МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

ФАЙЛОВОЕ ХРАНИЛИЩЕ НА BLOCKCHAIN С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ IPFS

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Жидкова Виктора Петровича

Научный руководитель
доцент, к. ф.-м. н. _____ В. М. Соловьёв

Заведующий кафедрой
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2023

Введение

В современном информационном обществе, хранение и обработка данных являются одними из самых важных задач. Традиционные централизованные системы хранения данных сталкиваются с рядом проблем, таких как уязвимость к атакам, централизация контроля и ограниченная прозрачность. В связи с этим, возникает потребность в разработке новых подходов и технологий для обеспечения безопасности, надежности и доступности данных.

Целью данной дипломной работы является разработка файлового хранилища на основе технологии Blockchain с использованием протокола InterPlanetary File System (IPFS). Это позволит реализовать децентрализованное хранение файлов с высокой степенью надежности, безопасности и доступности данных.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить основы технологий Blockchain и IPFS.
2. Проанализировать возможности и ограничения совместного использования этих технологий.
3. Спроектировать и разработать децентрализованное файловое хранилище на основе Blockchain и IPFS.
4. Провести тестирование и оценку разработанной системы.
5. Проанализировать потенциал применения разработанного решения в различных отраслях и сценариях использования.

Объектом исследования является процесс хранения и обработки данных в децентрализованных системах. Предметом исследования являются технологии Blockchain и IPFS, а также разработка и оценка децентрализованного файлового хранилища на их основе.

В данной работе используются следующие методы исследования: изучение научной литературы, анализ существующих решений и технологий, проектирование и разработка программного обеспечения, тестирование и оценка системы, а также сравнительный анализ с аналогами.

Структура дипломной работы состоит из пяти основных разделов. Во втором разделе рассматриваются теоретические основы исследования, включая обзор технологий Blockchain и IPFS. Третий раздел посвящен проектированию и разработке файлового хранилища. В четвертом разделе представлены результаты исследования, практическое применение разработанного файлово-

го хранилища и сравнение с существующими решениями на рынке. Заключение содержит выводы по дипломной работе, рекомендации для дальнейшего развития проекта и возможности применения результатов исследования в других сферах.

В ходе выполнения работы ожидается получение новых знаний о возможностях и ограничениях совместного использования технологий Blockchain и IPFS, а также разработка децентрализованного файлового хранилища с высоким уровнем безопасности и доступности данных. Это может стать основой для развития новых приложений и сервисов, использующих преимущества децентрализации и распределенного хранения данных.

Основное содержание работы

Структура бакалаврской работы

Работа состоит введения, теоретической и практической частей, заключения и приложения.

— В теоретической части описывается предметная область, основные понятия в теме файловых хранилищ. Далее обозревается технология IPFS, ее преимущества, недостатки и перспективы развития. После обозревается технология блокчейн, основные понятия из этой сферы, ее будущее и взаимодействие с протоколом IPFS.
— В практической части обозревается проектирование и разработка файлового хранилища.
— В приложении представлен исходный код программы

Файловые хранилища представляют собой системы, предназначенные для организации и управления хранением файлов и данных. Они играют важную роль в современных информационных системах, обеспечивая доступность, целостность и безопасность файлов.

Одним из основных принципов файловых хранилищ является структурированное хранение данных. Файлы и документы организуются в иерархические или плоские структуры, позволяя легко ориентироваться и находить нужные файлы. Различные метаданные, такие как название, тип, дата создания и модификации, обычно привязываются к каждому файлу для дальнейшей идентификации и управления.

Традиционные файловые хранилища основываются на централизованной архитектуре, где файлы хранятся на серверах и доступ к ним осуществляется посредством сетевых протоколов. Это позволяет централизованно управлять доступом, обеспечивать резервное копирование данных и применять политики безопасности. Однако, такие системы часто сталкиваются с проблемами масштабируемости, уязвимостью к отказам и сингл-поинтам отказа.

С развитием технологий появились новые подходы к файловым хранилищам, такие как децентрализованные хранилища. Они основаны на использовании распределенных сетей и протоколов, которые позволяют файлам храниться и быть доступными на различных узлах сети. Это обеспечивает повышенную отказоустойчивость, устойчивость к цензуре и возможность обмена файлами без привязки к центральным серверам.

Вместе с тем, файловые хранилища находят применение в различных сферах и отраслях. Они используются в облачных сервисах для хранения и синхронизации файлов между устройствами, в системах электронного документооборота для управления и архивации документов, в медицинских информационных системах для хранения медицинских данных и многое другое.

Таким образом, файловые хранилища играют важную роль в организации и управлении файлами и данных. Современные тенденции направлены на развитие децентрализованных решений, таких как хранилища на основе технологии Blockchain и протокола IPFS, которые обеспечивают более безопасное, надежное и эффективное хранение файлов в условиях современного информационного общества.

IPFS (InterPlanetary File System) - это протокол и распределенная система хранения и обмена файлами. Он был разработан с целью создания более эффективной и децентрализованной альтернативы традиционным клиент-серверным моделям передачи данных в сети Интернет.

Основной идеей IPFS является замена идентификации файлов на основе их местоположения (по IP-адресу) на идентификацию файлов на основе их содержимого. Вместо того, чтобы ссылаться на файлы по их физическому расположению, IPFS использует уникальные хэши содержимого файлов в качестве их адресов. Это позволяет обеспечить уникальность и целостность файлов, так как любые изменения в содержимом файла приведут к изменению его хэша.

IPFS использует распределенную сеть узлов, называемых пирами, для хранения и обмена файлами. Когда файл добавляется в IPFS, он разбивается на блоки данных, каждый из которых получает уникальный хэш. Затем эти блоки распределяются по сети IPFS, и каждый блок адресуется по его хэшу. Если в сети уже есть узел, который имеет данный блок, то он может быть получен напрямую от этого узла, что позволяет улучшить скорость и эффективность передачи данных.

IPFS также поддерживает кэширование данных, что позволяет узлам хранить и предоставлять доступ к наиболее популярным и часто запрашиваемым файлам. Это снижает нагрузку на сеть и повышает скорость доставки файлов.

Одной из ключевых особенностей IPFS является возможность верификации содержимого файлов. Так как каждый файл адресуется по его хэшу, то можно убедиться в том, что полученный файл точно соответствует ожидаемому

содержимому. Это обеспечивает доверие и защиту от подделки файлов.

IPFS предоставляет удобный интерфейс командной строки и API для управления файлами и доступа к данным. Он также поддерживает функции распределенного хэширования, децентрализованной идентификации и шифрования данных.

IPFS активно развивается и используется в различных сферах, включая хранение и обмен файлами, распределенные приложения, хостинг контента и резервное копирование данных. Его открытая архитектура и принципы децентрализации делают его привлекательным инструментом для создания новых парадигм обмена данными в сети Интернет.

Blockchain - это децентрализованная и распределенная технология, предназначенная для создания безопасных и прозрачных систем записи и передачи данных. Основной принцип Blockchain заключается в создании цепочки блоков, каждый из которых содержит набор транзакций или данных, а также уникальный идентификатор (хэш) предыдущего блока.

Основные принципы Blockchain:

- Децентрализация: Blockchain не имеет единой центральной точки управления. Вместо этого, данные и транзакции хранятся и проверяются множеством участников сети, известных как узлы.
- Распределенный реестр: Все транзакции и данные записываются в цепочку блоков, которая хранится на каждом узле в сети. Это создает распределенный реестр, который можно проверять и подтверждать участниками сети.
- Криптографическая безопасность: Каждый блок в цепочке связан с предыдущим блоком при помощи криптографических хэшей, обеспечивая целостность данных. Кроме того, транзакции подписываются криптографическими ключами, чтобы обеспечить аутентификацию и невозможность подделки.
- Консенсус: Участники сети должны достичь согласия по состоянию и правильности блоков и транзакций. Это достигается через механизмы консенсуса, которые определяют, какие блоки будут добавлены в цепочку и какие транзакции будут считаться действительными.

Архитектура Blockchain:

- Блоки: Блоки представляют собой единицы информации, которые содер-

жат набор транзакций или данных. Каждый блок имеет уникальный идентификатор (хэш) и ссылку на предыдущий блок в цепочке.

— Цепочка блоков: Блоки связаны между собой, создавая цепочку блоков. Это позволяет обеспечить непрерывность и целостность данных, поскольку изменение блока потребует изменения всех последующих блоков.
— Узлы: Узлы представляют участников сети, которые хранят и проверяют блоки и транзакции. Узлы могут быть разными - от полных узлов, которые хранят полную копию блокчейна, до легких узлов, которые хранят только часть блокчейна.
— Криптографические ключи: Криптографические ключи используются для подписи транзакций и обеспечения безопасности и аутентификации в сети. Публичные ключи используются для проверки подписи, а приватные ключи используются для создания подписи.
— Механизмы консенсуса: Механизмы консенсуса определяют, какие блоки будут добавлены в цепочку и какое состояние считается правильным. Примеры механизмов консенсуса включают Proof-of-Work (доказательство работы) и Proof-of-Stake (доказательство доли).

Вместе эти принципы и архитектура создают надежную и безопасную среду для записи и передачи данных в Blockchain.

Существует несколько типов блокчейн-сетей, каждая из которых имеет свои особенности и применения.

— Публичные блокчейн-сети (Public Blockchains): Это открытые сети, доступные для участия любому желающему. Все участники имеют равные права и могут участвовать в создании блоков, проверке транзакций и голосовании по изменениям протокола. Примером публичной блокчейн-сети является биткоин (Bitcoin).
— Приватные блокчейн-сети (Private Blockchains): Это закрытые сети, доступные только для определенных участников. Управление доступом к сети и принятием решений осуществляется центральным органом или ограниченным числом участников. Приватные блокчейн-сети обеспечивают повышенную конфиденциальность данных и контроль над сетью. Они часто используются в корпоративных средах или для специфических целей, таких как снабжение и логистика.
— Консорциальные блокчейн-сети (Consortium Blockchains): Это блокчейн-

сети, управляемые группой организаций или участников, которые сотрудничают и договариваются о правилах сети. Участники имеют предварительно определенные права и ответственности, и совместно поддерживают блокчейн-инфраструктуру. Консорциальные блокчейн-сети обычно более эффективны и масштабируемы, чем публичные сети, при сохранении некоторой степени децентрализации. Примером консорциальной блокчейн-сети является Hyperledger Fabric.

— Гибридные блокчейн-сети (Hybrid Blockchains): Это комбинация публичных и приватных блокчейн-сетей. Гибридные сети позволяют участникам выбирать уровень доступности и конфиденциальности, а также взаимодействовать с другими участниками в сети. Например, можно использовать публичную сеть для выполнения публичных транзакций и приватную сеть для более конфиденциальных операций.

Смарт-контракты являются программами, разработанными для автоматического выполнения и управления соглашениями и условиями, записанными в блокчейн-сети. Они работают на основе принципа «если-то» и позволяют двум или более сторонам взаимодействовать и выполнять сделки без необходимости доверять друг другу.

Основное преимущество смарт-контрактов состоит в том, что они обеспечивают автоматическое и надежное выполнение условий без необходимости привлечения посредников или центральных органов. Благодаря прозрачности и невозможности изменения записей в блокчейне, смарт-контракты обеспечивают высокую степень доверия между участниками сделки.

Децентрализованные приложения (DApps) строятся на базе блокчейн-технологии и смарт-контрактов. Они предлагают новую парадигму разработки и использования приложений, где вся логика и данные хранятся на блокчейне, а участники сети могут взаимодействовать друг с другом напрямую без посредников.

Децентрализованные приложения имеют ряд преимуществ. Во-первых, они обладают высокой степенью прозрачности и безопасности благодаря использованию блокчейна. Во-вторых, они позволяют пользователям сохранять контроль над своими данными и личной информацией, так как они не хранятся в централизованном сервере. В-третьих, децентрализованные приложения открыты для участия и вклада разработчиков и пользователей со всего мира, что

способствует инновациям и развитию.

Примеры децентрализованных приложений включают децентрализованные финансовые платформы (DeFi), децентрализованные биржи, онлайн-игры, системы голосования и многое другое. Эти приложения стремятся изменить способ взаимодействия, управления и экономических отношений, делая их более открытыми, прозрачными и справедливыми.

textbfВзаимодействие между IPFS и блокчейном может быть весьма полезным и эффективным, особенно в контексте создания децентрализованных приложений и систем хранения данных.

Одним из основных способов интеграции IPFS и блокчейна является использование хэшей IPFS в блокчейн-транзакциях. Вместо того, чтобы хранить сам файл в блокчейне, можно хранить только его хэш, который является адресом файла в IPFS. Это позволяет значительно сократить размер блокчейна и улучшить его масштабируемость. При необходимости получить доступ к файлу, можно использовать его хэш, чтобы загрузить его из сети IPFS.

Другой вариант взаимодействия состоит в использовании смарт-контрактов для управления доступом к файлам, хранящимся в IPFS. Смарт-контракты на блокчейне могут содержать логику управления правами доступа к файлам, таким образом, обеспечивая контроль и безопасность. Например, можно создать смарт-контракт, который разрешает доступ к определенному файлу только определенным пользователям или условиям.

IPFS и блокчейн также могут использоваться в совместных проектах для создания децентрализованных приложений (DApps). IPFS может служить для хранения и обмена файлами, в то время как блокчейн может обеспечивать управление правами доступа, аудит и целостность данных. Это открывает новые возможности для создания приложений, которые не зависят от централизованных серверов и предлагают более безопасное и прозрачное взаимодействие с данными.

Также стоит отметить, что IPFS и блокчейн дополняют друг друга в аспекте децентрализации. IPFS обеспечивает децентрализованное хранение и распределение файлов, в то время как блокчейн обеспечивает децентрализованную проверку и подтверждение транзакций и данных. Это позволяет создавать более надежные и устойчивые системы, которые не зависят от единой точки отказа или централизованного управления.

В целом, взаимодействие IPFS и блокчейна предоставляет мощный инструментарий для создания децентрализованных систем хранения данных и приложений. Оно объединяет преимущества распределенного хранения и управления данными, что открывает новые возможности для повышения безопасности, прозрачности и эффективности в различных областях, включая финансы, снабжение, здравоохранение и государственные услуги.

Для разработки системы управления файлами на основе блокчейн технологий, необходимо было определить архитектуру и выбрать подходящие инструменты и технологии.

Был выбран протокол Ethereum, так как он предлагает широкие возможности для создания умных контрактов и децентрализованных приложений (DApps). Ethereum обладает большим сообществом разработчиков, обширной документацией и активной поддержкой, что способствует успешной разработке проекта.

Для написания смарт-контрактов на платформе Ethereum использовался язык программирования Solidity. Это наиболее популярный и хорошо документированный язык для разработки умных контрактов на Ethereum [19]. В свою очередь, для разработки клиентской части приложения и взаимодействия с блокчейн, был использован язык Typescript. Он обеспечивает строгую типизацию и совместимость с JavaScript, что облегчает разработку и поддержку кода.

Для разработки и тестирования смарт-контрактов был выбран Hardhat - современный инструмент для работы с Solidity, который облегчает компиляцию, тестирование и развертывание контрактов. Для взаимодействия с Ethereum блокчейном и умными контрактами в клиентской части приложения была выбрана библиотека web3js, которая предоставляет простой и понятный интерфейс.

В смарт-контракте реализована структура для хранения информации о каждом файле, которая содержит 5 полей: название, размер строка CID, ссылка на файл и его расширение. Структура в языке Solidity инициализируется с помощью ключевого слова «struct». Связь пользователя и файлов реализована с помощью Mapping. Mapping в Solidity - это структура данных, которая отображает ключи на значения. Он аналогичен словарю или хеш-таблице в других языках программирования. Ключом будет являться адрес кошелька пользователя, а значение массив объектов типа File

Добавление файла. Аргументами функции являются CID строка файла,

название, размер, ссылка и расширение файла. Внутри функции происходит создание объекта типа File и добавление его в массив значений конкретного пользователя в структуре Mapping, делается это через ключевое слово msg.sender, которое содержит в себе адрес кошелька пользователя вызвавшего этот метод.

Удаление файла Аргументом функция принимает только CID строку файла, далее идет проверка, является ли пользователь владельцем данного файла и, собственно, удаление файла из Mapping. Проверка реализована через использование хеш-функции keccak256, в которую передается строка переведенная в байты. Функция keccak256 принимает на вход произвольные данные и вычисляет на их основе уникальный 256-битный хеш, который может быть использован для проверки целостности данных и для создания уникальных идентификаторов.

Получение всех файлов Функция не принимает никаких аргументов, так как получение значения из Mapping возможно с помощью ключевого слова msg.sender.

Заключение

В данной работе была исследована актуальная проблема создания надежного и эффективного файлового хранилища на основе блокчейн-технологии с использованием IPFS. Рассмотрены основные аспекты и преимущества комбинированного использования этих технологий, включая децентрализацию, безопасность, сокращение времени доступа к данным, устойчивость к цензуре и распределение нагрузки.

В ходе работы был проведен анализ существующих решений в области файловых хранилищ на блокчейн и их возможных применений. Определены основные требования и ограничения, которые должны учитываться при разработке таких систем. Был предложен подход к разработке и реализации файлового хранилища на базе блокчейн с использованием IPFS, а также определены основные этапы и компоненты системы.

Таким образом, в рамках данной работы были продемонстрированы возможности и перспективы использования комбинации блокчейн-технологии и IPFS в качестве альтернативного решения для организации файлового хранилища. В дальнейшем, это исследование может стать основой для разработки новых сценариев применения данных технологий, а также для улучшения и оптимизации существующих решений.

Возможными направлениями дальнейшего исследования являются углубленное изучение и оптимизация механизмов шифрования, интеграция с другими платформами и технологиями, разработка удобных и безопасных механизмов аутентификации пользователей, а также рассмотрение экономических аспектов применения разработанной системы.