

СОЗДАНИЕ КЛАССА ОБРАБОТКИ БЛОКЧЕЙН-ТРАНЗАКЦИЙ ДЛЯ ПЕРЕДАЧИ ДАННЫХ В СЕТИ УМНЫХ СЧЕТЧИКОВ НА PYTHON

Демин А.Ю.^{1,2}, Дмитрийчук Д.И.³, Зарубин В.А.⁴

¹НИ ТПУ, ИШИТР, ОИТ, ad@tpu.ru

²НИ ТГУ, ИПМКН, каф. ТОИ, ad@tpu.ru

³НИ ТПУ, ИШИТР, 8K11, did7@tpu.ru

⁴НИ ТПУ, ИШИТР, 8K13, vaz30@tpu.ru

Введение

С развитием технологий умные счётчики стали неотъемлемой частью современных энергетических систем, позволяя точно отслеживать и контролировать потребление электроэнергии. Эти устройства активно внедряются для повышения эффективности и снижения затрат, предоставляя данные для централизованной обработки. Однако при их использовании важнейшими аспектами остаются безопасность и прозрачность передаваемой информации, что критично для предотвращения мошенничества и ошибок учёта.

Одновременно с этим блокчейн-технологии начали активно распространяться за пределы криптовалют и нашли широкое применение в таких областях, как энергетика, логистика и здравоохранение. В энергетике блокчейн используют для совершенствования систем учёта и контроля за потреблением электроэнергии с помощью умных счётчиков. Децентрализованная структура блокчейна обеспечивает неизменность и сохранность записей, высокий уровень безопасности, а также возможность автоматизации процессов без участия посредников.

В последние годы был проведён ряд исследований, посвящённых интеграции блокчейна с умными счётчиками [1]. Эти работы показали, что блокчейн может значительно улучшить управление данными, защитив их от манипуляций и обеспечив прозрачность операций [2]. Ознакомившись с этими статьями, мы разработали алгоритм, который реализует данную концепцию, позволяя эффективно использовать блокчейн для управления данными в сети умных счётчиков.

В данной работе предлагается реализация этой концепции с использованием языка программирования Python и асинхронного веб-фреймворка FastAPI. Асинхронная архитектура системы позволяет обрабатывать множество параллельных запросов, повышая масштабируемость и эффективность системы. Это решение позволяет взаимодействовать с большим количеством умных счётчиков, обеспечивая бесперебойную обработку данных и выполнение блокчейн-транзакций в реальном времени.

Целью работы является создание системы, где каждый счётчик функционирует в качестве узла сети, а информация о потреблённой электроэнергии фиксируется в блокчейне, обеспечивая полную прозрачность и безопасность расчётов за электроэнергию.

Концепция использования

Блокчейн-архитектура для умных счётчиков, реализованная на Python с использованием асинхронного FastAPI, создаёт децентрализованную сеть, в которой каждый счётчик выполняет роль узла. Эти устройства собирают данные о потреблении электроэнергии и записывают их в блокчейн, что гарантирует точность, своевременность и защиту от манипуляций с информацией. Благодаря этому, данные становятся неизменяемыми и доступными всем участникам сети, что делает систему идеальной для энергетических сетей, где важны прозрачность и безопасность.

Асинхронная архитектура значительно повышает производительность системы, обрабатывая запросы и майнинг блоков параллельно. Каждый блок содержит информацию о потреблении за определённый период. В основе системы лежат механизмы консенсуса, такие как Proof of Stake (PoS) или Proof of Authority (PoA), которые обеспечивают безопасность и устойчивость: в PoS транзакции подтверждают узлы с наибольшими стейками, а в PoA — авторизованные участники, что упрощает управление сетью и снижает нагрузку.

Для обеспечения безопасности данные подписываются приватными криптографическими ключами каждого счётчика и передаются по защищённым каналам связи (например, с использованием TLS). Асимметричное шифрование, при котором данные шифруются публичным ключом сети, обеспечивает дополнительную защиту от перехвата.

Асинхронный подход FastAPI позволяет системе обрабатывать множество параллельных запросов, ускоряя запись данных в блокчейн. В случае отказа узла система автоматически пересчитывает

блоки и проверяет их целостность с помощью хешей. Дублирование данных на нескольких узлах повышает отказоустойчивость, а резервное копирование на внешние сервера или в облачные хранилища обеспечивает дополнительную надёжность.

В конце каждого расчётного периода (например, каждый месяц) блокчейн фиксирует данные о потреблении и автоматически подсчитывает итоговую стоимость электроэнергии для пользователей на основе умных контрактов. Счета отправляются автоматически, а данные о расходах сохраняются в блокчейне, что обеспечивает прозрачность и исключает ошибки.

Таким образом, предложенная блокчейн-архитектура, основанная на асинхронной обработке данных и современных механизмах безопасности, гарантирует высокую производительность, неизменность данных и автоматизацию расчётов.

Описание алгоритма

Основным элементом системы является класс Block, который содержит идентификатор счётчика, показания потребления, proof и ссылку на предыдущий блок через его хеш. Все блоки хранятся в цепочке, что обеспечивает целостность данных.

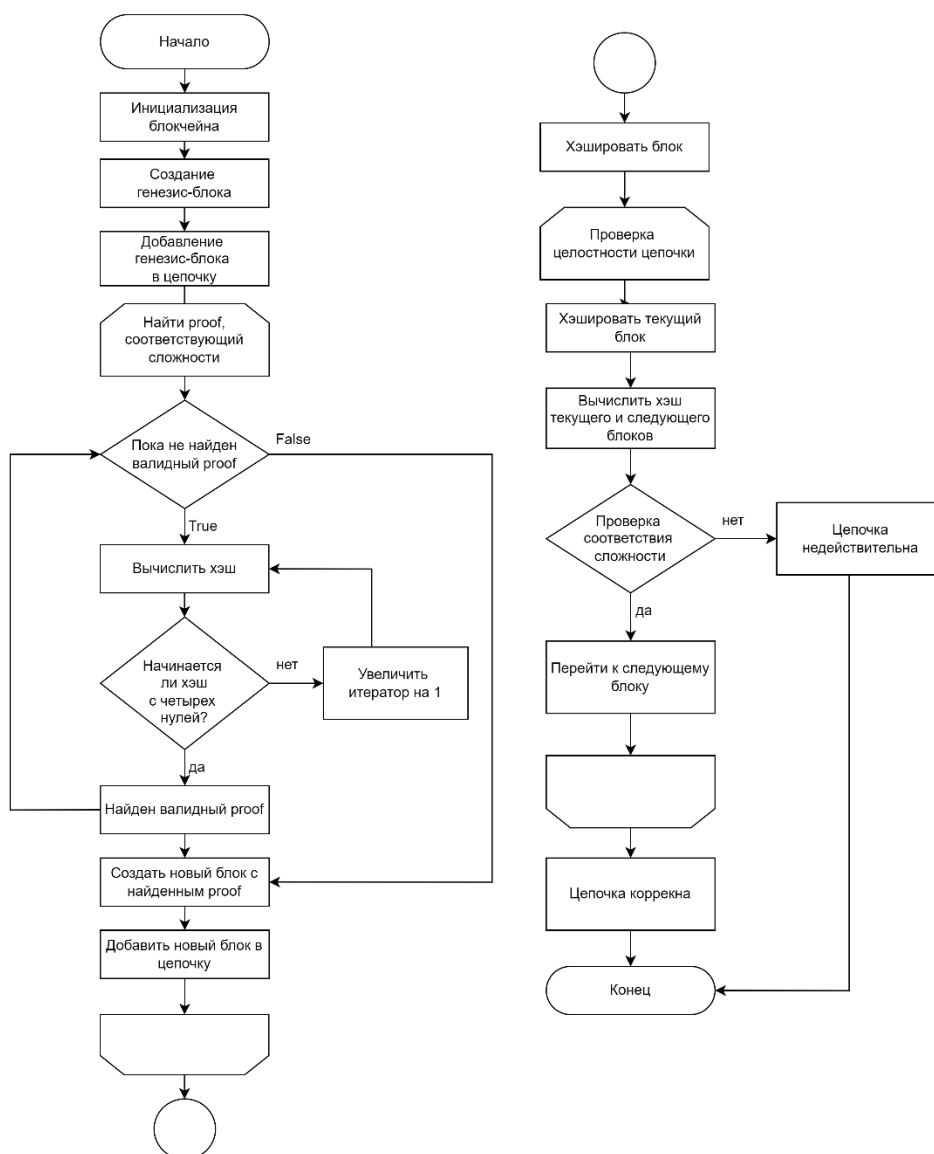


Рис. 1. Блок-схема работы алгоритма

Алгоритм основан на асинхронных функциях, что повышает производительность и масштабируемость. Блоки майнятся с помощью алгоритма Proof of Work (PoW), который требует нахождения proof, удовлетворяющего заданной сложности. В процессе майнинга используется асинхронная операция, которая не блокирует основную программу, повышая эффективность системы.

Каждый блок хешируется алгоритмом SHA-256 для проверки целостности данных. Асинхронный метод `is_chain_valid` проверяет корректность цепочки блоков, сравнивая хеши и `proof`, что предотвращает нарушения.

Кроме того, FastAPI предоставляет API конечные точки (эндпоинты) для взаимодействия с блокчейном: запросы цепочки блоков, проверка её целостности, майнинг новых блоков и получение истории потребления. Вся обработка данных, включая майнинг и валидацию, выполняется асинхронно, что делает систему подходящей для работы с большими объёмами данных в реальном времени.

Таким образом, этот алгоритм эффективно сочетает блокчейн-технологии и асинхронную обработку данных для создания надёжной и безопасной системы учёта электроэнергии.

Тестирование системы

Тестирование системы блокчейна для умных счётчиков было проведено с использованием фреймворка `pytest` для обеспечения корректной работы всех аспектов блокчейн-алгоритма, включая создание блоков, их валидацию и взаимодействие через API. Для проверки асинхронного кода применялось расширение `pytest.mark.asyncio`, что позволило тестировать процессы майнинга и валидации блоков, построенные на асинхронных операциях.

Одним из ключевых аспектов тестирования стало создание и проверка генезис-блока, который инициализирует цепочку. Важно было убедиться, что первый блок содержит корректные значения, такие как индекс, идентификатор счётчика, нулевое потребление и отсутствие предыдущего хеша. Следующие тесты проверяли корректность майнинга новых блоков, а также соблюдение требований доказательства работы (Proof of Work), где каждый новый блок должен иметь уникальное доказательство по сравнению с предыдущим.

Проверка целостности цепочки блоков проводилась с использованием метода `is_chain_valid`, который гарантировал, что каждый блок ссылается на правильный предыдущий блок. Для оценки устойчивости системы к ошибкам также был разработан тест, который умышленно искажал хеш предыдущего блока, после чего система корректно отмечала цепочку как недействительную.

Тестирование API системы также было критически важным. Были проведены тесты на создание нового блока через POST-запрос к соответствующей конечной точке. Система возвращала успешный ответ с данными о блоке при корректном выполнении запроса. Другие тесты включали запрос всей цепочки блоков и проверку валидации через соответствующие GET-запросы, что обеспечивало возможность внешней проверки состояния блокчейна.

Важным аспектом тестирования стало покрытие кода. Большинство ключевых модулей достигли уровня покрытия более 90%, что указывает на высокую степень проверяемости системы. Это свидетельствует о надёжности системы и её готовности к эксплуатации в условиях реальных данных и внешних запросов.

Результаты

В результате был создан блокчейн на Python и API, по которому он будет принимать данные от счётчиков. Умные счетчики передают свои показания через специальную конечную точку, где сервер обрабатывает эти данные и создает новый блок (Рис. 2). Каждый блок содержит уникальную информацию: идентификатор счетчика, количество потребленной электроэнергии и другие параметры. Процесс создания блока включает вычисление доказательства работы (`proof-of-work`), которое подтверждает корректность блока. Новый блок связывается с предыдущим через хеш, что гарантирует целостность и непрерывность цепочки блоков.

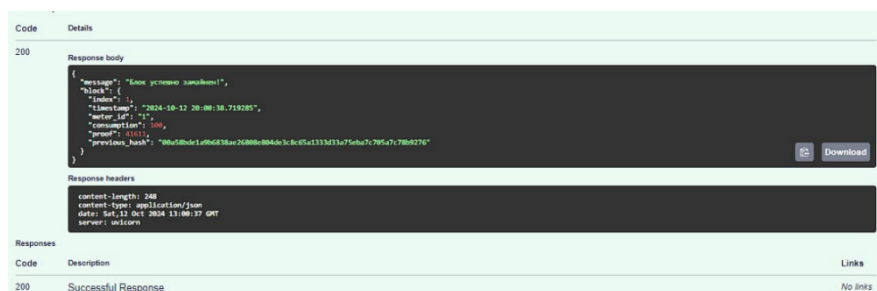


Рис. 2. Результаты работы при создании блока

После успешного создания блок добавляется в блокчейн, обновляя его структуру. Система предоставляет несколько конечных точек API, с помощью которых компании могут запросить полную цепочку блоков (Рис. 3), получить историю потребления электроэнергии по конкретному счетчику или проверить целостность блокчейна. Встроенная проверка цепочки гарантирует, что данные остаются неизменными и защищены от манипуляций. Таким образом, система обеспечивает надежную и прозрачную запись показаний электроэнергии, поддерживая неизменность и защищенность данных.

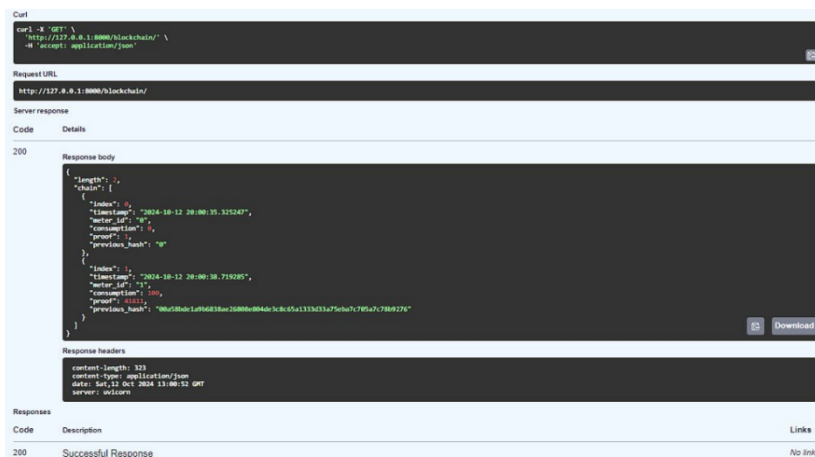


Рис. 3. Получение общей цепи блокчейна

Заключение

Разработанная система блокчейна для сбора данных от умных счетчиков электроэнергии обеспечивает надежную, защищенную и прозрачную платформу для обработки и хранения информации. Использование технологии блокчейна позволяет гарантировать неизменность данных, а механизм доказательства работы (proof-of-work) защищает систему от подделок и манипуляций. Благодаря асинхронной архитектуре FastAPI, система эффективно обрабатывает запросы, поддерживая высокую скорость работы и целостность цепочки блоков. Данная технология может значительно упростить и обезопасить процесс учета электроэнергии для электрокомпаний, обеспечивая достоверность и прозрачность учета потребления.

Список использованных источников

1. Белкин П.А., Посмаков Н.П., Ростовский Н.С. Применение технологии блокчейн в электроэнергетике как связующей цифровой технологии при переходе на децентрализованную генерацию // Национальный исследовательский ядерный университет «МИФИ».
2. Фазульянов Д.В. Проектирование информационной системы учета и оплаты услуг ЖКХ на основе технологии блокчейн // Электронный научный журнал «Вектор экономики», 2020. – № 6.
3. Как Python применяется в блокчейн. – Текст : электронный // Proglib.io. – URL: <https://proglib.io/p/kak-python-primenyaetsya-v-blokkcheyn-2021-03-19> (дата обращения: 29.09.2024).
4. Avgeros N. Building a simple blockchain with FastAPI: a beginner's guide. – Текст : электронный // Medium. – URL: <https://medium.com/@nik.avgeros/building-a-simple-blockchain-with-fastapi-a-beginners-guide-9024eff3369d> (дата обращения: 05.10.2024).
5. Как работает блокчейн: примеры на Python // Habr.com. – URL: <https://habr.com/ru/companies/ruvds/articles/589371/> (дата обращения: 29.09.2024).