

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**РАСПАРАЛЛЕЛИВАНИЕ МЕТОДА ОПТИМИЗАЦИИ
ИМИТАЦИИ ОТЖИГА В ОПЕРАЦИОННОЙ СИСТЕМЕ QNX
С ПОМОЩЬЮ СЕТИ QNET**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Горшенина Артёма Александровича

Научный руководитель
ст. преподаватель

Н. Е. Тимофеева

Заведующий кафедрой
доцент, к. ф.-м. н.

Л. Б. Тяпаев

Саратов 2025

ВВЕДЕНИЕ

В современных вычислительных системах оптимизация играет ключевую роль в решении сложных задач из различных областей науки и техники. Одним из эффективных стохастических методов оптимизации является метод имитации отжига, который широко применяется в комбинаторных и непрерывных задачах оптимизации. Однако, несмотря на свою гибкость и способность находить приближённые решения глобальных экстремумов, данный метод характеризуется высокой вычислительной сложностью, что ограничивает его применение в задачах с большими объёмами данных.

Для повышения скорости выполнения алгоритма имитации отжига возможно его распараллеливание, что позволяет задействовать ресурсы многопоточных и многопроцессорных систем. Развитие параллельных вычислений и распределённых систем делает возможным существенное ускорение процессов оптимизации, что особенно актуально для задач, требующих значительных вычислительных мощностей. Важным направлением в данном контексте является использование операционных систем реального времени, которые обеспечивают детерминированное выполнение задач и эффективное управление вычислительными ресурсами.

Современные встраиваемые системы предъявляют высокие требования к надёжности, предсказуемости и эффективности, что делает операционные системы реального времени (RTOS) неотъемлемой частью их функционирования. Благодаря своей микроядерной архитектуре, эффективному управлению ресурсами и поддержке многопроцессорности, они широко применяются в критически важных сферах, таких как автомобилестроение, медицинское оборудование и промышленная автоматизация, обеспечивая минимальные задержки и высокую отказоустойчивость.

С ростом значимости встраиваемых систем требования к их производительности и надёжности становятся все более жесткими. В этом контексте особый интерес представляет использование RTOS для организации кластерных вычислений, которые позволяют распределять нагрузку между несколькими узлами системы. В условиях увеличения объемов данных и потребности в высокопроизводительных real-time вычислениях исследование возможностей в этой области приобретает особую актуальность.

Практическая значимость работы заключается в повышении эффек-

тивности вычислительных процессов при решении оптимизационных задач. Реализация параллельного варианта метода имитации отжига может найти применение в таких областях, как логистика, машинное обучение, обработка изображений и научные вычисления. Использование распараллеливания позволяет добиться высокой степени надёжности и предсказуемости выполнения вычислительных процессов, а также эффективно организовывать вычисления в распределённых системах.

Целью данной работы является разработка и реализация параллельной версии алгоритма имитации отжига с использованием инструментов распараллеливания и операционной системы реального времени QNX для решения задач оптимизации в условиях высоких требований к производительности и детерминированности выполнения.

Для достижения цели были поставлены следующие задачи:

- Изучить принципы построения параллельных вычислительных систем.
- Описать принципы работы параллельного алгоритма и разработать его блок-схему.
- Освоить необходимое программное обеспечение для распараллеливания алгоритма.
- Настроить работу с операционной системой QNX по сети QNET с использованием комплекта разработчика QNX Momentics на языке программирования C++.

Методологические основы распараллеливания представлены в работах В.П Гергель, Р.Г. Стронгин, К. В. Корняков, И.Б. Мееров, А.А. Сиднев, А.В. Сысоев, А.В. Шишков, Д. Алексеев, Е. Видревич, А. Волков, С.Н. Зыль, А. В. Высоцкий, А. С. Тараканов, К. И. Шоломов, Н. Е. Тимофеева, А. А. Ерофтиев.

Бакалаврская работа состоит из введения, теоретических и практических разделов, заключения, списка использованных источников и А, Б приложений. Общий объем работы – 80 страниц, из них 68 страниц – основное содержание, 5 страниц - приложения, включая 33 рисунка, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первом разделе "Постановка задачи распараллеливания" отмечено, что использование параллельных вычислительных систем является ключевым направлением развития современной вычислительной техники. Это обусловлено не только ограничениями скорости традиционных последовательных ЭВМ, но и постоянным усложнением вычислительных задач, требующих всё большей производительности. Многие актуальные научные и инженерные проблемы, такие как моделирование климатических процессов, геномная инженерия и разработка лекарственных препаратов, требуют вычислительных мощностей, превышающих 1 терафлопс. Параллельные вычисления становятся зачастую единственным возможным решением для выполнения ресурсоёмких задач. Параллельное выполнение операций осуществляется за счёт многопроцессорности, обеспечивающей избыточность функциональных устройств. Ускорение возможно при декомпозиции алгоритма на независимые компоненты и их распределении между процессорами. Эффективность определяется степенью распараллеливания и числом вычислительных узлов, а предельное ускорение ограничивается количеством независимых фрагментов и архитектурой системы.

Для эффективного параллельного выполнения достаточно выделения крупных независимых фрагментов расчётов, что снижает сложность разработки методов параллельной обработки. Такой подход также минимизирует межузловой обмен данными в кластерных системах, повышая эффективность распределённых вычислений.

Метод имитации отжига — эффективный алгоритм поиска глобального минимума, но его последовательная реализация требует значительных затрат, делая его неэффективным для задач с большими объёмами данных.

Основная цель распараллеливания — повышение скорости алгоритма за счёт одновременного выполнения независимых процессов. Это сокращает время поиска оптимального решения и делает метод применимым в системах реального времени (RTOS).

Решение этой задачи позволит разработать эффективный параллельный вариант метода имитации отжига для критически важных систем, таких как управление транспортом, медицинские устройства и промышленные автоматизированные комплексы.

Во втором разделе "Принципы построения параллельных вычислительных систем" содержится информация, о принципах, которые легли в основу реализации параллельного алгоритма. Эффективность ПВС ограничивается законом Амдала, который определяет максимальное ускорение вычислений в зависимости от доли последовательных операций. Например, если 10% алгоритма не поддаётся распараллеливанию, максимальное ускорение не превысит 10 раз, независимо от количества процессоров.

Для достижения параллелизма необходимо обеспечить:

1. Независимость аппаратных компонентов — минимизацию блокировок между процессорами, устройствами ввода-вывода и памятью.
2. Избыточность ресурсов — дублирование вычислительных элементов, таких как специализированные процессоры или многоуровневая память.
3. Конвейерную обработку — одновременное выполнение различных стадий вычислений для разных данных.

Важную роль играет топология коммуникации между узлами, которая влияет на производительность системы. В работе используется тип топологии "Звезда" — централизованное управление через главный узел. Для обмена данными используются метод передачи сообщений (МПС) — отправка полного объёма данных за один раз.

Выбор метода зависил от требований к производительности, задержкам и надёжности. В работе также подчёркивается важность микроядерной архитектуры (как в QNX) для эффективного управления ресурсами и поддержки распределённых вычислений через сети типа QNET. Эти принципы легли в основу реализации параллельного алгоритма имитации отжига, что позволило достичь значительного ускорения вычислений.

В третьем разделе "Кластерные вычисления" посвящён кластерным вычислениям и многопроцессорным системам с распределённой памятью. Рассматриваются два основных класса:

- Симметричная многопроцессорная обработка (СМО) — высокопроизводительные системы с сотнями или тысячами процессоров, работающих параллельно. Каждый процессор имеет локальную память, а обмен данными происходит через высокоскоростную сеть.
- Кластерные системы — совокупность вычислительных узлов, объединённых сетью и управляемых специализированным ПО.

Для эффективного использования многопроцессорных архитектур вычислительный процесс должен обладать высокой степенью параллелизма. Подходящие задачи включают однотипные вычисления с минимальной итерационной зависимостью, например восстановление зашифрованного текста перебором ключей, моделирование нестационарных процессов.

Существует две основные модели распределённой обработки:

- Сильносвязанные системы (SMP) — процессоры используют общую память и ресурсы ввода-вывода, параллелизация на уровне потоков.
- Слабосвязанные системы (кластеры) — узлы независимы, взаимодействуют через каналы передачи данных, параллелизация на уровне процессов.

SMP-системы ориентированы на производительность, а кластеры дополнительно обеспечивают отказоустойчивость: при выходе из строя до N-1 узлов система остаётся работоспособной. Остальные архитектуры комбинируют элементы этих моделей.

В четвёртом разделе "Используемое программное обеспечение" представлены инструменты, применённые для реализации параллельной версии алгоритма имитации отжига. В работе использовались:

1. Операционная система QNX Neutrino - это операционная система реального времени, основанная на микроядерной архитектуре. Её ключевые особенности включают:
 - Высокую надежность и отказоустойчивость, что критично для встраиваемых систем.
 - Поддержку многопроцессорных и распределённых вычислений.
 - Механизм обмена сообщениями, обеспечивающий эффективное взаимодействие между процессами.
2. Среда разработки QNX Momentics предоставляет инструменты для создания, отладки и профилирования приложений под QNX Neutrino. Основные возможности:
 - Поддержка языков C, C++ и Java.
 - Удалённая отладка и мониторинг процессов.
 - Инструменты для анализа производительности и использования памяти.
3. Сеть QNET используется для организации кластерных вычислений в

среде QNX. Она позволяет:

- Объединять несколько узлов в единую вычислительную систему.
 - Обеспечивать прозрачный доступ к ресурсам удалённых узлов.
 - Минимизировать задержки при передаче данных.
4. Реализация алгоритма выполнена на C++ с использованием стандартной библиотеки и возможностей многопоточности. Для распараллеливания применены технологии MPI.
 5. Библиотека ExrgTK - использовалась для парсинга и вычисления математических выражений, что позволило сделать алгоритм гибким и применимым к различным функциям.

Выбор данного ПО обусловлен требованиями к производительности, надёжности и совместимости в задачах реального времени.

В пятом разделе "Алгоритм имитации отжига" представлено описание метода, вдохновлённого процессом отжига в металлургии, где материал нагревается и медленно охлаждается для достижения стабильной структуры. В контексте оптимизации этот метод используется для поиска глобального минимума функции, даже при наличии множества локальных минимумов. Основные этапы алгоритма:

1. Задание начального состояния x_0 : случайным образом выбирается начальная точка $x = x_0$, $x_0 \in \Omega$. Текущее значение энергии E устанавливается в значение $f(x_0)$.
2. Установка начальной температуры $T(k)$ (закон изменения температуры), где k - номер шага.
3. Задание параметров охлаждения, таких как коэффициент охлаждения a и критерия остановки.
4. Функция принятия нового состояния $h(\Delta E, T)$
5. Порождающие семейство вероятностных распределений $\zeta(x, T)$. Выбирается как семейство нормальных распределений с математическим ожиданием x и дисперсией T и соответственно задаётся плотностью:

$$g(x'; x, T) = (2\pi T)^{-n/2} \cdot \exp(-|x' - x|^2 / (2T)),$$

где n — размерность метрического пространства состояний.

Повторение следующих шагов, пока не будет достигнут критерий остановки, при k -ой итерации алгоритма:

1. Генерация нового состояния x' вблизи текущего состояния x с использованием некоторой функции изменения (например, малое случайное изменение текущего состояния).
2. Оценка нового состояния: сравнить энергию системы E в состоянии x с найденным на настоящий момент глобальным минимумом, вычислив изменения стоимости $\Delta E = f(x') - f(x)$. Если $E < \min$, $E = f(x)$, то нужно изменить значение глобального минимума;
3. Принятие решения о переходе:
 - Если $\Delta E \leq 0$, то новое состояние x' принимается.
 - Если $\Delta E > 0$, то новое состояние x' принимается с вероятностью

$$P = \exp(-\Delta E/T), \quad (3)$$

где T — текущая температура.

Текущее состояние системы заменяется новым состоянием, созданным на предыдущем шаге, то есть если новое состояние x' принято, то $x \leftarrow x'$, а именно необходимо сгенерировать новую точку $x' = G(x, T(k))$ и вычислить значение функции в ней $E' = f(x')$. Потом сгенерировать случайное число a из интервала $[0;1]$.

Если $a < h(E' - E, T(k))$, то установить $x = x'$, $E = E'$, и перейти к следующей итерации. Иначе, повторять генерацию пока не будет найдена подходящая точка x' .

Изначально метод имитации отжига использовали по схеме, известной как больцмановский отжиг. В этой схеме изменение температуры задаётся формулой:

$$T(k) = \frac{T_0}{\ln(1 + k)}, k > 0. \quad (4)$$

Для данной схемы доказано, что при достаточно больших T_0 и количестве шагов k гарантируется нахождение глобального минимума.

Недостатком схемы больцмановского отжига является очень медленное уменьшение температуры T . Решение этой проблемы возможно путём замены закона изменения температуры, например, на следующий $T(k) = a \cdot T(k - 1)$, где температурный коэффициент a выбирается, как правило, в пределах от 0.7 до 0.99.

$$T \leftarrow aT,$$

где $a < 1$ — коэффициент охлаждения, после каждого k -го шага.

Алгоритм завершает работу, когда достигается заданное условие остановки, например, температура становится ниже порогового значения или количество итераций превышает максимально допустимое. После завершения цикла алгоритм выводит минимальное значение функции, а также оптимальные значения x и y , при которых оно достигается.

Метод имитации отжига использует упорядоченный случайный поиск. Благодаря этому алгоритм способен избегать ”ловушек” в локальных минимумах оптимизируемой функции и принимать временные ухудшения результата, что помогает ему находить глобальный минимум многоэкстремальной целевой функции даже при ограниченных ресурсах, предоставляя при этом приемлемые локальные минимумы.

В шестом разделе ”Практическая реализация” описывается разработка и реализация алгоритма имитации отжига на языке C++ с последующим его распараллеливанием с использованием технологии QNET в операционной системе QNX. Основные этапы включают:

1. Разработка алгоритма.

- Реализация последовательной версии алгоритма имитации отжига для оптимизации функции Розенброка.
- Использование библиотеки ExprTK для парсинга математических выражений.
- Включение механизмов для предотвращения переполнения и оптимизации работы с памятью.

2. Распараллеливание.

- Организация распределённых вычислений между главным узлом (Master) и вычислительными узлами (Workers) через сеть QNET.
- Применение MPI (Message Passing Interface) для обмена сообщениями между узлами.
- Решение проблем синхронизации генераторов случайных чисел и дублирования данных.

3. Оптимизация.

- Использование разделяемой памяти для уменьшения дублирова-

ния данных.

- Динамическое распределение задач между узлами для балансировки нагрузки.

4. Результаты.

- Демонстрация работоспособности алгоритма на примере функции Розенброка.
- Сравнение времени выполнения и использования памяти для последовательной и параллельной версий.

Также анализируются результаты тестирования алгоритма на различных тестовых функциях (Аккли, Химмельблау, Eggholder) и оценивается выигрыш от распараллеливания:

1. Метрики эффективности:

- Ускорение вычислений: Параллельная версия показала ускорение в 1.4–3.47 раза по сравнению с последовательной.
- Точность: Параллельный алгоритм находит решения, близкие к глобальному минимуму, с меньшей погрешностью.
- Использование памяти: Оптимизация за счёт разделяемой памяти и минимизации дублирования данных.

2. Результаты тестирования:

- Функция Аккли: Улучшение точности в 4 раза, ускорение в 2.75 раза.
- Функция Химмельблау: Обнаружение нескольких глобальных минимумов, линейное ускорение ($3x$).
- Функция Eggholder: Параллельная версия почти достигла глобального минимума, тогда как последовательная версия застряла в локальных минимумах.

Использование параллельных вычислений особенно эффективно при работе со сложными функциями, имеющими множество локальных минимумов. Количество задействованных узлов подбирается в зависимости от задачи и доступных ресурсов. Такой подход не только ускоряет процесс, но и позволяет находить более качественные решения за счёт параллельного изучения разных зон пространства параметров.

Эти результаты подтверждают практическую значимость предложенного подхода для задач глобальной оптимизации в распределённых системах.

ЗАКЛЮЧЕНИЕ

В рамках данной дипломной работы была успешно решена задача распараллеливания алгоритма имитации отжига с использованием технологий QNX и QNET. Проведённое исследование позволило разработать эффективный подход к распределению вычислительной нагрузки между несколькими узлами, что особенно актуально для задач, требующих значительных вычислительных ресурсов.

Основное внимание в работе было уделено изучению принципов построения параллельных вычислительных систем и их применению для стохастических алгоритмов оптимизации. Была разработана архитектура решения, сочетающая централизованное управление с инструментальной машины и распределённое выполнение вычислений на узлах QNX. Особое значение имела проработка механизмов взаимодействия между компонентами системы, обеспечивающая минимальные накладные расходы.

Важным результатом работы стало освоение комплекса программных технологий, включая систему разработки QNX Momentics, библиотеки для параллельных вычислений и средства межпроцессного взаимодействия. Практическая реализация продемонстрировала работоспособность предложенного подхода и его преимущества по сравнению с последовательным выполнением алгоритма.

Проведённое исследование открывает перспективы для дальнейшего развития проекта. В частности, представляет интерес адаптация решения для работы в гетерогенных вычислительных средах и интеграция с современными системами распределённых вычислений. Разработанные методы и подходы могут быть применены для решения широкого круга задач оптимизации в различных предметных областях.

Результаты работы имеют как теоретическое, так и практическое значение, демонстрируя эффективность использования распределённых вычислений для алгоритмов глобальной оптимизации. Полученный опыт может быть полезен при создании подобных систем в промышленных и исследовательских проектах.

Основные источники информации:

- 1 Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. Учебное пособие - Нижний Новгород: Изд-во ННГУ им. Н.И Лобачевского, 2003.
- 2 Практика работы с QNX/Д. Алексеев, Е. Видревич, А. Волков и др.- М.: КомБук, 2004.
- 3 Зыль С. Н. QNX Momentics: основы применения. - СПб.: БХВ-Петербург, 2005.
- 4 А. В. Высоцкий, А. С. Тараканов, К. И. Шоломов, Н. Е. Тимофеева, А. А. Ерофтиев // Исследование эффективности поиска глобального экстремума методом имитации отжига, распараллеленного различными способами // Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2013. Т.13, вып. 3
- 5 Корняков К. В., Мееров И. Б., Сиднев А. А., Сысоев А. В., Шишков А. В. Инструменты параллельного программирования в системах с общей памятью. - Учебное пособие / Под ред. проф. В.П. Гергеля. - Нижний Новгород: Изд-во Нижегородского госуниверситета, 2010.
- 6 Алгоритм моделирования отжига в оптимизации: [Электронный ресурс] URL: <https://trituenhantaio.io/kien-thuc/giai-thuat-simulated-annealing-trong-toi-uu-hoa/> (15.04.2025)