Gravity: протокол распределенного реестра управления данными

10 июля 2018 г.

Аннотация

Концепция распределенного реестра транзакций защищенного криптографией продемонстрировала свою эффективность. Четвертая промышленная революция задает новый уровень требований к децентрализованным протоколам. Проект Gravity смотрит на задачи распределенного реестра через призму потребностей управления увеличивающемся массивом данных, создаваемых в результате экономической деятельности в цифровом обществе. Архитектура протокола стремиться соответствовать этой задаче, используя алгоритм консенсуса Делегированное доказательство значимости (Delegated Proof of Importance, DPoI). DPoI - высокопроизводительный, энергоэффективный и стимулирующий развитие сети алгоритм, вознаграждающий участников сети за операции, являющуюся положительной для экономики системы. Делегированное доказательство значимости развивает существующие блокчейн решения, объединяя концепции Делегированного доказательства доли (Delegated Proof of Stake, DPoS) и Теории сетей. Протокол системы спроектирован с учетом требований бизнеса и рядовых пользователей, таких, как приватность, аудируемость, мягкая волатильность, достигаемая за счет динамической эмиссии пропорциональной росту активности сети, опираясь на Закон Меткалфа.. [1].

1 Introduction / Введение

Последние годы были отмечены необычайным ростом интереса к технологии блокчейн, при этом основным ее применением пока было создание распределенных платежных сетей. Такие сети являются децентрализованными и позволяют осуществлять быстрые и недорогие транзакции без посредников. Хотя экономические достоинства и недостатки платежных

сетей, основанных на блокчейне, заслуживают отдельного рассмотрения в рамках данной бумаги мы планируем подробнее остановиться на технических особенности работы алгоритмов консенсуса в блокчейне.

Алгоритм консенсуса в блокчейне – это механизм, с помощью которого, в условиях отсутствия централизованного ведения журнала транзакций (реестра) одним субъектом, узлы сети могут достичь консенсуса о содержимом реестра. Именно алгоритм консенсуса является фундаментом любой блокчейн сети, в значительной степени определяющим ее технические особенности.

1.1 Previous Work / Обзор литературы

Задача распределенного консенсуса для сетей, в которых узлы могут быть злоумышленниками (т.н. задача византийских генералов), впервые была сформулирована задолго до появления блокчейна - в 1982 г. [2] С тех пор был разработан целый ряд различных вариантов ее решения [3], однако применительно к блокчейнам первым вариантом решения проблемы стал изложенный в статье Сатоши Накамото [4] алгоритм Proof-of-Work (PoW, доказательство работы). Несмотря на свои преимущества PoW характеризуется рядом недостатков: проблемы с масштабируемостью и производительностью [5], безопасностью [6], постепенная централизация сети в руках крупнейших майнинг-пулов [8], и самое важное - необходимость использования значительного объема реальных ресурсов (электроэнергии и вычислительной техники) для генерации каждого [9]. На сегодня вычислительные ресурсы, расходуемые на хеширование блоков в биткоине, огромны и намного превышают мощности крупнейших суперкомпьютеров, а расход энергии на майнинг сопоставим с потреблением электроэнергии целых стран и продолжает расти [7]. Для устранения данных недостатков был придуман альтернативный алгоритм консенсуса - Proof-of-Stake (PoS, доказательство доли), впервые реализованный в 2012 году в криптовалюте PPCoin (сейчас известна под названием PeerCoin) [10]. В сетях с консенсусом PoS вероятность сформировать следующий блок зависит от объема токенов на депозите участника сети. Данный алгоритм также обладает рядом недостатков и, по мнению ряда экспертов, в текущем виде не является адекватной заменой PoW [11] [12]. Одним из важнейших недостатков PoS является то, что он дает дополнительную мотивацию к накоплению средств в одних руках, что способствует централизации сети. Развитием PoS стал алгоритм консенсуса Delegated Proof-of-Stake (DPoS, делегированное доказательство доли) [13]. Разделение участников сети на делегирующих и валидирующих (делегатов) обеспечивает лучшую масштабируемость и производитель-

ность. Однако в DPoS сохраняется проблема мотивации участников сети к консолидации токенов, вместо их активного использования, что отрицательно сказывается на росте и управляемости сети. Для стимулирвоания активности участников сети был спроектирован алгоритм консенсуса Proof-of-importance (PoI, доказательство значимости), используемый в блокчейне NEM [14]. Ключевое отличие от PoS протоколов состоит в том, что при выборе валидатора блока учитывается не только баланс токенов акаунта, но и транзакционная активность участника. Таким образом, алгоритм мотивирует пользователей быть активными, совершая транзакции и поощряя участие в развитии сети. Несмотря на свои достоинства PoI не лишен недостатков, связанных с производительностью.

1.2 Driving factor / Цель проекта

Цель Gravity – создание протокола управления распределенным реестром с эффективным перераспределением влияния в системе, поощряющего пользователей принимать активное участие в развитии сети и препятствующего централизации. Для решения существующих в современных блокчейн решениях проблем с масштабируемостью, производительностью и безопасностью в протоколе Gravity реализован алгоритм достижения консенсуса DPoI (Delegated Proof of Importance). Алгоритм сочетает в себе достоинства DPoS и PoI, обладая преимуществами оных, реализует возможность делегирования права валидации блоков ограниченному количеству эккаунтов для достижения высокой производительности и масштабируемости сети, учитывая транзакционную активность участников протокола, способствуя ее развитию.

2 Принцип работы Gravity Protocol Consensus Algorithm

Алгоритм консенсуса Gravity базируется на алгоритме Делегированное доказательство значимости (Delegated Proof of Importance, DPoI), основанном на измененном ядре Graphene, учитывая дополнительно к доли владения узла его транзакционную активность. В протоколе Gravity всем эккаунтам сети доступна возможность делегировать право валидировать блоки ограниченному кругу эккаунтов. Для борьбы с имитацией активности между несколькими аффилированными эккаунтами используется разбиение транзакционного графа на кластеры с помощью алгоритма SCAN. В общих чертах принцип работы алгоритма консенсуса протокола Gravity описан в последующих разделах.

2.1 Вычисление индекса значимости

Индекс значимости эккаунта вычисляется следующим образом:

$$r_i = (1 - \omega)v_i + \omega\pi_i$$

Здесь v_i – индекс доли владения, π_i – индекс активности, полученный с помощью алгоритма NCDAwareRank, ω — весовой коэффициент.

Индекс доли владения определяется количеством токенов, принадлежащих эккаунту, и представляет собой отношение баланса эккаунта к общему количеству токенов в системе. Таким образом, у любого эккаунта с ненулевым балансом будет ненулевой индекс значимости. Индекс активности зависит от истории транзакций, связанных с данным эккаунтом. Индекс активности вычисляется не для всех эккаунтов, а только для тех, баланс которых на момент вычисления индекса превышает некоторый порог A_0. Это значение не является фиксированным, и определяется участниками комитета. Также при вычислении инекса активности учитываются не все транзакции, а только те, в которых сумма передаваемых токенов превышает порог T_0, который также определяется комитетом. Скрытые транзакции не учитываются при вычислении индекса активности. Учитываются транзакции за все время, но алгоритм вычисления работает таким образом, что вклад каждой каждой транзакции уменьшается со временем экспоненциально.

2.2 Алгоритм NCDAwareRank

Индекс активности в соответствии с алгоритмом NCDAwareRank вычисляется с помощью следующего рекуррентного соотношения:

$$\boldsymbol{\pi}^{(i+1)} = (\eta\boldsymbol{O} + \mu\boldsymbol{M} + (1 - \eta - \mu)\boldsymbol{E})\boldsymbol{\pi}^{(i)}$$

$\pi^{(i)}$ – вектор, каждый элемент которого – индекс активности эккаунта. Вектор нормированный, сумма его элементов равна 1. $\boldsymbol{O}$ – outlink-матрица, $\boldsymbol{M}$ – матрица межуровневой близости. Об определении этих матриц см. ниже. η и μ – весовые коэффициенты, определяющие вклад матриц $\boldsymbol{O}$ и $\boldsymbol{M}$. Их сумма должна быть меньше единицы. $\boldsymbol{E}$ – матрица телепортации, добавленная для обеспечения сходимости ряда. Она определяется так:

$$\boldsymbol{E} = \frac{1}{N}\boldsymbol{e}$$

где N – количество эккаунтов, и $\boldsymbol{e}$ – матрица, все элементы которой равны 1. Вычисления продолжаются, пока при некотором i не будет выполнено условие:

$$norm(\boldsymbol{\pi}^{(i+1)} - \boldsymbol{\pi}^{(i)}) < \varepsilon$$

Здесь norm() - это L^1-норма вектора, определенная как сумма абсолютных значений его элементов, ε — заранее заданная точность вычислений. В качестве начального приближения, $\pi^{(0)}$, может быть использован вектор, все элементы которого равны $\frac{1}{N}$.

2.3 Вычисление outlink-матрицы

Outlink-матрица O вычисляется следующим образом. Сначала вычисляется матрица весов:

$$w_{ij} = \sum_{k|i\to j, h_k \geq H_0 \wedge h_k \leq H_0+W} \theta(a_k - T_0)\theta(s_i - A_0)\theta(s_j - A_0)a_k \exp\left(lnK[\frac{h_k}{D}]\right)$$

Здесь a_k — сумма k-й транзакции, h_k — глубина k-й транзакции (порядковый номер блока, отсчитанный от текущего момента), s_i - баланс i-го эккаунта, T_0 и A_0 – параметры, определяющие порог для суммы транзакции и баланса эккаунта, $\theta(x)$ – ступенчатая фенкция, равная 0 для $x < 0$ и 1 для $x \geq 0$, K и D — параметры, определяющие, насколько уменьшается вклад каждой транзакции со временем. Смысл этих параметров в том, что каждые D блоков, созданных после определенной транзакции, ее вклад уменьшается как $w' = Kw$.

Суммирование ведется по всем транзакциям, перемещающим некоторую сумму с эккаунта i на эккаунт j, глубина которых лежит в промежутке между H_0 и $H_0 + W$. Здесь H_0 и W - параметры, для тестнета используются значения $H_0 = 2419200$ (что соответствует 28 дням при продолжительности блока в 1 секунду) и $W = 1000$. Таким образом, при вычисления индекса активности учитывается только транзакции, попадающие в окно шириной W блоков, последний блок которого отстоит от текущего блока на H_0.

$$\hat{o}_{ij} = \begin{cases} w_{ji} - w_{ij} & \text{если } w_{ji} - w_{ij} > 0, \\ 0 & \text{в ином случае.} \end{cases}$$

Далее, полученная матрица нормируется таким образом, что сумма элементов в каждом столбце была равна единице.

$$o_{ij} = \begin{cases} \frac{\hat{o}_{ij}}{\sum\limits_k \hat{o}_{kj}} & \text{если } \sum\limits_k \hat{o}_{kj} > 0, \\ 0 & \text{в ином случае.} \end{cases}$$

2.4 Вычисление матрицы межуровневой близости

Множество всех эккаунтов W, участвующих в расчете индекса активности, разбивается на непересекающиеся подмножества A_i, называемые NCD-блоками (NCD – nearly completely departed). О принципе разбиения см. описание алгоритма SCAN. Для данного эккаунта u рассмотрим множество всех эккаунтов G_u, такое, что для любого эккаунта $v \in G_u$ соответствующий член outlink-матрицы больше нуля: $o_{uv} > 0$. Множество ближних эккаунтов u тогда определяется следующим образом:

$$\chi_u = \bigcup_{v \in \{u\} \cup G_u} A_{(v)}$$

Здесь $A_{(v)}$ – NCD-блок A_i, такой, что $v \in A_i$,

Матрица межуровневой близости определяется тогда так:

$$M_{vu} = \begin{cases} \frac{1}{N_u |A_{(v)}|} & \text{если } v \in \chi_u \ , \\ 0 & \text{в ином случае.} \end{cases}$$

Здесь N_u означает число NCD-блоков в χ_u.

2.5 Разбиение графа на кластеры с помощью алгоритма SCAN

Построим ненаправленный граф $W = \{V, E\}$, в котором каждая вершина соответствует эккаунту, и каждое ребро – ненулевому элементу outlink-матрицы. Структурой данной вершины v назовем множество всех соседних вершин:

$$\Gamma(v) = \{w \in V | (v, w) \in E\} \cup \{v\}$$

Структурным сходством двух вершин назовем следующую величину:

$$\sigma(v, w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{\sqrt{|\Gamma(v)||\Gamma(w)|}}$$

ε-окружением вершины назовем множество вершин такое, что

$$N_\varepsilon(v) = \{w \in \Gamma(v) | \sigma(v, w) \geq \varepsilon\}$$

Ядром назовем вершину, у которой число элементов в ε-окружении больше некоторого порога μ.

$$CORE_{\varepsilon,\mu}(v) \Leftrightarrow |N_\varepsilon(v)| \geq \mu$$

Вершина w находится в прямой структурной доступности от вершины v, если

$$DirREACH(v, w) \Leftrightarrow CORE_{\varepsilon,\mu}(v) \vee w \in N_{\varepsilon}(v)$$

Вершина w находится в структурной доступности от вершины v, если

$$REACH(v, w) \Leftrightarrow \exists v_1, ...v_n \in V \forall i \in \{1, ...n-1\} DirREACH(v_i, v_{i+1})$$

Вершина v структурно соединена с вершиной w, если

$$CONNECT(v, w) \Leftrightarrow \exists u \in VREACH(u, v) \vee REACH(u, w)$$

Подмножество вершин, каждая из которых структурно соединена со всеми остальными, называется кластером. Можно показать, что каждая вершина может принадлежать только одному кластеру.

2.6 Использование индекса значимости в системе

Индекс значимости используется для двух целей.

Во-первых, при очередной эмиссии индекс значимости определяет долю новых токенов, которую получит каждый эккаунт.

Во-вторых, индекс значимости определяет вес данного эккаунта при голосовании. Голосование позволяет делегировать определенные полномочия в системе ограниченному количеству эккаунтов.

Путем голосования выбираются узлы, которые осуществляют формирование блоков (witness-ноды).

Также путем голосования выбираются члены комитета. Комитет может голосованием менять параметры блокчейна, такие, как размер комиссии за трансфер, вознаграждение witness-нодам и т.п.

3 Масштабируемость

4 Защита / Security

5 Токен протокола

Объем обрабатываемых операций блокчейн протоколом напрямую зависит от доступной вычислительной мощности, предоставленной участниками сети. Для эффективного распределения ресурсов системы и предотвращения spam атак, при совершении операций в сети Gravity с пользователей удерживается комиссия системы в криптографическом токене

протокола. Протокол позволяет совершать транзакции трансфера токена между эккаунтами участников сети и вызова умных контрактов системы, таких как мульти-подпись, регистрация эккаунтов, создание пользовательских токенов и т.д.

6 Эмиссия и распределение премайнед монет

Стартовая эмиссия составляет 1000000000 токенов протокола, распространяется по изначальным эккаунтам сети для запуска функционирования протокола. В проекте Gravity используется динамическая эмиссия. Эмиссия производится регулярно, в моменты времени $t_0, t_1, ...t_i$, где $t_{i+1} = t_i + T$. Объем эмиссии зависит от роста активности сети за предыдущий период T.

6.1 Вычисление активности сети за период

Вычислим сперва матрицу весов по следующей формуле:

$$w_{ij}(t_n) = \sum_{k|i \to j, t_k \in [t_{n-1}, t_n]} a_k$$

Здесь a_k – сумма k-й транзакции, t_k – время создания k-й транзакции. Суммирование ведется по всем транзакциям, перемещающим некоторую сумму с эккаунта i на эккаунт j, созданным за период от t_{n-1} до t_n. Учитываются все транзакции за этот период, параметры T_0 и A_0 не влияют на значение активности сети за период.

Каждый элемент матрицы w_{ij} представляет собой вес связи между эккаунтом i и j за данный период.

Вычислим теперь матрицу связей l:

$$l_{ij}(t_n) = \begin{cases} 1 & \text{если } w_{ji}(t_n) - w_{ij}(t_n) > 0, \\ 0 & \text{в ином случае.} \end{cases}$$

Активность за период мы будем вычислять так:

$$A(t_n) = \sum_{i,j} l_{ij}(t_n)$$

Таким образом, активность вычисляется как количество связей между активными эккаунтами за данный период.

6.2 Вычисление объема эмиссии

Объем эмиссии зависит от роста активности сети.

Определим величину E_T, которую назовем целевой величиной эмиссии. Она задает верхнюю границу совокупной величины эмиссии, достижимой при данном значении активности A.

$$\Delta A(t_n) = A(t_n) - A_{max}(t_{n-1})$$

$$E_T(t_n) = \begin{cases} E_T(t_{n-1}) + K_E \Delta A(t_n), & \text{если } \Delta A(t_n) > 0, \\ 0 & \text{в ином случае.} \end{cases}$$

Здесь K_E - коэффициент, определяющий максимальную величину эмиссии при увеличении активности на единицу, $A_{max}(t_{n-1})$ - максимальное предыдущее значение активности с момента запуска системы:

$$A_{max}(t_{n-1}) = max\Big(A(t_i), t_i \in [t_0, t_{n-1}]\Big)$$

Величина эмиссии, выпускаемая в момент времени t, вычисляется по формуле:

$$E(t_n) = \lambda S(t_{n-1}) f\Big(\kappa \frac{E_T(t_n) - E_S(t_{n-1})}{\lambda S(t_{n-1})}\Big)$$

Здесь λ – предельный рост количества токенов в системе S за одну эмиссию. Он определяется из параметра L, который задает предельный рост S в год (выраженный в процентах):

$$\lambda = (1 + \frac{L}{100})^{1/N} - 1$$

Здесь N - количество эмиссий в год.

E_S - суммарная эмиссия за предыдущий период:

$$E_S(t_n) = \sum_{k \in [0,n]} E(t_k)$$

$f(x)$ - сигмоидальная функция (Рис. 1). В имеющейся реализации алогитма в качестве этой функции используется гиперболический тангенс.

κ – коэффициент от 0 до 1, определяющий скорость, с которой полная эмиссия приближается к целевой эмиссии E_T, в случае, если активность остается но одном уровне на протяжении длительного срока.

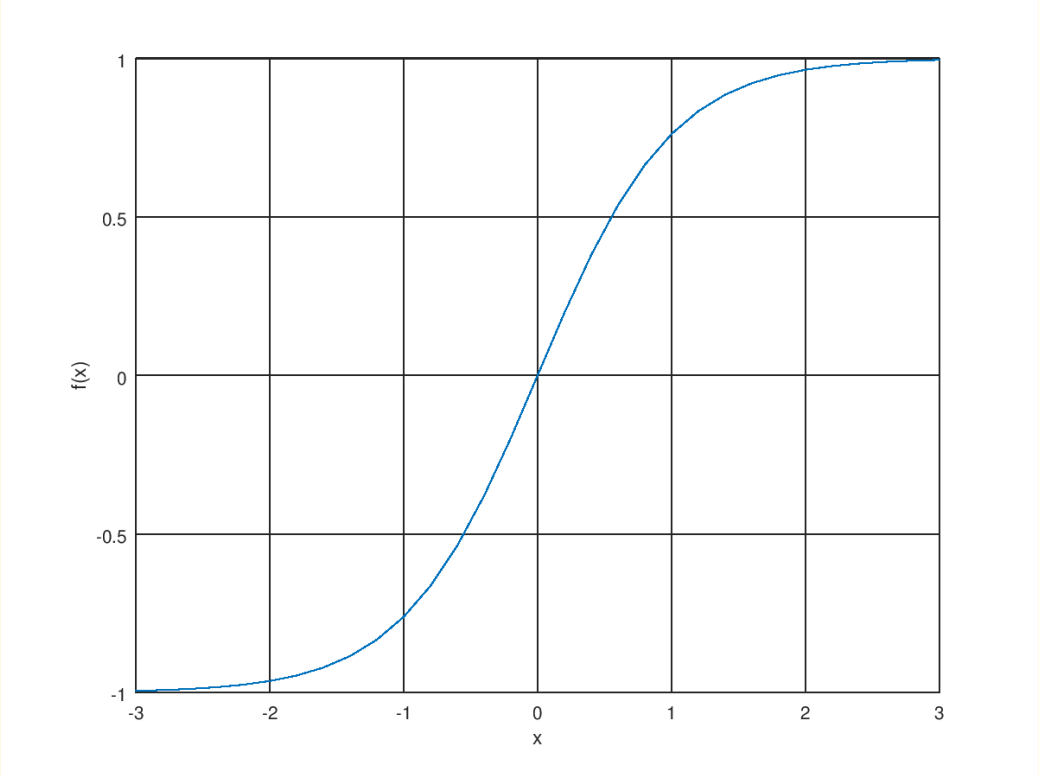

Рис. 1: Сигмоидальная функция

6.3 Анализ формулы динамической эмиссии

7 Заключение

Список литературы

[1] Metcalfe, B. (2013). Metcalfe's law after 40 years of ethernet. Computer, 46(12), 26-31. URL: http://ieeexplore.ieee.org/abstract/document/6636305/

[2] Lamport, L., Shostak, R., Pease, M. (1982). The Byzantine generals problem. ACM Transactions on Programming Languages and Systems (TOPLAS), 4(3), 382-401. URL: https://www.microsoft.com/en-us/research/uploads/prod/2016/12/The-Byzantine-Generals-Problem.pdf

[3] Castro, M., Liskov, B. (2002). Practical Byzantine fault tolerance and proactive recovery. ACM Transactions on Computer Systems (TOCS), 20(4), 398-461. URL: https://dl.acm.org/citation.cfm?doid=571637.571640

[4] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. URL: https://bitcoin.org/bitcoin.pdf

[5] Croman, K. et al. (2016). On scaling decentralized blockchains. In International Conference on Financial Cryptography and Data Security (pp. 106-125). Springer, Berlin, Heidelberg. URL: http://www.comp.nus.edu.sg/ prateeks/papers/Bitcoin-scaling.pdf

[6] Eyal, I., Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. In International conference on financial cryptography and data security (pp. 436-454). Springer, Berlin, Heidelberg. URL: https://arxiv.org/pdf/1311.0243.pdf

[7] Bitcoin Energy Consumption Index. digiconomist.net. URL: https://digiconomist.net/bitcoin-energy-consumption

[8] Buterin, V. (2014). Mining Pool Centralization at Crisis Levels. URL: https://bitcoinmagazine.com/articles/mining-pool-centralization-crisis-levels-1389302892/

[9] Bentov, I., Gabizon, A., Mizrahi, A. (2016). Cryptocurrencies without proof of work. In International Conference on Financial Cryptography and Data Security (pp. 142-157). Springer, Berlin, Heidelberg. URL: https://link.springer.com/chapter/10.1007/978-3-662-53357-4_10/

[10] King, S., Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. URL: https://peercoin.net/assets/paper/peercoin-paper.pdf

[11] Demeester, T. (2017). Critique of Buterin's A Proof of Stake Design Philosophy. URL: https://medium.com/@tuurdemeester/critique-of-buterins-a-proof-of-stake-design-philosophy-49fc9ebb36c6

[12] Poelstra, A. (2014). Distributed consensus from proof of stake is impossible. URL: https://download.wpsoftware.net/bitcoin/old-pos.pdf

[13] Dantheman. (2017). DPOS Consensus Algorithm - The Missing White Paper. URL: https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper

[14] NEM Technical Reference. Version 1.2.1. February 23, 2018 URL: https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf

Содержание