on orthogonal transformations / N. Vershkov, M. Babenko, V. Lutsenko [et al.] // Proceedings of the Institute for System Programming of the RAS. – 2024. – Т. 36, № 4. – С. 57-68.
6. Lutsenko, V. Construction of Akushsky Core Functions Without Critical Cores / V. Lutsenko, M. Babenko, M. Deryabin // Mathematics. – 2024. – Т. 12, № 21. – С. 3399.
7. Algorithm for Determining the Optimal Weights for the Akushsky Core Function with an Approximate Rank / E. Shiriaev, N. Kucherov, M. Babenko [et al.] // Applied Sciences. – 2023. – Т. 13, № 18. – С. 10495.

Другие публикации автора по теме диссертации

8. Fedina, A. Analytical Review of Classification and Clustering Methods of Cyber Attacks Based on Data Mining and Neural Network Approach / A. Fedina, V. Lutsenko, N. Gladkova // Conference on Current Problems of Applied Mathematics and Computer Systems. – Springer. 2023. – С. 285-294.
9. High-Speed Parity Number Detection Algorithm in RNS Based on Akushsky Core Function / V. Lutsenko, A. Geryugova, M. Babenko [et al.] // International Conference on Communication and Computational Technologies. – Springer. 2024. – С. 491-504.
10. Lutsenko, V. An efficient implementation of the Montgomery algorithm using the Akushsky core function / V. Lutsenko, E. Bezuglova // International Workshop on Advanced Information Security Management and Applications. – Springer. 2023. – С. 166-177.
11. Lutsenko, V. Comparative Analysis of Methods and Algorithms for Building a Digital Twin of a Smart City / V. Lutsenko, M. Babenko // International Conference on Actual Problems of Applied Mathematics and Computer Science. – Springer. 2022. – С. 277-287.
12. Lutsenko, V. Fault Tolerant System for Data Storage, Transmission and Processing in Fog Computing Using Artificial Neural Networks / V. Lutsenko, M. Zgonnikov // Conference on Current Problems of Applied Mathematics and Computer Systems. – Springer. 2023. – С. 199-212.
13. Shiriaev, E. An Approximate Algorithm for Determining the Sign Function of a Number Using Neural Network Methods / E. Shiriaev, V. Lutsenko, M. Babenko // International Workshop on Advanced Information Security Management and Applications. – Cham : Springer Nature Switzerland, 2023. – С. 247-255.
14. Lutsenko, V. Investigation of Neural Network Methods for Error Detection and Correction in the Residue Number System / V. Lutsenko, M. Zgonnikov // International Workshop on Advanced Information Security Management and Applications. – Springer. 2024. – С. 194-206.
15. Сравнительный анализ методов восстановления чисел, заданных на основе модульного представления / Е.В. Непретимова, П.А. Ляхов, А.В. Гладков [и др.] // Естественные науки-основа настоящего и фундамент для будущего. – 2019. – С. 70-73.
16. Луценко, В.В. Исследование нейросетевых методов обнаружения и исправления ошибок в системе остаточных классов / В.В. Луценко, М.Г. Бабенко // Всероссийская научно-практическая конференция им. Жореса Алфёрова: сборник тезисов статей. – 2023. – С. 248.

Свидетельства о государственной регистрации программы для ЭВМ

17. Свидетельство о государственной регистрации программы для ЭВМ 2023619967 Российская Федерация. Реализация вычисления знака числа в системе остаточных классов / Кучеров Н.Н., Ширяев Е.М., Безуглова Е.С., Луценко В.В., Гробова С.К.; заявитель и правообладатель Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». – № 2023617877; заявл. 26.04.2023; опубл. 17.05.2023. – 1 с.
18. Свидетельство о государственной регистрации программы для ЭВМ 2023662227 Российская Федерация. Реализация сравнения чисел в системе остаточных классов / Кучеров Н.Н., Ширяев Е.М., Безуглова Е.С., Луценко В.В., Колбин М.Д.; заявитель и правообладатель Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». – № 2023617795; заявл. 26.04.2023; опубл. 07.06.2023. – 1 с.
19. Свидетельство о государственной регистрации программы для ЭВМ 2024619754 Российская Федерация. Программный комплекс для определения критических ядер при построении функции ядра Акушского / Луценко В.В., Бабенко М.Г.; заявитель и правообладатель Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». – № 2024619005; заявл. 25.04.2024; опубл. 25.04.2024. – 1 с.
20. Свидетельство о государственной регистрации программы для ЭВМ 2024619970 Российская Федерация. Программа для определения четности числа в системе остаточных классов с использованием функции ядра Акушского / Луценко В.В., Бабенко М.Г., Герюгова А.Э.; заявитель и правообладатель Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». – № 2024619046; заявл. 25.04.2024; опубл. 02.05.2024. – 1 с.
21. Свидетельство о государственной регистрации программы для ЭВМ 2024660104 Российская Федерация. Программа для определения переполнения при сложении чисел в системе остаточных классов с использованием функции ядра Акушского / Луценко В.В., Бабенко М.Г.; заявитель и правообладатель Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». – № 2024619026; заявл. 25.04.2024; опубл. 02.05.2024. – 1 с.
22. Свидетельство о государственной регистрации программы для ЭВМ 2024660122 Российская Федерация. Программа для подбора оптимальных весов

для функции ядра Акушского с использованием генетического алгоритма / Луценко В.В., Бабенко М.Г., Безуглова Е.С., Ширяев Е.М; заявитель и правообладатель Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». – № 2024619092; заявл. 25.04.2024; опубл. 02.05.2024. – 1 с.

23. Свидетельство о государственной регистрации программы для ЭВМ 2024660861 Российская Федерация. Программа для реализации итерационного деления чисел в системе остаточных классов с использованием функции ядра Акушского / Луценко В.В., Бабенко М.Г., Безуглова Е.С., Ширяев Е.М; заявитель и правообладатель Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». – № 2024619025; заявл. 25.04.2024; опубл. 14.05.2024. – 1 с.
24. Свидетельство о государственной регистрации программы для ЭВМ 2024661039 Российская Федерация. Программа для реализации масштабирования чисел в системе остаточных классов с использованием функции ядра Акушского / Луценко В.В., Бабенко М.Г.; заявитель и правообладатель Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». – № 2024619051; заявл. 25.04.2024; опубл. 15.05.2024. – 1 с.