Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

ИНДЕКСАЦИЯ ОБЪЕКТОВ БЛОКЧЕЙНА ETHEREUM В БАЗУ ДАННЫХ

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 273 группы
направления 01.04.02 — Прикладная математика и информатика
факультета КНиИТ
Белоусова Александра Александровича

Научный руководитель
Доцент, к.ф.-м.н. _____ В. М. Соловьев

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

Саратов 2018

ВВЕДЕНИЕ

Актуальность темы: в последнее время пользователей, использующих блокчейн технологии становится все больше. В число таких пользователей попадают, как обычные рядовые люди, так и крупные компании. В настоящее время используется порядка 1000 [1] различных блокчейн сетей. Наибольший интерес представляют те сети, которые имеют большую капитализацию. Так например первое место держит Bitcoin, капитализация, которого достигла 97 миллиардов долларов [1]. Второе место занимает Ethereum с капитализацией в 27 миллиардов [1] долларов. Не стоит забывать о таких качествах программного обеспечения, как эффективность и гибкости. Такие свойств присутствуют в контрактно ориентированном блокчейне Ethereum. Актуальность темы работы определяется тем, что в настоящее время в процессе использования технологии блокчейн актуальным является вопрос об удобстве взаимодействия с ними. Один из необходимых для работы с любым из блокчейнов средстовм является возможность быстро и эффективно просматривать историю операций, определять смарт-контракты и методы взаимодействия с ними.

Однако таких программных продуктов с открытым исходным кодом для Ethereum блокчейна просто нет, поэтому пользователям приходится полагаться на веб приложения сторонних компаний. В случае, если на основе данных, полученных из приложения, автоматизированной системе нужно принять решение, то это ставит под угрозу безопасность такой системы. **Основная цель работы** – разработка масштабируемого веб приложения с открытым исходным кодом, позволяющим эффективно индексировать операции и объекты блокчейна Ethereum.

Задачи, поставленные для достижения этой цели:

- изучение теории блокчейна и реализации протокола Ethereum;
- изучение типов объектов, из которых состоит Ethereum;
- изучение работы виртуальной машины EVM;
- изучение существующих стандартов смарт-контрактов и способов токенизации цифровых активов;
- изучить способы масштабирования веб-приложений;
- разработка смарт-контракт, реализующий интерфейсы ERC20/ERC223;
- разработка механизм развертывания смарт-контракт;
- рассмотреть результаты индексирования переводов токенов.

— разработка многопоточного, масштабируемого веб приложения для индексации в базу данных и для получения данных по протоколу HTTP.

Методологические основы принципы устройства блокчейна Ethereum George Icahn «Ethereum: The Complete Guide to Understanding Ethereum»[6], а также материал по разработке масштабируемых приложений от Vaughn Vernon «Reactive Messaging Patterns with the Actor Model»[18].

Практическая значимость магистерской работы. В ходе выполнения практической части магистерской работы было реализовано веб-приложение для индексации объектов блокчейна Ethereum в базу данных. Код приложения доступен по ссылке https://github.com/LykkeCity/EthereumSamurai.

Структура и объем магистерской работы. Работа состоит из введения, семи разделов, заключения, списка использованных источников информации и 5 приложений. Общий объем работы – 74 страниц, из них 51 страницы – основное содержание, включая 27 рисунков и 2 таблицы, цифровой носитель в качестве приложения, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Краткая характеристика задачи» посвящен составлению требования для разрабатываемого приложения, а также обзору существующих аналогов.

Требования к приложению — включает в себя сформулированные требования для разрабатываемого приложения.

Существующие аналоги включает в себя описание существующих приложений.

Второй раздел «Термины и экономическое применение» содержит в себе список терминов, применимых к блокчейну Ethereum, а также примеры использования в современном мире и краткую историю.

Экономическое применение — содержит краткое описание и примеры использования данного блокчейна на рынке.

История — краткое описание всех событий, связанных с разработкой и применением блокчейна Ethereum с 2013 года.

Словарь терминов — словарь терминов далее употребляющихся в работе, все термины связаны, или с блокчейном Ethereum, или с общими понятиями технологии блокчейн.

Третий раздел «Объекты блокчейна» содержит в себе описание объектов, из которых состоит блокчейн с техническими подробностями.

Аккаунт. Существует два типа аккаунтов:

— обычные или управляемые внешне;

— контракты;

Оба типа контрактов могут содержать эфир. Эфир - это основная криптовалюта блокчейна Ethereum. Майнеры получают ее, как вознаграждение за сборку новых блоков, а пользователи тратят эфир на комиссию для выполнения транзакций. Транзакции могут быть отправлены с обоих типов аккаунтов, хотя контракты отправляют транзакции, называемые внутренними сообщениями,только в ответ на другие транзакции или внутренние сообщения, которые они получили. Так что все операции начинаются с транзакций, отправленных с аккаунта управляемого внешне. Самая простая транзакция - это перевод средств с одного обычного аккаунта на другой [2].

Блок. Блок - это пакет данных, содержащий 0 или более транзакций, хеш предыдущего блока, и другие опциональные данные. Каждый блок, за исключением первоначального, ссылается на родительский, такая структура данных и называется блокчейн [3].

Транзакция. Ethereum блокчейн можно представить как основанный на транзакциях автомат, где транзакции меняют состояние автомата, а состояние содержит в себе информацию о взаимодействиях [3].

Событие.Особая форма хранения данных для асинхронного ответа на действие и для удешевления стоимости сохранения результата операции. [3].

Токены. Стандарты цифровых валют или ценных бумаг, представленные в виде смарт контракта с интерфейсом ERC20/223.

Четвертый раздел «Устройство памяти блокчейна» содержит в себе описание применения префиксных деревьев для надежного хранения данных в блокчейне.

Дерево Меркла — содержит информацию об основной структуре блокчейна Ethereum.

Доказательство Меркла в Эфириуме — посвящен объяснению использования хеш-деревьев для защиты данных блокчейна от подмены.

Префиксное дерево — включает теорию законченной модели блокчейна для хранения данных.

Пятый раздел «Схемы хранения данных» в данном разделе описывается выбор хранимых данных в базе MongoDB. Для хранения данных выбраны следующие сущности блокчейна – блок, транзакция, внутреннее сообщение, ERC 20 совместимые токены, переводы ERC 20 токенов, балансы ERC 20 токенов.

Схема хранения сущностей — содержит перечисление выбранных для индексации объектов.

Схема хранения блока — содержит описание хранения блока в MongoDB с выбранными индексами.

Схема хранения транзакции — содержит описание хранения транзакции в MongoDB с выбранными индексами.

Схема хранения внутреннего сообщения — содержит описание хранения внутреннего сообщения в MongoDB с выбранными индексами.

Схема хранения контрактов — содержит описание хранения контрактов в MongoDB с выбранными индексами.

Схема хранения переводов токенов — содержит описание хранения переводов токенов в MongoDB с выбранными индексами.

Схема хранения балансов токенов — содержит описание хранения балансов токенов в MongoDB с выбранными индексами.

Шестой раздел «Создание приложения» этот раздел представляет из себя описание процесса разработки приложения.

Описание исполняемых модулей — информация о разработанных исполняемых модулях и зависимостях.

Приложение состоит из трех исполняемых модулей, консольных приложений – Lykke.Job.EthereumSamurai, Lykke.Job.EthereumSamurai.Slave, и вебсервиса, написанного на фреймворке ASP .NET – Lykke.Service.EthereumSamurai.

Источником данных для индексации служит приложение Parity Rpc – узел пиринговой сети, который постоянно находится в процессе синхронизации с основной сетью блокчейна Ethereum, хранит актуальное состояние блокчейна. Доступ к Parity Rpc осуществляется по протоколу HTTP/HTTPS.

База данных MongoDB, в которую записываются данные, используется для операций чтения Lykke.Service.EthereumSamurai, и для операций чтения/записи Lykke.Job.EthereumSamurai.

Описание API — список методов веб-приложения Lykke.Service.EthereimSamurai для доступа к проиндексированным данным.

Описание реализации распределенных вычислений — описание масштабирования пройесса индексации.

Для решения задачи горизонтального масштабирования индексера идеально подходит модель акторов [4] и конкретная его реализация — фреймворк Akka.NET. В фреймворк Akka.NET входит пакет Akka.Cluster, который. Кластер представляет отказоустойчивую, эластичную, одноранговую сеть приложений Akka.NET [5]. Akka.Cluster — это модуль, который дает вам возможность создавать эти приложения. Akka.Cluster — это пакет, который обеспечивает поддержку кластеризации Akka.NET, и это достигается путем добавления в Akka.NET следующих возможностей:

- упрощает создание одноранговых сетей приложений Akka.NET. Позволяет приложениям автоматически обнаруживать новые узлы и автоматически удалять мертвых без изменений конфигурации;
- позволяет пользовательским классам подписываться на уведомления об изменениях в доступности узлов в кластере;
- представляет концепцию «ролей» для различения различных приложений Akka.NET в кластере;
- позволяет создавать кластерные маршрутизаторы, которые являются расширением встроенных маршрутизаторов Akka.NET, за исключением того, что кластерные маршрутизаторы автоматически настраивают список своих маршрутов на основе доступности узлов.

Преимущества Akka.Cluster:

- отказоустойчивость — кластеры изящно восстанавливаются после сбоев (особенно сетевых разделов);
- эластичность — кластеры по своей сути эластичны и могут масштабироваться вверх / вниз по мере необходимости;
- децентрализованность — возможно иметь несколько равных реплик заданного микросервиса или состояния приложения, выполняющегося одновременно в кластере;
- одноранговая сеть — новые узлы могут связываться с существующими одноранговыми узлами, получать уведомления о других партнерах и полностью интегрироваться в сеть без каких-либо изменений конфигу-

рации;

— нет единой точки отказа / узкого места — несколько узлов могут обслуживать запросы, увеличивая пропускную способность и отказоустойчивость.

Основные концепции Akka.Cluster [6]:

— узел — логический член кластера.
— кластер — набор узлов, соединенных через службу членства. Несколько приложений Akka.NET могут быть частью одного кластера.
— «сплетня» — основные сообщения, которые запускают сам кластер.
— лидер — один узел внутри кластера, который добавляет / удаляет узлы из кластера.
— роль — названная ответственность или приложение внутри кластера. Кластер может иметь в нем несколько приложений Akka.NET, каждый из которых имеет свою собственную роль. Узел может существовать одновременно в нескольких ролях.
— конвергенция — когда кворум (простое большинство) сообщений сплетен соглашается на изменение состояния члена кластера.

Настройка приложения — пример настройки всех частей разработанного приложения и запуск.

Настройка узла Parity — пример настройки узла CLI Parity для доступа к данным блокчейна по протоколу HTTP.

Разработка смарт-контрактов — описание процесса разработки токена, являющегося смарт-контрактом для дальнейшего использования в примере.

Развертывание смарт-контрактов — описание процесса развертывания токена, в тестовой сети Ropsten.

Сравнение с работой Etherscan — содержит пример работы разработанного приложения, а также сравнение с результатом работы Etherscan.

Процесс индексации — описание применения модели акторов к процессу индексации.

Алгоритмы индексации — сам алгоритм индексации. Состоит алгоритм индексации блоков из следующих шагов:

— Скачивание метаданных блока;
— Определение, появился ли форк в сети, если да, то запускаем данный алгоритм для предыдущего блока;

— Получение информации по каждой из транзакций блока;
— Получение трейса выполнения транзакции(если транзакция вызывает функцию смарт контракта);
— Разбор внутренних сообщений транзакций(если имеются);
— Формирование данных для записи в базу данных;
— Удаление старых данных из базы по номеру блока;
— Запись данных в базу.

Исследование работы кластера заключительный раздел содержит результаты эффективности работы выполнения индексации и оценки влияния параллелизма на время выполнения индексации на ограниченном объеме блоков.

Сравнение результатов — содержит результаты индексации и начальные условия.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы были достигнуты все поставленные цели, а результатом данной работы является приложение с открытым исходным кодом. Приложение удовлетворяет всем поставленным требованиям.

Исследования в этом направлении могут быть продолжены. Это приложение может быть расширено и использоваться не только для индексации стандартов сети Ethereum, но и для индексации любых событий и операций пользовательских смарт-контрактов.

Практическая значимость разработки заключается в том, что используя и модифицируя архитектуру данного приложения можно написать индексер для любого блокчейна, а также любой желающий сможет использовать данное приложение для разработки систем, использующих блокчейн. Более того, благодаря использованию фреймворка Akka.Cluster можно значительно ускорить индексацию блокчейна подключая дополнительные узлы в кластер.

В процессе разработки были получены знания по конкретной блокчейн платформе Ethereum и рассмотрены стандарты цифровых активов, а также применена модель акторов для параллельных вычислений. Код проекта находится в общем доступе по ссылке

https://github.com/LykkeCity/EthereumSamurai.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Swan, M.* Blockchain: Blueprint for a New Economy / M. Swan. — O'Reilly Media, 2015.

2 *Etherscan,* Etherscan / Etherscan. — 2016. — URL: https://etherscan.io (Дата обращения 30.09.2017). Загл. с экр. Яз. англ.

3 *Etherchain,* Etherchain / Etherchain. — 2016. — URL: https://etherscan. io (Дата обращения 30.09.2017). Загл. с экр. Яз. англ.

4 *Mougayar, W.* The Business Blockchain (summary): Promise, Practice, and Application of the Next Internet Technology / W. Mougayar. — getAbstract, 2016.

5 *Morley, J.* That Book on Blockchain: A One-hour Intro / J. Morley. — CreateSpace Independent Publishing Platform, 2017.

6 *Icahn, G.* Ethereum: The Complete Guide to Understanding Ethereum / G. Icahn. — CreateSpace Independent Publishing Platform, 2017.

7 *Eddison, L.* Ethereum: A Deep Dive Into Ethereum / L. Eddison. — Createspace Independent Publishing Platform, 2017.

8 *Ozer, R.* Ethereum: The Insider Guide to Blockchain Technology, Cryptocurrency and Mining / R. Ozer. — Createspace Independent Publishing Platform, 2017.

9 *Ray, J.* Ethereum whitepaper / J. Ray. — 2016. — URL: https://github.com/ethereum/wiki/wiki/White-Paper (Дата обращения 30.09.2017). Загл. с экр. Яз. англ.

10 *Dannen, C.* Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners / C. Dannen. — Apress, 2017.

11 *Ray, J.* Ethereum development tutorial / J. Ray. — 2016. — URL: https://github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial (Дата обращения 30.09.2017). Загл. с экр. Яз. англ.

12 *Prusty, N.* Building Blockchain Projects / N. Prusty. — Packt Publishing, 2017.

13 *Mukhopadhyay, M.* Ethereum Smart Contract Development / M. Mukhopadhyay. — Packt Publishing, 2018.

14 *Antonopoulos, A.* Mastering Bitcoin: Programming the Open Blockchain / A. Antonopoulos. — O'Reilly Media, 2017.

15 *Drescher, D.* Blockchain Basics: A Non-Technical Introduction in 25 Steps / D. Drescher. — Apress, 2017.

16 *Diedrichs, H.* Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralised Autonomous Organisations / H. Diedrichs. — CreateSpace Independent Publishing Platform, 2016.

17 *Brass, P.* Advanced Data Structures / P. Brass. — CAMBRIDGE UNIVERSITY PRESS, 2008.

18 *Vaughn-Vernon,* Reactive Messaging Patterns with the Actor Model / Vaughn-Vernon. — Pearson Education, 2015.

19 *Brown, A.* Reactive Applications with Akka.net / A. Brown. — Manning Publications, 2017.

20 *Bernhardt, M.* Reactive Messaging Patterns with the Actor Model: Applications and Integration in Scala and Akka / M. Bernhardt. — Manning Publications, 2016.