ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 26 за 2021 г.

ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Е.А. Бахвалова, В.А. Судаков

Исследование алгоритмов консенсуса для блокчейн-платформ

Рекомендуемая форма библиографической ссылки: Бахвалова Е.А., Судаков В.А. Исследование алгоритмов консенсуса для блокчейн-платформ // Препринты ИПМ им. М.В.Келдыша. 2021. № 26. 16 с. https://doi.org/10.20948/prepr-2021-26
https://library.keldysh.ru/preprint.asp?id=2021-26

Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук

Е.А. Бахвалова, В.А. Судаков

Исследование алгоритмов консенсуса для блокчейн-платформ

Москва — 2021

Е.А. Бахвалова, В.А. Судаков

Исследование алгоритмов консенсуса для блокчейн-платформ

В данной работе исследованы несколько алгоритмов консенсуса, произведен сравнительный анализ этих алгоритмов, результаты которого немаловажны в условиях активного роста рынка криптовалют. Кроме того, в условиях развития пандемий актуальным является создание блокчейн-технологий распределенного хранения достоверных медицинских данных. Выделены ключевые свойства алгоритмов консенсуса, и произведен детальный анализ алгоритма перспективного алгоритма Istanbul BFT. С использованием многокритериального анализа альтернатив показано, как можно выбирать подходящий алгоритм консенсуса под потребности конкретных задач.

Ключевые слова: блокчейн, консенсус, византийская отказоустойчивость, многокритериальный анализ

Bakhvalova Ekaterina Aleksandrovna, Sudakov Vladimir Anatolievich

Research of consensus algorithms for blockchain platforms

In this paper, several consensus algorithms are investigated, a comparative analysis of these algorithms is made, the results of which are important in the context of the active growth of the cryptocurrency market. In addition, in the context of pandemics, the creation of blockchain technologies for the distributed storage of reliable medical data is relevant. The key properties of consensus algorithms are highlighted, and a detailed analysis of the algorithm of the promising Istanbul BFT algorithm is carried out. Using multi-criterion analysis of alternatives, it is shown how you can select the appropriate consensus algorithm for the needs of specific tasks.

Key words: blockchain, consensus, Byzantine fault tolerance, multicriteria analysis

Исследование выполнено при финансовой поддержке РФФИ и CNPq(Бразилия), Фонда содействия инновациям(Россия), DBT, DST (Индия), MOST, NSFC(Китай), SAMRC(ЮАР) в рамках научного проекта № 20-51-80002

Введение

С популяризацией криптовалют и технологии блокчейн в целом возрос интерес к практическим последствиям различных алгоритмов распределенного консенсуса. Большинство существующих систем в реальной жизни не могут должным образом удовлетворить потребность в широкомасштабном развертывании, поскольку они сталкиваются с серьезными ограничениями. Многие из этих ограничений связаны с лежащим в основе системы алгоритмом консенсуса. Таким образом, в стремлении создать наиболее подходящие практические системы блокчейн основное внимание уделяется именно распределенному консенсусу.

Алгоритм консенсуса – это основной компонент блокчейн системы, который напрямую определяет, как система ведет себя и какой производительности она может достичь. Распределенный консенсус был широко изучаемой темой исследований в распределенных системах, однако с появлением блокчейна он получил особое внимание. Поскольку характеристики различных типов блокчейн систем фундаментально зависят от используемых ими согласованных алгоритмов, необходим систематический анализ существующих алгоритмов консенсуса.

Использование технологии блокчейн в сфере медицины позволяет обеспечить высокий уровень конфиденциальности и надежность хранения данных, а также сокращает потерю времени в процессе обслуживания пациентов и передачи информации о них. С каждым годом объем данных о пациентах только возрастает, а обрабатывать их становится все сложнее, поэтому внедрение блокчейн видится оптимальным решением проблемы как на уровне медицинских организаций и учреждений, так и на государственном уровне во всей системе здравоохранения. Ряд компаний создали объединение для использования технологии суверенной личности в рамках инициативы COVID Credentials, целью которой является разработка паспортов иммунитета [1]. Благодаря блокчейну подделка тестов на COVID становится невозможной. Только сам пациент и организация, предоставляющая данные, имеют доступ к результатам теста. Делиться полученной информацией пользователь может на свое усмотрение.

В настоящей работе проведен полный анализ алгоритма консенсуса IBFT, сформирована сводная таблица ключевых свойств алгоритмов консенсуса на основе работы Md Sadek Ferdous, Alan Colman [2], по которым в дальнейшем можно будет проводить сравнительный анализ алгоритмов, результаты которого смогут помочь в выборе наиболее подходящего алгоритма под конкретные случаи.

Свойства алгоритмов консенсуса

Для алгоритмов консенсуса можно выделить четыре основные группы свойств: структурные свойства, свойства блока и вознаграждения, свойства безопасности и производительности.

Структурные свойства характеризуют структуру узлов сети, которые участвуют в процессе достижения консенсуса. В рамках этой группы выделяют типы, состояния или роли узлов, способы структурирования узлов и механизм определения ключевых в процессе достижения консенсуса узлов.

Немного подробнее рассмотрим способы структурирования узлов. В сети узлы объединяются в комитеты, однако в некоторых алгоритмах подразумевается наличие нескольких комитетов, которые могут образовывать плоскую или иерархическую структуру. Для комитета могут устанавливаться правила включения узла в комитет, он может быть доступен как всем узлам, так и только некоторому кругу узлов. Состав узлов в комитете может быть статическим или изменяться со временем.

Выделяют два типа механизмов определения узлов – голосование и лотерея. В основе лотереи может находиться либо вероятностный механизм на основе криптографии, либо другие рандомизированные механизмы. Процесс голосования в алгоритме может производиться в один или несколько этапов.

На основании всех упомянутых выше характеристик можно вывести дерево структурных свойств алгоритмов, которое представлено на рис.1.

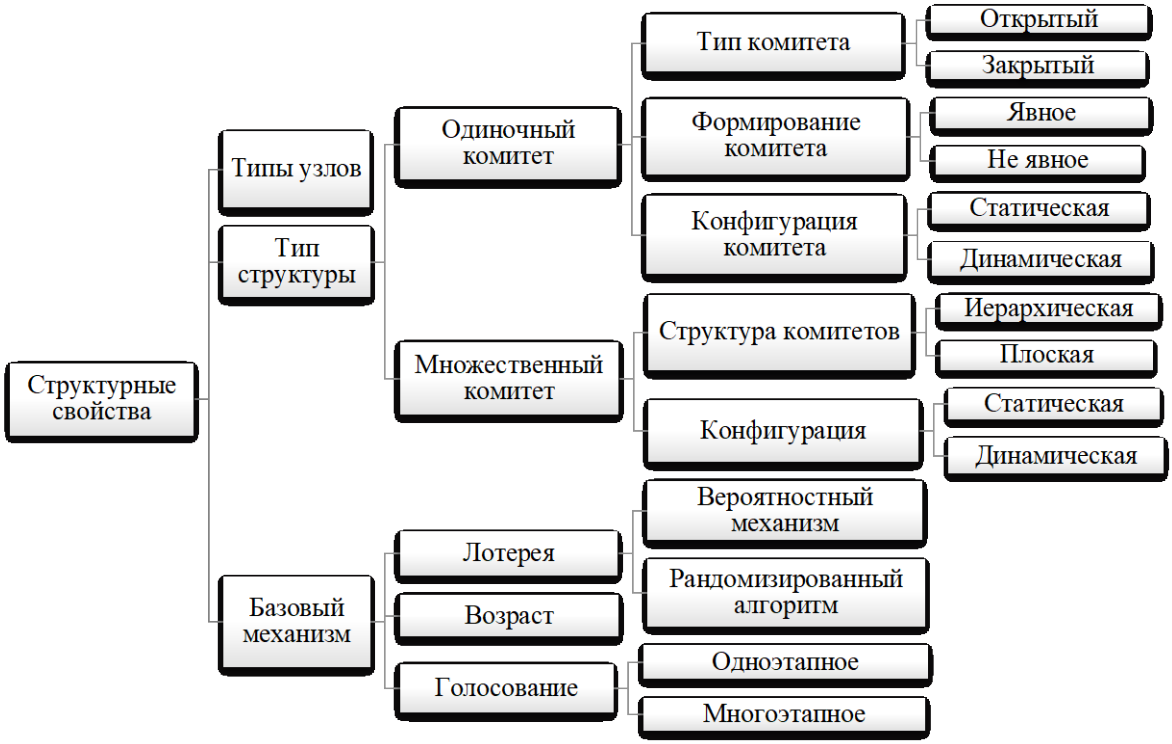

Рис. 1. Структурные свойства

Свойства блока и вознаграждения выделяют для алгоритмов, лежащих в основе различных криптовалют. В основном эти свойства выражают количественные характеристики, такие как, например, вознаграждение за создание блока, общий объем предложения криптовалюты, среднее время создания нового блока. Также среди свойств блока выделяют дату создания первого блока.

Большинство указанных выше характеристик имеют прямое и косвенное влияние на достижение консенсуса в блокчейн платформе, основанной на криптовалюте. Например, вознаграждение за создание майнером нового блока стимулирует его действовать определённым образом в процессе решения криптографической головоломки, которая предназначается для достижения консенсуса в сети.

Схематично свойства блока и вознаграждения представлены на рис. 2.

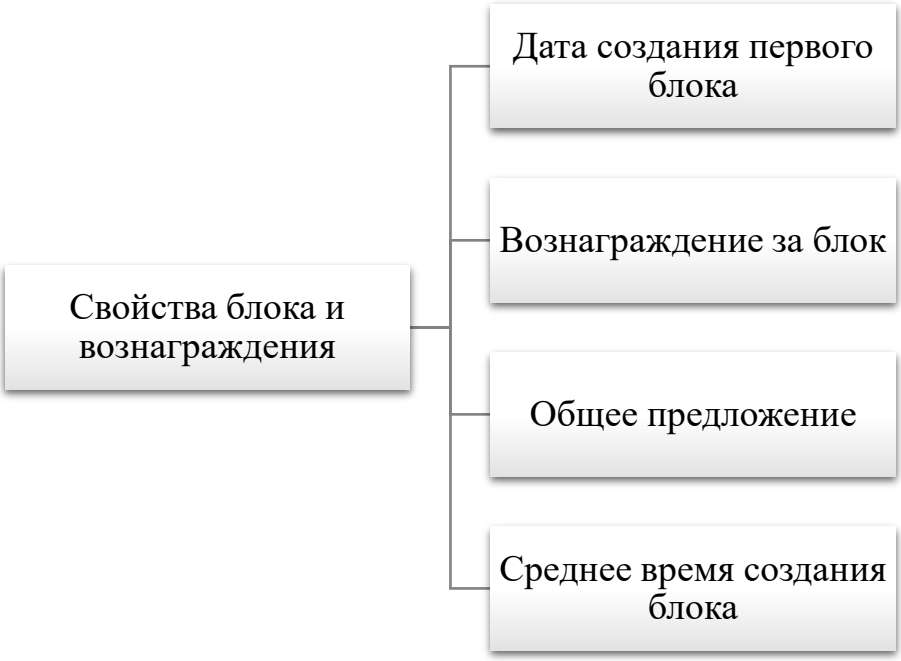

Рис. 2. Свойства блока и вознаграждения

Рассмотрим следующую группу свойств – свойства безопасности. В этом блоке содержатся такие характеристики, как наличие или отсутствие обязательной аутентификации узлов, непосредственно участвующих в процессе достижения консенсуса, соответствие стандартам неотказуемости, устойчивость к цензуре. Также алгоритмы описываются с точки зрения защищенности от различных векторов атак, например, таких как атака Сибиллы, DDoS-атаки и прочих. Дополнительно вводится характеристика сопротивляемости сети, выраженной в максимально возможном количестве византийских узлов, при котором алгоритмом консенсуса все так же гарантируется бесперебойная работа сети.

Атака Сибиллы получила такое название от клинического случая, связанного с пациенткой с диссоциативным расстройством личности. В рамках этой атаки узел-злоумышленник также может дублировать свою личность, как это происходило в аналогичном случае и с этой пациенткой. В распределенной среде злоумышленник множит количество своих сущностей для незаконного получения преимуществ в сети, что, например, вполне может привести к опасности «атаки 51».

Итак, свойства безопасности алгоритмов консенсуса отображены на рис. 3.

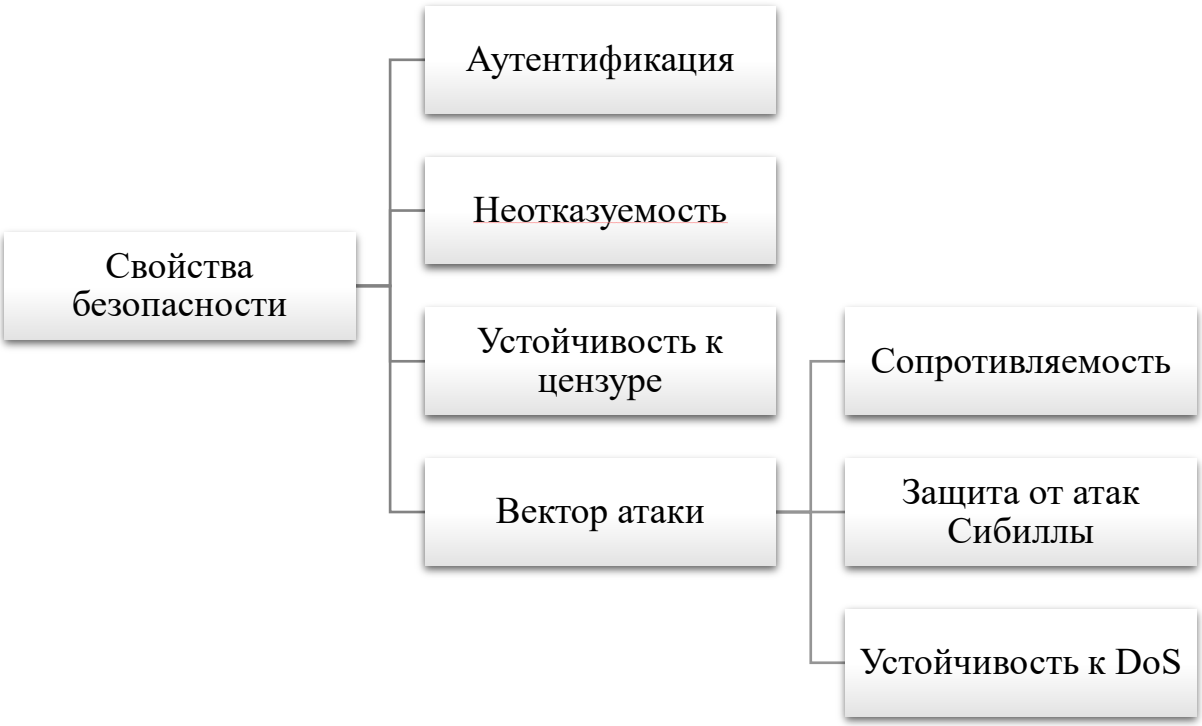

Рис. 3. Свойства безопасности

Наконец, рассмотрим свойства производительности. Свойства этой группы используются для измерения количественных характеристик алгоритмов консенсуса. В этот блок свойств входят следующие показатели: максимальное количество неисправных узлов, которое допускает алгоритм консенсуса, количество транзакций, которые алгоритм обрабатывает за единицу времени, равную, например, одной секунде, возможность увеличения размера сети и функциональности без ухудшения показателей производительности исходной системы, период времени, за который предложение с транзакциями проходит процесс согласования узлами и до момента включения блока с транзакциями в цепочку блоков, также в этот блок свойств входит показатель количества потребляемой алгоритмом энергии [3]. Описанные свойства производительности представлены на рис.4.

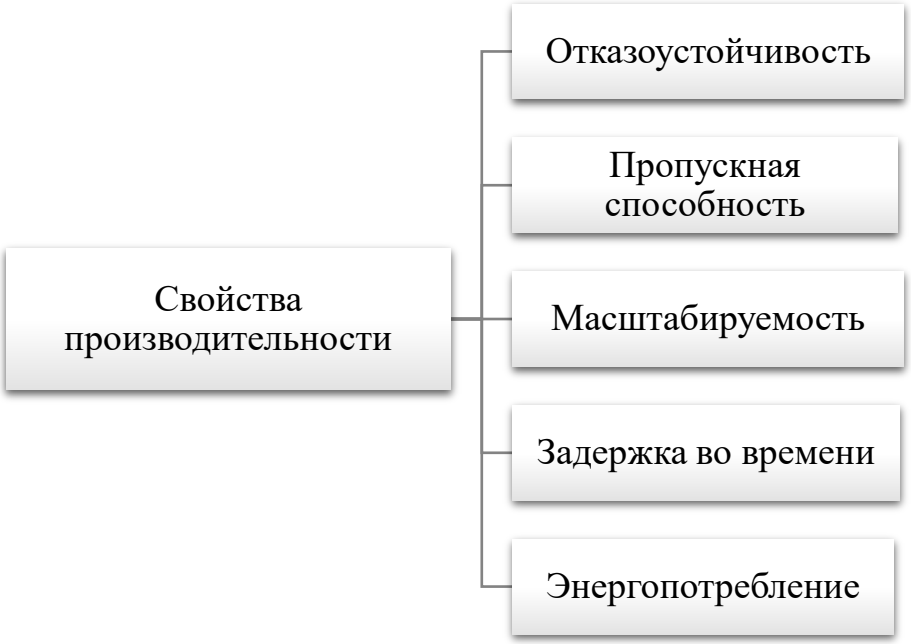

Рис. 4. Свойства производительности

Представленный подход к описанию алгоритмов консенсуса в наиболее лаконичной форме отображает наиболее значимые и ключевые характеристики алгоритмов, по которым можно сформировать полное представление о механизме работы алгоритма.

Далее рассмотрим алгоритм консенсуса Istanbul BFT и попробуем сформировать небольшую сводную таблицу с представленными выше свойствами.

Алгоритм Istanbul BFT

Istanbul Byzantine Fault Tolerance (IBFT) – простой и элегантный византийский отказоустойчивый консенсусный алгоритм, который используется для реализации репликации конечного автомата в блокчейн платформе Quorum. Quorum – проект с открытым исходным кодом на GitHub, целью которого является предоставление блокчейна, который давал бы возможность выполнять транзакции не только публично, но и в приватном режиме. С технической точки зрения, Quorum – это модернизированный Ethereum, у него есть также и свой модифицированный Geth-клиент, чтобы иметь возможность делать приватные транзакции, что актуально для медицинских данных.

Для корректной работы алгоритма IBFT необходима частично синхронная модель связи, при которой безопасность не зависит от каких-либо предположений по времени, а живучесть алгоритма зависит только от периодов синхронизации. В частично синхронной модели присутствует неизвестная верхняя граница задержек выполнения и связи, которая сохраняется в течение

неизвестного периода времени, который принято называть моментом глобальной стабилизации (или сокращенно GST) [4].

Istanbul BFT перенял у алгоритма PBFT трехфазный консенсус, состоящий из этапов $PRE-PREPARE$, $PREPARE$ и $COMMIT$. Система допускает не более f неисправных узлов в сети n узлов, где $n = 3f + 1$. Перед каждым раундом узлы, выполняющие роль валидаторов, выбирают одного из них на роль лидера-предлагающего. Избранный лидер предлагает новое предложение с транзакциями для включения в цепочку блоков и передает его валидаторам совместно с сообщением $PRE-PREPARE$. После получения $PRE-PREPARE$ сообщения от предлагающего валидаторы меняют свое состояние на $PRE-PREPARED$ с последующим отправлением сообщения $PREPARE$. После совершения этого действия все узлы должны удостовериться, что все валидаторы работают в одной последовательности и в одном и том же раунде. После получения валидатором $2f + 1$ сообщений $PREPARE$ он приобретает новое состояние $PREPARED$ и затем передает сообщение $COMMIT$. На этом шаге валидатор ставит в известность своих партнеров о том, что он принимает предложенный блок и собирается добавить его в цепочку. Дождавшись $2f + 1$ сообщений $COMMIT$, валидаторы переходят в состояние $COMMITTED$ и добавляют новый блок в цепочку [5].

Одним из отличительных свойств алгоритма IBFT от других отказоустойчивых алгоритмов является тот факт, что в IBFT алгоритме узлам не присуще полное доверие лидеру-предлагающему блок. В данном алгоритме каждый предложенный лидером блок требует несколько раундов согласования валидаторами для того, что путем голосования между валидаторами было достигнуто взаимное соглашение о принятии нового блока, которое было бы подкреплено набором подписей о содержимом блока.

Алгоритм Istanbul BFT – детерминированный оптимально устойчивый алгоритм, допускающий f неисправных процессов из n, где $n \geq 3f + 1$. Общая сложность связи алгоритма равняется $O(n^2)$. В периоды хорошей связи IBFT достигает завершения за три сообщения.

В алгоритме гарантируется немедленная завершенность. Это означает, что каждый добавленный в цепочку блок не будет удален из нее, за исключением случаев превышения количества византийских узлов заданному ограничению. Также алгоритм использует Proof-of-Activity в качестве метода предотвращения атак Сибиллы [6].

Проведенные испытания алгоритма позволили рассчитать показатели производительности алгоритма. Испытания проводились с участием четырех валидаторов и размером блока – не более 2000 транзакций. Качественные характеристики и показатели на основе испытаний приведены в табл. 1.

Таблица 1

Свойства алгоритма IBFT

Структурные свойства	Типы узлов	Предлагающий лидер Валидатор
	Тип структуры	Единый динамический комитет
	Базовый механизм	Голосование
Свойства блока и вознаграждения	-	Алгоритм не предназначен для криптовалют
Свойства безопасности	Неотказуемость	Не гарантируется работа в синхронной сети
	Сопротивляемость	$3f + 1$
	Вектор атаки	Защита от атак Сибиллы - с помощью алгоритма Proof of Authority (PoA)
Свойства производительности	Отказоустойчивость	$3f + 1$
	Пропускная способность	400 ~ 1200 транзакций в секунду
	Масштабируемость	Хорошо масштабируемый
	Задержка	3 сообщения
	Сложность коммуникации	$O(n^2)$

Рассмотрим доказательства корректности алгоритма IBFT. Для свойств алгоритмов – согласованности, действительности и прекращения действия – приведем доказательства теорем.

Теорема 1. *Валидность (действительность).*

Для заданного извне предиката β, если правильный процесс определяет некоторое значение v, то $\beta\ (v)$ истинно.

Доказательство: чтобы принять решение, правильный процесс должен получить кворум действительных сообщений $\langle COMMIT, \lambda, r, v\rangle$. Сообщение считается действительным, только если оно содержит значение v, такое, что $\beta\ (v)$ истинно. Таким образом, если правильный процесс определил v, тогда $\beta\ (v)$ должно быть истинным [4].

Теорема 2. *Терминация (прекращение действия)*.

Каждый правильный процесс завершается.

Для этого доказательства предполагается, что правильный процесс p_i еще не принял решения, но гарантируется, что он в итоге завершится.

1. Процесс p_i должен в конечном итоге достичь некоторого раунда r (т.е. он устанавливает $r_i = r$ и передает сообщение $\langle ROUND - CHANGE, \lambda, r, -, - \rangle$) после GST с правильным лидером p_L.

Доказательство: поскольку по предположению p_i не принимает решения, его таймер раунда должен продолжаться, пока он не достигнет раунда r или он не получит $f + 1$ сообщений $ROUND - CHANGE$, где r – самый высокий раунд.

Теперь нужно рассмотреть два случая. Один, в котором некоторый правильный процесс p_i уже определил какое-то значение v, и другой, в котором еще не определился правильный процесс.

2. Случай: некоторый правильный процесс p_i уже определил какое-то значение.

2.1. p_i принимает сообщение $\langle ROUND - CHANGE, \lambda, r, -, - \rangle$, транслируемое p_i, и отправляет на p_i кворум допустимых сообщений $COMMIT$ для того же значения v.

Доказательство: согласно пункту 1 после GST все сообщения, отправленные правильными процессами, доставляются вовремя.

2.2. Таким образом, после GST p_i должен получить кворум действительных сообщений $COMMIT$ для значения v, отправленного p_i, и принять решение.

3. Случай: не принято решение о правильном процессе.

3.1. Каждый правильный процесс в конечном итоге достигает раунда r и передает действительное сообщение $\langle ROUND - CHANGE, \lambda, r, -, - \rangle$.

Доказательство: поскольку по предположению правильный процесс не принимает решения, его таймер раунда должен продолжать бесконечно истекать, пока он не достигнет раунда r или он не получит $f + 1$ сообщений $ROUND - CHANGE$, где r – самый высокий раунд.

3.2. Правильный лидер p_L передает действительное и обоснованное сообщение $\langle PRE - PREPARE, \lambda, r, v \rangle$.

Доказательство: после GST p_L должен получать каждое сообщение $\langle ROUND - CHANGE, \lambda, r, -, - \rangle$, передаваемое правильным процессом. Поскольку эти сообщения объединены с сообщениями $PREPARE$, p_L может создать действительное и обоснованное сообщение $\langle PRE - PREPARE, \lambda, r, v \rangle$.

3.3. Таким образом, поскольку это происходит после GST, алгоритм продолжит работу в обычном режиме и p_i примет решение v в конце раунда r.

Доказательство *достижения соглашения между узлами* разделено на леммы 1 и 2 и теорему 3. Лемма 1 используется в качестве поддержки для доказательства достижения соглашения в одном и том же раунде, а лемма 2 – для доказательства достижения соглашения в разных раундах. Теорема 3 использует обе леммы для завершения рассуждений.

Лемма 1. Если какой-то правильный процесс готовит значение v в раунде r, то ни один другой правильный процесс не готовит значение v' в раунде r, так что $v' \neq v$.

Для этого доказательства предполагается, что правильный процесс подготовился для значения v в раунде r и что ни один правильный процесс не может подготовиться к другому значению v' в том же раунде, потому что не набирается кворум сообщений $PREPARE$ для v'.

1. Правильный процесс получил $\left\lceil \frac{n+f}{2} \right\rceil + 1$ действительных сообщений $\langle PREPARE, \lambda, r, v \rangle$.

Доказательство: это следует из предположения. Чтобы подготовиться к значению v и раунду r, правильный процесс должен получить $\left\lceil \frac{n+f}{2} \right\rceil + 1$ действительных сообщений $\langle PREPARE, \lambda, r, v \rangle$.

2. $f + 1$ правильных процессов передали сообщение $\langle PREPARE, \lambda, r, v \rangle$.

Доказательство: это следует из п. 1. Если правильный процесс получил $\left\lceil \frac{n+f}{2} \right\rceil + 1$ действительных сообщений $\langle PREPARE, \lambda, r, v \rangle$, то $f + 1$ из этих сообщений должны быть переданы правильными процессами.

3. Для любого значения $v' \neq v$ было передано не более $\left\lceil \frac{n+f}{2} \right\rceil$ действительных сообщений $\langle PREPARE, \lambda, r, v' \rangle$.

Доказательство: в п. 2 было установлено, что $f + 1$ правильных процессов транслировали сообщение $\langle PREPARE, \lambda, r, v \rangle$. Отсюда следует, что не более $\left\lceil \frac{n+f}{2} \right\rceil$ процессов могли передать сообщение $\langle PREPARE, \lambda, r, v' \rangle$.

4. Итоговое доказательство леммы следует из п. 3, поскольку для подготовки к значению v’ в раунде r правильный процесс требует $\left\lceil \frac{n+f}{2} \right\rceil + 1$ действительных сообщений $\langle PREPARE, \lambda, r, v' \rangle$.

Лемма 1 показала, что никакие два правильных процесса не могут подготовиться к разным значениям в одном и том же раунде.

Лемма 2 показывает, что согласие наступает также и между раундами.

Лемма 2. Если $f + 1$ правильных процессов готовятся к значению v и раунд r, то для любого $\langle PRE - PREPARE, \lambda, r', v' \rangle$ сообщения m, такого что $r' > r$ и $v' \neq v$, $JustifyPreprepare(m)$ должно быть ложным.

Используя две предыдущие леммы, докажем свойство согласованности консенсуса в теореме 3.

Теорема 3. Если правильный процесс p_i определяет какое-то значение v, то ни один правильный процесс p_j не определяет значение v', такое что $v' \neq v$.

Доказательство:

Предположим, что некоторый правильный процесс p_i принимает решение первым и определяет значение v в течение некоторого цикла r. Из леммы 1 можем вывести, что ни один правильный процесс не может решить v' в течение r, потому что никакой правильный процесс не готовится к v' в течение r.

Для любого раунда r', такого что $r' > r$, из леммы 2 следует, что ни один правильный процесс не может решить v' в течение r', потому что нет сообщения m $\langle PRE - PREPARE, \lambda, r', v' \rangle$, такого, что $JustifyPreprepare(m)$ верно. Следовательно, никакое сообщение $\langle PRE - PREPARE, \lambda, r', v' \rangle$ не принимается ни одним правильным процессом и алгоритм не продвигается к решению по v'.

Качественное сравнение свойств алгоритмов

Сравнивая алгоритм IBFT с уже проанализированными другими авторами алгоритмами, предназначенными также для закрытых платформ, наглядно можем увидеть, по каким характеристикам алгоритм отличается от уже ранее проанализированных.

Проведем сравнения с алгоритмами RBFT и PoET. Перед этим кратко опишем механизм выбора лидера по каждому алгоритму.

Отличительной особенностью алгоритма RBFT (Redundant Byzantine Fault Tolerance) является тот факт, что каждый участвующий узел развертывает два (или более) экземпляра протокола, метко названных экземпляром протокола Master и Backup, каждый из которых выполняется параллельно. Лидер, в свою очередь, выбирается из главного и резервного экземпляра протокола. Он несет ответственность за упорядочивание транзакций. Его характеристики, то есть задержка и пропускная способность, периодически наблюдаются другими узлами, и, если его производительность ухудшается, из резервного экземпляра выбирается другой лидер.

Алгоритм PoET (Proof-of-Elapsed-Time) в каждом раунде поддерживает смену лидера. Каждый узел валидатора в сети запрашивает время ожидания от доверенной функции. Валидатор, которому назначено самое короткое время ожидания, выбирается лидером этого раунда. Победивший валидатор затем может предложить блок, состоящий из серии транзакций из определенного семейства транзакций. Другие валидаторы могут проверить, назначила ли доверенная функция самое короткое время победившему валидатору, а выигравший валидатор ждал ли указанное количество времени. Кроме того, другие валидаторы проверяют действительность блока до того, как он будет включен в реестр. Алгоритм помогает достичь огромной масштабируемости, поскольку ему не нужно решать сложную, требующую больших вычислительных ресурсов криптографическую головоломку. Кроме того, он может быть использован не только для разрешенного реестра, но и для публичного реестра.

Сравнивая структурные свойства трех алгоритмов, можем увидеть, что у алгоритмов IBFT RBFT базовый механизм – голосование, а для алгоритма PoET действующим базовым алгоритмом является механизм лотереи (см. табл. 3). По типам узлов можем отметить, что в алгоритме RBFT отсутствует как таковой валидатор, по сравнению с другими алгоритмами, а у алгоритма PoET также

еще принимает участие процессор, который определяет время ожидания узла, отталкиваясь от которого и происходит выбор узла-лидера. По типу структуры можем отметить, что у IBFT наблюдается динамический комитет, в то время как у остальных алгоритмов он статичный.

Таблица 2

Сравнение структурных свойств

	Структурные свойства		
Алгоритм	Типы узлов	Тип структуры	Базовый механизм
IBFT	Предлагающий лидер, Валидатор	Единый динамический комитет	Голосование
RBFT	Лидер, Клиент	Единый закрытый статический комитет	Голосование
PoET	Лидер, Валидатор, Процессор	Единый закрытый статический комитет	Лотерея

Сравнивая свойства безопасности, видим, что все алгоритмы имеют механизмы защит от атак Сибиллы, владеют свойством неотказуемости. Единственным различающимся параметром является сопротивляемость алгоритмов (см. табл. 3). У IBFT и RBFT сопротивляемость $3f + 1$, у PoET – $\Theta(\frac{\log\log n}{\log n})$.

Таблица 3

Сравнение свойств безопасности

	Свойства безопасности		
Алгоритм	Неотказуемость	Вектор атаки	
IBFT	Да, но не гарантируется работа в синхронной сети	Сопротивляемость $3f + 1$	Защита от атак Сибиллы с помощью алгоритма Proof of Authority (PoA)
RBFT	Да	Сопротивляемость $3f + 1$	Защита от атак Сибиллы
PoET	Да	Сопротивляемость $\Theta(\frac{\log\log n}{\log n})$	Защита от атак Сибиллы

Многокритериальный анализ алгоритмов

Среди свойств производительности можем сравнить между собой свойства отказоустойчивости – видим, что значения у алгоритмов различаются, а по алгоритму PoET нет данных вовсе. Пропускная способность у всех алгоритмов хорошая. Сравнивая масштабируемость, можем сделать вывод, что по этому показателю алгоритм RBFT проигрывает остальным алгоритмам, а если сравнить уровень задержки, видим, что поигрывает алгоритм PoET.

Таблица 4

Сравнение свойств производительности

	Свойства производительности				
Алгоритм	Отказоустой-чивость	Пропускная способность (транзакций в секунду)	Масштаби-руемость	Задержка	Сложность коммуника-ции
IBFT	$3f + 1$	400 ~ 1200	Хорошая	Низкая	$O(n^2)$
RBFT	$2f + 1$	500 ~ 1000	Средняя	Низкая	$O(n^4)$
PoET	-	470 ~ 1250	Хорошая	Средняя	$O(n^3)$

Таким образом, можем сделать вывод, что у каждого алгоритма есть свои особенности и каждый из них может быть наиболее подходящим для решения определенных задач.

Рассмотрим желательные свойства алгоритмов как критерии. Видно, что проанализированные алгоритмы принадлежат Парето-оптимальному множеству. Хотя алгоритм RBFT и обеспечивает меньшую пропускную способность в наилучшем случае, но в наихудшем случае его производительность выше.

Для многокритериального анализа альтернатив воспользуемся комбинированным методом выявления предпочтений, предложенным и описанным в общем виде в работе [7]. Данный метод реализован в форме веб-сервиса на портале ws-dss.com. Для апробации метода на задаче многокритериального анализа свойств алгоритмов консенсуса было задано дерево агрегирования критериев, указаны области предпочтений и веса критериев. Результаты работы по подготовке информации и ранжированию приведены на рисунке 5. Из рисунка видно, что при заданных предпочтениях наилучшим оказался алгоритм IBFT.

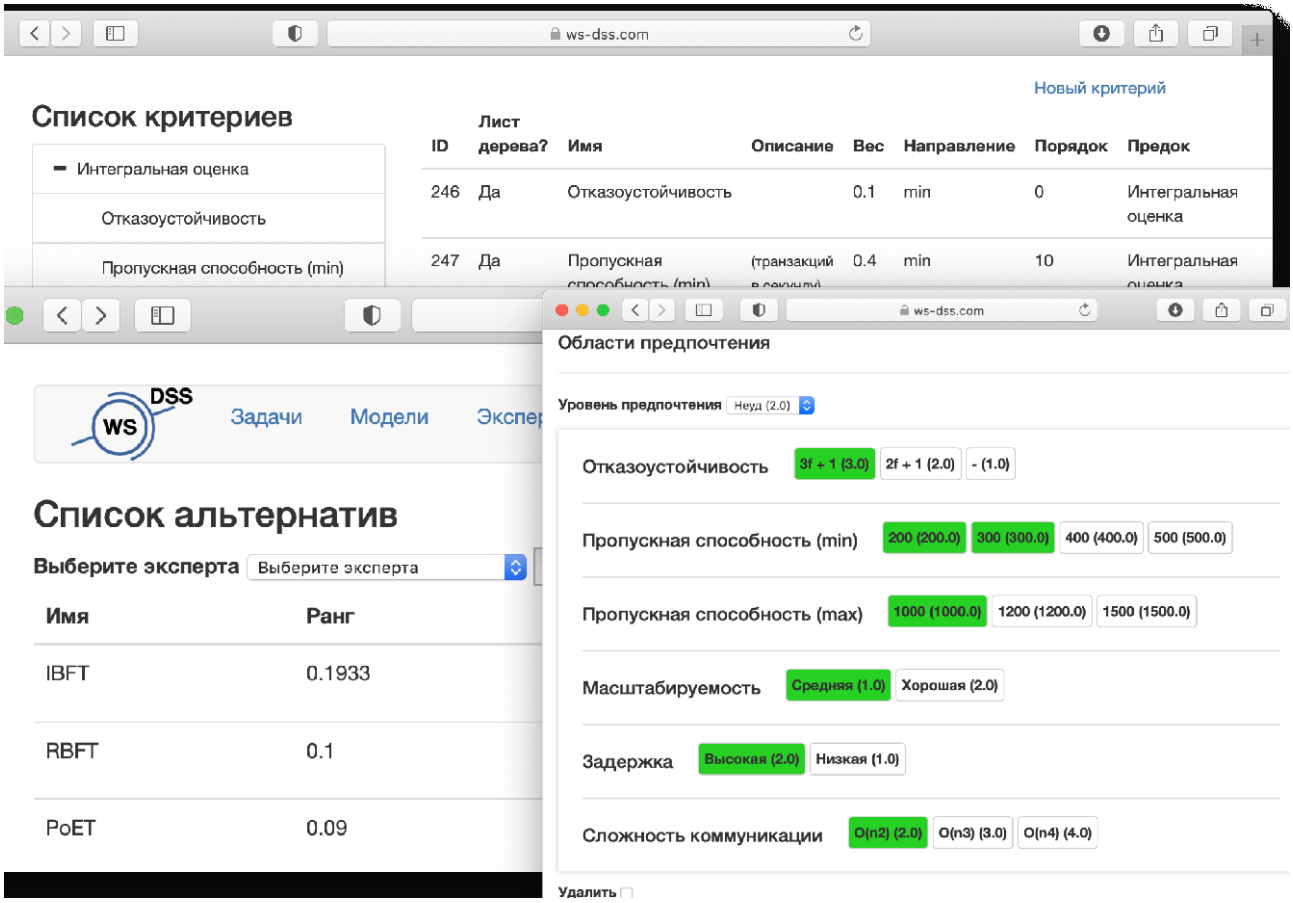

Рис. 5. Интерфейс ввода предпочтений и ранжирования алгоритмов консенсуса

Заключение

Анализ, проделанный в данной работе, вносит вклад в общую аналитическую деятельность, направленную на изучение применимости алгоритмов консенсуса. К ряду алгоритмов, анализ которых проводился по той же таксономии свойств, например таких, как PoET, Tendermint Burrow, YAC и RBFT, добавлен еще один алгоритм – Istanbul BFT.

Структурирование свойств алгоритмов позволяет сравнивать алгоритмы по их ключевым показателям и в наглядной форме предоставлять результаты этого анализа. Это может быть удобно в случае, когда нужно подобрать рациональный алгоритм, удовлетворяющий заранее заданному перечню требований. Данное исследование способствует развитию программного обеспечения для борьбы с пандемиями, так как чем больше будет проанализировано по предложенной структуре алгоритмов, тем шире будет выбор для разработчиков медицинских распределенных баз данных.

Библиографический список

1. Ahmad R.W., Salah K., Jayaraman R., Yaqoob I., Ellahham S., Omar M. Blockchain and COVID-19 Pandemic: Applications and Challenges. TechRxiv Preprint. – 2020. https://doi.org/10.36227/techrxiv.12936572.v1.

2. Ferdous M.S., Colman A. Blockchain Consensus Algorithms: A Survey. – 2020. URL: https://arxiv.org/abs/2001.07091
3. Bano S., Sonnino A., Al-Bassam M., Azouvi S., McCorry P., Meiklejohn S., Danezis G. SoK: Consensus in the Age of Blockchains. In Proceedings of the 1st ACM Conference on Advances in Financial Technologies (AFT '19). Association for Computing Machinery, NY, USA. – 2019. С. 183–198. URL: https://doi.org/10.1145/3318041.3355458.
4. Moniz H., The Istanbul BFT Consensus Algorithm – 2020. URL: https://arxiv.org/abs/2002.03613.
5. Istanbul Byzantine Fault Tolerance. URL: https://github.com/ethereum/EIPs/issues/650. (дата обращения: 15.04.2021).
6. Saltini R., Hyland-Wood D., Correctness Analysis of Istanbul Byzantine Fault Tolerance – 2019. URL: https://arxiv.org/abs/1901.07160v2
7. Осипов В.П., Судаков В.А. Комбинированный метод поддержки принятия многокритериальных решений // Препринты ИПМ им. М.В.Келдыша. 2015. No 30. 21 с. URL: http://library.keldysh.ru/preprint.asp?id=2015-30.

Оглавление