**Государственное бюджетное общеобразовательное учреждение города Москвы «Школа № 1532»**

**Метод распределенных вычислений на языке программирования python**

11 класс, ГБОУ школа № 1532,

Воронов Никита Рустамович

Руководитель:

учитель информатики, ГБОУ школа №1532,

Сергиенко Антон Борисович

**Москва, 2024**

### **Содержание**

[1. Введение 3](#_m2n120clopop)

[2. Формирование целей и задач проекта 3](#_xv0ulx27su5t)

[2.1 Определение цели проекта 3](#_nik809zc1p1u)

[2.2 Определение задач проекта 4](#_6xawi1hyldt8)

[2.3 Методы выполнения задач 4](#_ab9z15cz3zrf)

[3. Техническая реализация 5](#_ipgv8isdmm3c)

[3.1 Используемые компоненты 6](#_uwjblomb9rm3)

[3.2 Работа с окружением 6](#_ot2u0zauj32i)

[4. Функции представленного метода 7](#_4oy6x5cdaxac)

[4.1 Функциональные особенности 7](#_yg0frgud2cf9)

[4.2 Список параметров 7](#_43sxk1dpob3n)

[5. Тестирование и развертывание проекта 7](#_bywsrq4jhxjx)

[6. Итоги и результаты 7](#_pavr4rwjk8r1)

[7. Список используемой литературы 7](#_8wbj4yn32tge)

# 

# 

# **1. Введение**

В современном мире, окутанном пеленой цифровых возможностей, когда в наших карманах обитают устройства, столь мощные, что даже тени первых компьютеров, отправивших человека в космос, блекнут в их свете, проявляется чрезвычайная динамика потребностей. Неудовлетворенность, ощущаемая современными пользователями, становится своеобразным катализатором, подталкивающим к мысли, что идеала не достичь.

Способность современных устройств лишь задевает поверхность амбиций человека в области вычислений. Хочется не просто владеть технологией, но и влиять на нее, управлять ей, вносить свой вклад в тот стремительный поток инноваций, который нам суждено встречать каждый день. Именно в таком контексте метод распределенных вычислений вступает в игру, предоставляя ответ на эти амбиции и желание превзойти собственные ожидания.

Этот подход к вычислениям – не просто следующий шаг в эволюции технологий, а целый эпос, в котором вычислительные ресурсы не просто объединяются, но создают симфонию мощи и ума. В мире распределенных вычислений каждый вычислительный узел – не просто элемент сети, но активный участник в глобальном процессе созидания и преобразования. Таким образом, среди этого водоворота технологических изменений, человек не просто наблюдает, но и становится творцом своего собственного вычислительного будущего.

# **2. Формирование целей и задач проекта**

## **2.1 Определение цели проекта**

Цель данного исследовательского проекта заключается в разработке инновационного метода распределенных вычислений, основанного на мощностях и потенциале современных устройств. Задача работы сводится к созданию эффективной системы, способной использовать повседневные аппараты в качестве ключевого элемента для управления и участия в вычислительных процессах.

Использование языка в качестве управляющего инструмента направлено на то, чтобы сделать вычисления более доступными везде.

Таким образом, работа направлена не только на технологические усовершенствования, но и на переосмысление способов взаимодействия человека с вычислительными ресурсами. Проект нацелен на создание инновационного метода, который открывает новые перспективы в области распределенных вычислений, делая их более гибкими и удобными.

## **2.2 Определение задач проекта**

В качестве задач для проекта были выделены следующие пункты:

1. Разработка и внедрение эффективной системы распределенных вычислений, ориентированной на взаимодействие с повседневными устройствами.
2. Исследование и определение оптимальных методов интеграции повседневных устройств для управления и мониторинга вычислительных ресурсов.
3. Создание механизмов для автоматизированного управления и мониторинга вычислительных процессов с использованием повседневных устройств.
4. Оптимизация алгоритмов обработки данных, передаваемых от повседневных устройств, с целью повышения производительности и эффективности вычислительных процессов.
5. Исследование методов обеспечения безопасности в контексте использования повседневных устройств для управления распределенными вычислениями.
6. Разработка пользовательских интерфейсов, удобных для взаимодействия с повседневными устройствами, с целью упрощения и улучшения управления вычислительными ресурсами.
7. Тестирование и апробация разработанных решений в реальных условиях для оценки их эффективности и пригодности в повседневном использовании.

## **2.3 Методы реализации проекта**

В ходе выполнения проекта по разработке системы распределенных вычислений, ориентированной на взаимодействие с повседневными устройствами, используются различные методы и технологии для достижения поставленных задач. Для разработки программного обеспечения используется язык программирования Python, который обеспечивает гибкость и высокую производительность.

Для обеспечения взаимодействия между устройствами и сервером используется технология сокетов (sockets), что позволяет эффективно передавать данные между клиентами и сервером. Для оптимизации обработки запросов и данных применяется многопоточность, что способствует параллельному выполнению задач и повышению общей производительности системы.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 1. Принцип ожидания подключения

В контексте технологий обмена данными можно выделить работу на уровне транспортного уровня OSI. Использование этого уровня обеспечивает надежную передачу данных между устройствами в сети, управляет потоком информации, а также обеспечивает контроль ошибок и восстановление данных при необходимости. Таким образом, выбор транспортного уровня OSI влияет на эффективность обмена данными в системе распределенных вычислений, обеспечивая стабильность и надежность передачи информации между устройствами и сервером.

# **3. Техническая реализация**

В основе предложенного метода распределения вычислений лежит идея построения сети с использованием встроенных инструментов Python. Эта сеть предназначена для непосредственной работы с распределенной инфраструктурой, а не с локальными вычислительными ресурсами. Ключевыми элементами этой концепции являются два типа файлов: серверные и структурные файлы библиотеки.

Структурные файлы библиотеки, согласно данному подходу, представляют собой компоненты, которые могут быть напрямую импортированы в исходный код программы на Python. Этот процесс обеспечивает непосредственное подключение вашей программы к распределенной сети, что, в свою очередь, способствует более эффективному и гибкому управлению вычислительными ресурсами.

Таким образом, предлагаемый метод представляет собой интегрированный подход, позволяющий легко взаимодействовать с распределенной сетью, делая акцент на использовании инструментов Python и эффективном импорте библиотек напрямую в программный код. Это открывает новые возможности для эффективного управления ресурсами и интеграции в распределенные вычислительные среды.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 2. Пример функции приема данных от клиента

## **3.1 Используемые компоненты**

Разработанный метод распределения вычислений строится на эффективном взаимодействии с встроенными инструментами языка программирования Python. Используя threading, socket и subprocess, создается устойчивая архитектура, способная эффективно управлять вычислительными задачами в распределенной сети.

Threading обеспечивает возможность параллельного выполнения задач, способствуя оптимизации процесса обработки данных. Socket используется для установления и поддержания соединений между различными узлами сети, обеспечивая эффективное общение и передачу данных. Subprocess позволяет запускать дополнительные процессы, расширяя возможности взаимодействия с внешними ресурсами и библиотеками.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 3. Пример открытия нового потока на сервере для обработки текущегзапроса

В контексте предлагаемой системы, серверные и структурные файлы библиотеки обрабатываются с использованием threading для распараллеливания вычислений. Socket обеспечивает связь между клиентскими и серверными узлами, а subprocess позволяет эффективное взаимодействие с внешними приложениями и ресурсами.

Таким образом, методика применения threading, socket и subprocess в сочетании с использованием структурных файлов библиотеки предоставляет мощный инструментарий для построения эффективных распределенных систем. Этот подход обеспечивает высокую производительность, гибкость и управляемость вычислительными ресурсами в распределенной среде.

## **3.2 Работа с окружением**

Распределенные системы предоставляют ряд значимых преимуществ при работе с окружением, обеспечивая эффективное управление ресурсами и повышение производительности. Во-первых, такие системы позволяют распределить вычислительные задачи между несколькими узлами, что способствует параллельному выполнению задач и сокращению времени обработки данных.

Дополнительно, распределенные системы обеспечивают гибкость и масштабируемость, позволяя адаптироваться к изменяющимся требованиям окружения. Это особенно важно в современных динамичных условиях, где необходимость в высокой отзывчивости и устойчивости к нагрузкам становится все более актуальной. Именно по этой причине разработка на python является наиболее эффективной, так как данный язык программирования способен интерпретироваться на любом устройстве.

Одним из ключевых преимуществ является возможность резервного копирования и обеспечения отказоустойчивости. В случае сбоя на одном из узлов другие узлы могут продолжить выполнение задач, минимизируя временные задержки и предотвращая потерю данных. За данный параметр отвечает сервер, который может использовать различные структуры RAID массивов для сохранности данных.

Благодаря возможности взаимодействия между различными устройствами и серверами, распределенные системы способствуют интеграции разнородных технологий и устройств, что расширяет функциональные возможности и обеспечивает более широкий спектр приложений. В целом, распределенные системы содействуют эффективной организации работы с окружением, предоставляя инструментарий для эффективного управления ресурсами, повышения отказоустойчивости и обеспечения гибкости в адаптации к меняющимся требованиям современного информационного ландшафта.

## **3.3 Распределение задачи**

В качестве первой задачи, которую необходимо рассчитать была взята задача нахождения наиболее выгодного маршрута в графе маршрутов. Или, как еще ее называют решение задачи коммивояжера.

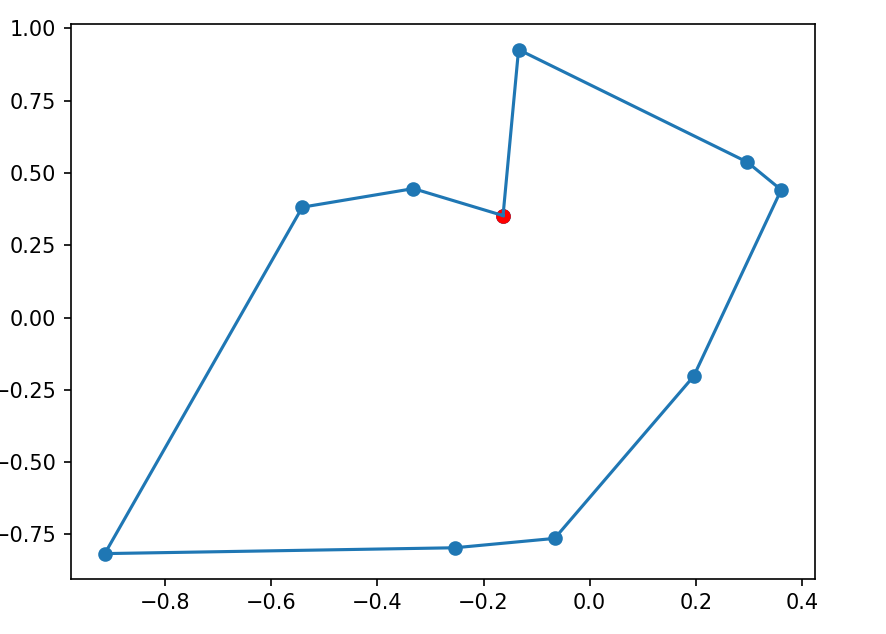


Рисунок 3. Пример построения маршрута между случайными точками.

Стоит понимать, что задача коммивояжера — это задача на оптимизацию, а не на полный перебор, но для данного случая решение полным перебором может сыграть на руку. После запуска, можно заметить, что задачи не были распределены рисунок 2. Это связано с тем, что в задачах, где циклическая составляющая идет последовательно, одна за другой, использование многопоточности через модуль threading немного некорректен.

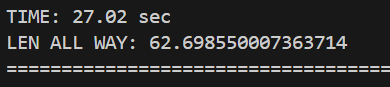


Рисунок 4. Время выполнения алгоритма полного перебора

Чтобы решить данную проблему пришлось использовать связывание потоков, чтобы регулировать приводимые данные в массив.

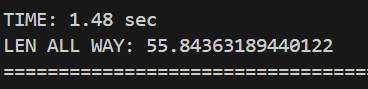


Рисунок 5. Время выполнения алгоритма полного перебора с библиотекой распределения вычислений

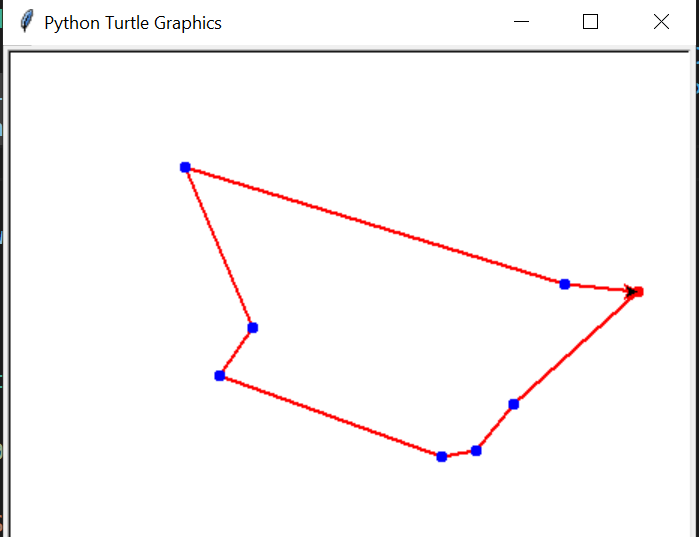


Рисунок 6. Ускоренное решение задачи

# **5. Тестирование и развертывание проекта**

В ходе проведения эксперимента по оптимизации вычислительных процессов в рамках моего проекта по распределенным вычислениям на платформе Python, был взят в качестве тестовой задачи классический вопрос коммивояжера. Применен был стандартный алгоритм полного перебора параметров, который в последствии был успешно распределен на 5 вычислительных узлах, входящих в локальную сеть.

Результаты эксперимента оказались весьма впечатляющими: время выполнения задачи практически мгновенно сократилось по сравнению с последовательным выполнением на одном устройстве. Неоспоримо демонстрируется преимущество параллельной обработки данных в данном контексте.

Замечательным фактом является то, что с увеличением объема переборов задачи, алгоритм продолжал демонстрировать высокую эффективность, превосходя время выполнения, которое было бы затрачено на одном вычислительном устройстве. Это говорит о высокой степени масштабируемости разработанной системы, что является важным достижением в контексте распределенных вычислений.

В итоге, проведенный эксперимент подтверждает эффективность реализованного распределенного подхода к вычислениям на основе Python, открывая перспективы для более сложных и объемных задач, которые могут быть успешно оптимизированы и обработаны в кратчайшие сроки.

# **6. Итоги и результаты**

В результате исследования по разработке инновационного метода распределенных вычислений выявлены перспективы создания универсальной системы, способной взаимодействовать с повседневными устройствами, что делает вычисления более доступными и гибкими. Проект ориентирован не только на технологические улучшения, но и на переосмысление взаимодействия человека с вычислительными ресурсами.

В результате исследования видится возможность создания инновационного метода, оптимизирующего вычислительные процессы и переопределяющую роль повседневных устройств. Этот метод повышает эффективность обработки данных и производительность вычислений, открывая новые возможности для взаимодействия с технологиями.

Ключевыми элементами успеха проекта являются эффективные механизмы управления и мониторинга, интегрированные с повседневными устройствами, а также разработка безопасных методов обеспечения целостности и конфиденциальности данных в процессе распределенных вычислений.

Таким образом, исследовательский проект не только направлен на создание технологически продвинутой системы, но и предлагает новый взгляд на взаимодействие человека с вычислительной средой, открывая широкие перспективы в области распределенных вычислений и трансформируя их в более удобные и интуитивно понятные процессы.

# **7. Список используемой литературы**

1. Г. Куц, А. Стрейт/Й. Шнейдер – текст – учебное пособие – Москва 2007г. – 130 стр. - "Распределенные системы. Архитектура и программирование" – URL: http://sp.cmc.msu.ru/courses/sdpi/mdwrbook.pdf (дата обращения: 12.11.2023)
2. А. Романовский, М. Парамонов – текст – официальное пособие – Москва 2023 - "Многозадачные системы. Подходы и технологии" – URL: https://www.rgups.ru/site/assets/files/78119/programma\_po\_osnovam\_informatcionnykh\_sistem\_2024.pdf (дата обращения 20.11.2023)
3. С. Лифшиц, В. Костров - текст – учебное пособие – СПБ 2014 – 155 стр. - "Параллельные и распределенные вычисления: модели и методы" - URL: https://books.ifmo.ru/file/pdf/1551.pdf (дата обращения 13.12.2023)
4. В. А. Балашов - текст – материалы курса – Москва 2013 – 49 стр. - "Распределенные базы данных: технологии и реализация" - URL: https://www.galaktika-dmk.com/upload/iblock/39a/978\_5\_97060\_391\_8.pdf (дата обращения 02.01.2024)
5. А. Таненбаум, М. Ван Стин - текст - учебно-методическое пособие – Москва 2021 – 40 стр. - "Распределенные системы. Принципы и парадигмы" - URL: https://msupress.com/ebook/978-5-19-011913-8\_e-book.pdf (дата обращения 12.12.2023)
6. В.Н. Садовский - сайт – информационный ресурс - "Многозадачные вычисления в распределенных системах" - URL: https://studylib.ru/doc/2675761/sistema-raspredelennyh-vychislenij-v (дата обращения 12.01.2024)