

# RISE: Уровень 2 Gigagas

*Самуэль Баттенелли*

*Хай Нгуен*

*Тхань Нгуен*

RISE Labs

## Аннотация

Мы представляем RISE, инновационную платформу второго уровня (Layer 2 или L2), разработанную для решения острых проблем производительности в экосистеме Ethereum rollup. Несмотря на значительные достижения, текущие решения Ethereum L2 отстают по пропускной способности транзакций, значительно уступая конкурентам, таким как Solana. RISE использует параллельную виртуальную машину Ethereum (EVM), непрерывный конвейер выполнения и новую архитектуру доступа к состоянию, построенную на базе Reth SDK, написанного на Rust, чтобы существенно улучшить пропускную способность и производительность. Основная цель RISE — достичь показателя The Surge (100 000 транзакций в секунду (TPS) и более 1 Gigagas в секунду), с потенциалом для дальнейшего масштабирования. Эта работа подробно описывает существующие проблемы технологий L2, архитектурные новшества RISE и направления будущей оптимизации масштабируемости и эффективности блокчейна. RISE обещает не только достичь уровня наиболее производительных решений уровня 1 (L1), но и установить новый стандарт в технологии блокчейна.

## 1 Введение

### 1.1 Технологический ландшафт

Когда Виталик Бутерин впервые запустил блокчейн Ethereum в июле 2015 года [1], он не только продвинул фундаментальные технологии, представленные Bitcoin [2], но и создал инновационную концепцию: смарт-контракты. Смарт-контракты открыли двери для приложений и глобальной координации, которые казались невозможными до их появления. Ясно, что мир начинает осознавать потенциал этой удивительной технологии. Однако широкое принятие выявило проблемы масштабирования, которые в конечном итоге привели к предложению Ethereum Endgame от Виталика Бутерина [3], которое эволюционировало в дорожную карту Ethereum [4]. Одним из ключевых компонентов дорожной карты является The Surge, целью которого является достижение 100 000 транзакций в секунду и более на rollup'ах. Arbitrum и Optimism сыграли значительную роль в прогрессе на пути к этой цели, продемонстрировав, что EVM rollup является неоспоримым решением для масштабирования. Однако их пропускная способность все еще недостаточна [5][6]. Технология готова ко второму поколению rollup'ов, которые будут

сосредоточены на производительности для устранения этого разрыва. Благодаря недавним инновациям, The Surge становится реальностью. Виталик утверждает, что мы приближаемся к экспоненциальному участку кривой возможностей [7]; RISE нацелен на разблокирование этих быстрых изменений и следующей волны приложений, которые пока даже не осознаются.

## 1.2 Введение

Экосистема rollup'ов Ethereum сделала значительный прогресс в улучшении масштабируемости блокчейнов, надежно поддерживая различные приложения и захватив значительную рыночную долю. Однако, даже при этих успехах, совокупная пропускная способность экосистемы второго уровня (L2) обычно обрабатывает около 100 транзакций в секунду (TPS)[8], что далеко от показателей конкурентов, таких как Solana, которая стабильно достигает более 1000 TPS, что в десять раз превышает показатели всех L2 вместе взятых [9]. Solana достигла этого, уделяя приоритетное внимание масштабируемости за счет децентрализации, с меньшим количеством валидаторов и высокими расходами на инфраструктуру. Принятие Solana показывает поддержку рынком прагматического подхода к масштабированию. Успех Solana и различия в производительности между Solana и L2 также ясно демонстрируют, что существует спрос на высокопроизводительные, недорогие решения L2 на базе EVM, но этот спрос пока не был удовлетворен существующими технологиями.

Мы представляем RISE — стек второго уровня Ethereum Virtual Machine (EVM) следующего поколения. Ключевые компоненты, разработанные с нуля, позволяют RISE максимально увеличить пропускную способность транзакций и достичь амбициозной цели в 100 000 транзакций в секунду (TPS), указанной в дорожной карте Ethereum. Доступность данных (Data Availability, DA) исторически была значительным узким местом для rollup'ов, но недавние достижения, такие как Celestia, EigenDA и EIP-4844, существенно снизили эту проблему. Теперь основное внимание переключилось на производительность выполнения транзакций. В борьбе за производительность выполнения транзакций RISE объединяет лучшие из существующих технологий и новые инновации, такие как параллельное выполнение EVM, непрерывное выполнение блоков и инновационный дизайн доступа к состоянию. Все это интегрировано с надежным Reth SDK на базе Rust.

Стек RISE ориентирован на будущее с централизацией вокруг rollup'ов; чтобы обеспечить гибкие развертывания, и Sequencing, и DA являются независимыми по дизайну, что позволяет развертываниям цепочек приложений RISE выбирать собственные решения Sequencing и DA. В основе стека RISE лежит высокопроизводительное выполнение. RISE mainnet будет поддерживать пропускную способность, превышающую самые производительные решения уровня 1 (L1), как и любая цепочка приложений RISE.

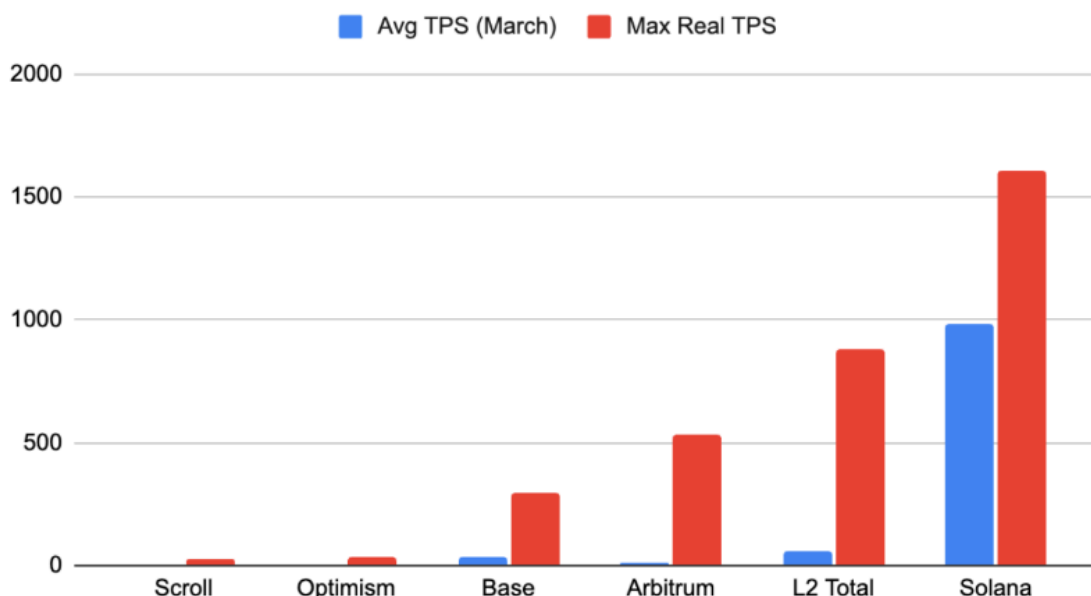
В этой работе представлены стек RISE и его инновации в производительности. Однако она не охватывает амбиции RISE по децентрализации с помощью основанной на данных

обработки и участия в нейтральной и модульной экосистеме; это будет рассмотрено в будущих работах команды RISE Labs.

### 1.3 Состояние rollup'ов

Подход Ethereum к масштабированию оказался успешным. L2 платформы суммарно достигли более 40 миллиардов долларов в виде заблокированной стоимости и с февраля 2024 года обрабатывают более 100 транзакций в секунду (TPS). С более чем 40 rollup'ами в активе, мы являемся свидетелями ранних дней горизонтального масштабирования Ethereum. Этот рост не был без трений: отсутствие общего состояния вызывает фрагментацию приложений, пользователей и ликвидности. Учитывая рост горизонтального масштабирования L2, можно справедливо заключить, что вертикальное масштабирование столкнулось с ограничениями и антинетворкинговыми эффектами, замедляющими принятие и активность отдельных rollup'ов

#### TPS Comparison



#### Сравнение производительности: Layer 2 и Solana

Эта проблема становится еще более очевидной, если сравнить принятие и производительность L2 с самым производительным конкурентом, Solana. Solana в среднем обрабатывала около 1000 TPS в марте 2024 года, что в десять раз превышает показатели всех Ethereum L2 вместе взятых. Arbitrum и Base достигли пиковых значений TPS в 532 и 293 соответственно. Однако обе сети испытали сбои при пиковых нагрузках, тогда как Solana стабильно поддерживает более 2000 TPS. Solana добилась значительных успехов на фоне EVM, что подтверждает вывод о том, что существует неудовлетворенный спрос на производительные решения L2.

## 1.4 Выбор виртуальной машины

Поддержка EVM очень мощна: EVM имеет больше активных продуктов, инвестиций, талантов и пользователей, чем любая другая виртуальная машина для смарт-контрактов. Создание технологии для поддержки 100 000 TPS имеет смысл только при наличии поддержки как разработчиков, так и пользователей. Однако есть аспекты в дизайне EVM, которые значительно усложняют масштабирование, но мощь EVM слишком велика, чтобы с ней соперничать, поэтому RISE сосредоточен исключительно на совместимом с EVM стеке второго уровня (L2).

## 1.5 Zero Knowledge против Optimistic

zkEVM L2 достигли значительного прогресса в последнее время, их принятие на рекордно высоком уровне, и скорость доказательства улучшается. RISE стремится ускорить The Surge и достичь цели в 100 000 TPS. Это станет возможным в будущем с использованием zkEVM, но для этого потребуются аппаратные решения, такие как zkASIC, которые пока недоступны. С оптимистичным rollup'ом мы уверены, что высокая пропускная способность может быть достигнута без аппаратного ускорения. Поэтому команда RISE поначалу сосредоточится на Optimistic стеке и перейдет на zkEVM, как только соответствующее оборудование станет доступным.

## 1.6 Структура документа

В этом документе мы начнем с раздела 2, где обозначим проблемы производительности при построении высокопроизводительного стека rollup'ов; в разделе 3 мы подробно рассмотрим стек технологий RISE, а завершим в разделе 4 обсуждением будущих работ и заключительными замечаниями.

## 2 Проблемы производительности

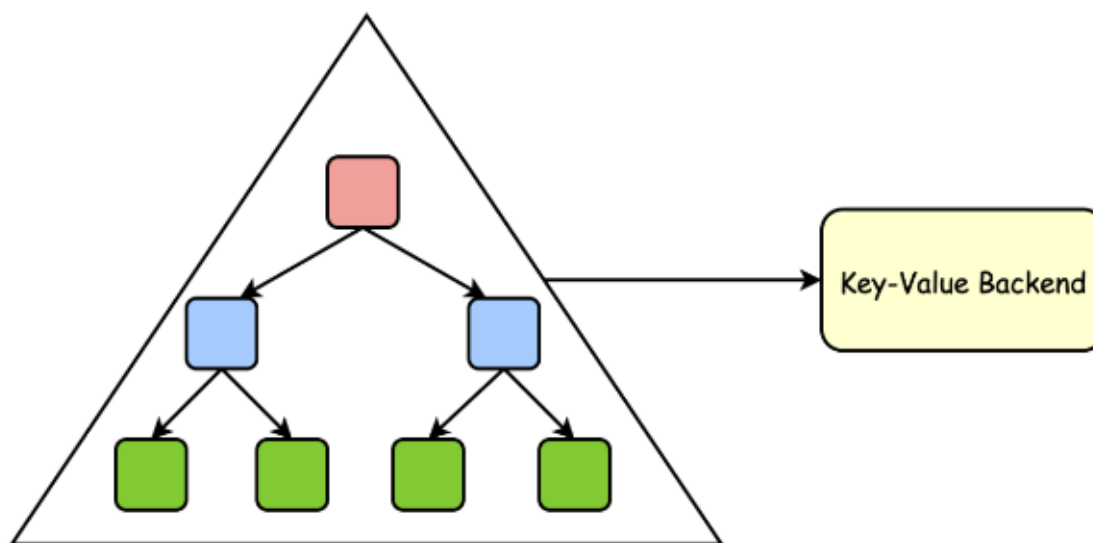
Основной проблемой производительности L2 решений до настоящего времени была доступность данных (Data Availability, DA). Однако благодаря инновациям в области внешней DA, таким как Celestia и eigenDA, а также встроенным решениям, таким как EIP-4844 [8], это больше не является узким местом. Теперь перед нами встают другие проблемы, такие как сеть, выполнение, ограничения размера блоков, ограничения на доказательства мошенничества (fraud-proof), меркелизация и ввод-вывод (I/O) в хранилище. Буквально за одну ночь производительность L2 стала захватывающей и сложной областью проектирования с различными направлениями.

### 2.1 Выполнение EVM

Большинство реализаций выполнения EVM работают в однопоточном режиме. Из-за этого сегодня верхний предел производительности определяется тактовой частотой одного центрального процессора (CPU). Переход к параллельной модели выполнения также не является тривиальной задачей, поскольку транзакции в блоке по своей природе последовательны.

## 2.2 Доступ к состоянию Ethereum

Ethereum требует аутентифицированных структур данных, закодированных как модифицированные Merkle-Patricia Tries (MPT), для хранения и аутентификации ключево-значимых данных.



*Рисунок: Ключевые данные в Ethereum кодируются в MPT, который в конечном итоге хранится в базе данных на основе ключево-значимого хранилища. Хранилище обычно реализуется как структура данных типа B-Tree или LSM-Tree.*

К сожалению, доступ к хранилищу в Ethereum медленный по нескольким причинам:

- Ключево-значимые данные не хранятся напрямую в базе данных; они кодируются в MPT, который затем сохраняется в базе данных. Поэтому операция чтения/записи ключа на уровне приложения приводит к множественным операциям чтения/записи на уровне базы данных. Это многократное чтение/запись приводит к низкой скорости доступа к хранилищу, что, в свою очередь, снижает пропускную способность выполнения.
- Операция записи требует пересчета хешей всех внутренних узлов вдоль пути MPT.
- И, наконец, самое важное: ввод-вывод в хранилище синхронен. Поток выполнения должен ждать завершения дорогостоящих операций чтения/записи, прежде чем приступить к обработке следующей транзакции.

Влияние этого значительно. Увеличение состояния приводит к антинетворковому эффекту. По мере роста состояния увеличивается и средняя глубина дерева, что еще больше замедляет доступ к хранилищу. Кроме того, корень MPT необходимо пересчитывать для каждого блока, в результате чего значительная часть времени выполнения уходит на меркелизацию.

### **2.3 Ввод-вывод в хранилище**

Хранение состояния не является явно указанной частью дизайна EVM; клиенты узлов могут внедрять свои решения для хранилищ. LevelDB, PebbleDB и MDBX — это все хранилища типа "ключ-значение", используемые в клиентах Ethereum и Layer 2, хотя они не являются нативными аутентифицированными хранилищами (AKVS). Нативное AKVS решение имело бы свои преимущества, хотя на сегодняшний день не существует подходящих решений с открытым исходным кодом.

### **2.4 Размер блока**

Очевидным ограничением Layer 2 является предел газа блока. Это ограничение можно легко увеличить, если система может поддерживать более высокую пропускную способность. Однако увеличение предела блока может повлиять на систему доказательств мошенничества (fraud-proof). Если оспаривающие стороны должны отправить весь блок для доказательства мошенничества, они рискуют столкнуться с ограничениями по размеру блоков Ethereum.

### **2.5 Производство блоков Layer 2**

Все клиенты L2 начинали как клиенты Ethereum и часто сохраняют концепции уровня 1 (L1). Процесс создания блока для L2 по своей природе отличается от L1. Это открывает возможности для оптимизации и параллелизации процесса создания блоков в L2-секвенсоре и клиенте выполнения.

### **2.6 Взаимодействие и децентрализация**

Взаимодействие через общий секвенсор станет важным шагом для L2, а децентрализация секвенсора также является важным этапом. Каждое из этих изменений вводит ограничения на выполнение и секвенсор, которые необходимо учитывать при создании системы, рассчитанной на будущее.

## **3 Дизайн стека RISE**

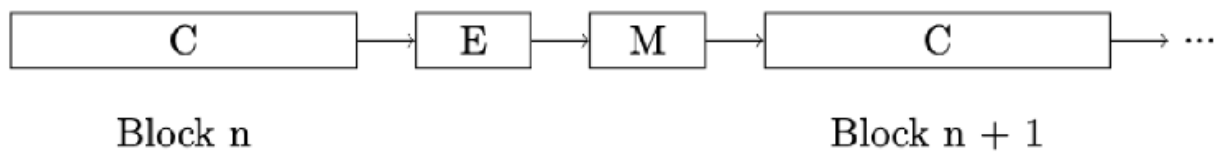
Стек RISE основывается на OP Stack с Reth в качестве клиента выполнения. Это позволяет нам начать с надежного фундамента и сосредоточиться на производительности.

### 3.1 Непрерывный конвейер блоков

Производство блоков L2 включает следующие последовательные этапы [13]:

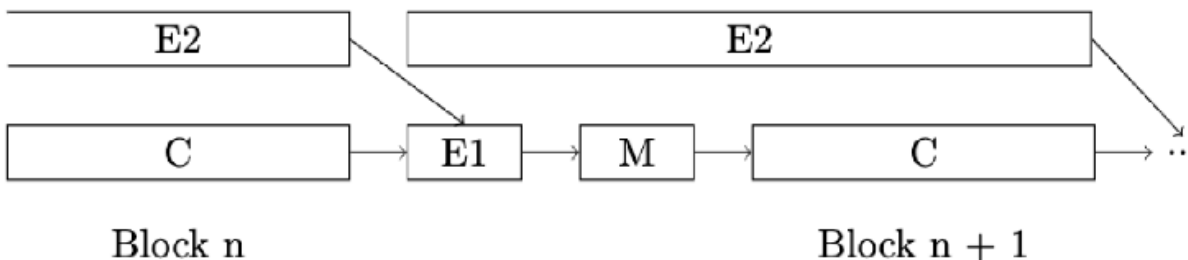
1. С: Консенсус, извлекающий транзакции L1 и новые атрибуты блока
2. Е: Выполнение извлеченных транзакций L1 (1) и транзакций L2 из mempool (2)
3. М: Меркелизация состояния цепочки и результатов выполнения для завершения заголовка блока

Обычно эти этапы последовательны; рисунок ниже иллюстрирует типичный конвейер блоков для L2 с односекундным временем блока и большим состоянием. Предел газа блока будет ограничен тем, сколько газа может быть обработано за время, выделенное на выполнение; однако в этой системе выполнение отводится только на меньшую часть времени построения блока. На практике, С занимает 40-80% времени небольшого блока, а по мере роста состояния М занимает до 60% оставшегося времени выполнения (12-36% общего времени блока). В худшем случае Е получит только 8% времени блока для выполнения, что очень неэффективно, и предельные значения газа блока должны учитывать худший сценарий, чтобы избежать дрейфа времени блока и реорганизаций.



\*Последовательный конвейер блоков\*

RISE вводит непрерывный конвейер блоков (CBP), параллелизованный конвейер блоков с одновременными этапами и непрерывным потоком выполнения (CE). Поток CE отслеживает Mempool для транзакций L2 и выполняет их в нескольких сегментах блока, больше не дожидаясь консенсуса для запроса нового блока. Вспомним худший сценарий с 8% выполнения; для сравнения, CBP выполняет транзакции почти 100% времени. Меркелизация также происходит одновременно с выполнением.



*Непрерывный конвейер блоков (CBP)*

### 3.2 Параллельный EVM

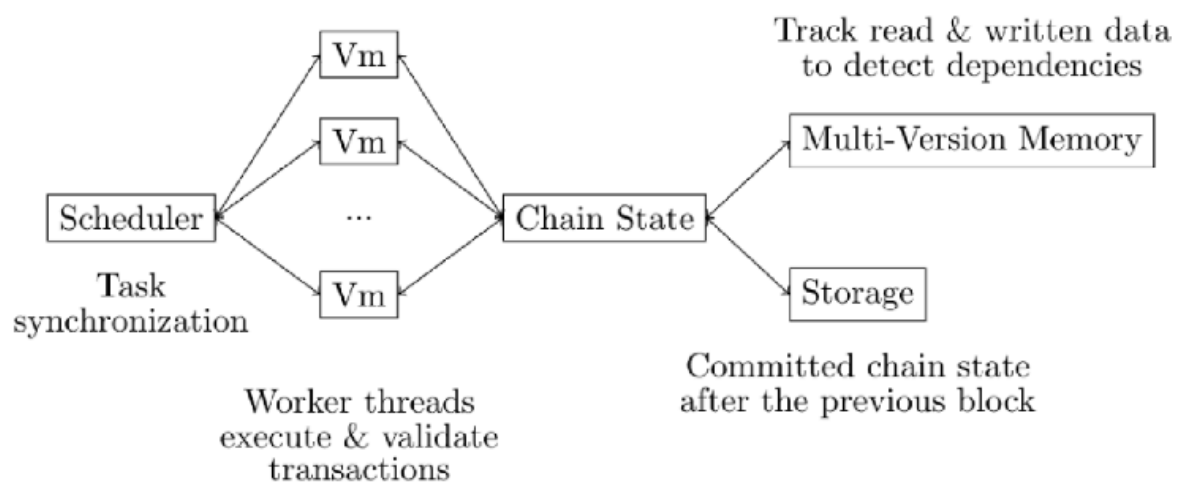
Популярность параллельного выполнения для увеличения пропускной способности блокчейна растет благодаря проектам Aptos, Sui и Solana. Однако до сих пор это не давало больших результатов в среде EVM. Ранние адаптации Block-STM[13] от Polygon и Sei показали ограниченное ускорение из-за отсутствия оптимизаций, специфичных для EVM, и ограничений в реализации. Оба проекта используют язык Go, который использует сборку мусора и не подходит для оптимизаций на микросекундном уровне. Параллельное выполнение на Go обычно медленнее, чем последовательное выполнение на Rust и C++ для блоков Ethereum mainnet.

RISE pevm решает эту проблему, разрабатывая специализированный параллельный исполнитель для EVM и реализуя его на Rust, чтобы минимизировать накладные расходы времени выполнения. Результат — самый быстрый блокочный исполнитель EVM с максимальным ускорением в 22 раза и исходной пропускной способностью выполнения 55 Gigagas/c на 32 CPU AWS Graviton3. RISE pevm также позволяет разрабатывать новые параллельные приложения для EVM, такие как шардированный автоматический маркет-мейкер (Sharded AMM) [17].

## Дизайн

Выполнение блокчейна должно быть детерминированным, чтобы участники сети соглашались с блоками и переходами состояния. Поэтому параллельное выполнение должно давать тот же результат, что и последовательное выполнение. Наличие условий гонки, влияющих на результаты выполнения, нарушило бы консенсус.

RISE pevm базируется на оптимистическом выполнении Block-STM. Мы также используем совместный планировщик и многоверсионную структуру данных для обнаружения конфликтов состояния и повторного выполнения транзакций, если это необходимо.



Мы внесли несколько улучшений, тонко настроенных для EVM. Например, все транзакции EVM в одном блоке читают и записывают на счет получателя для оплаты газа, что по



умолчанию делает все транзакции зависимыми. RISE pevm решает эту проблему, откладывая обновление баланса получателя. Мы имитируем баланс при чтении оплаты газа, чтобы избежать регистрации зависимости от состояния, и оцениваем его только в конце блока или при явном чтении. Мы применяем ту же технику к типичным сценариям, таким как прямые переводы ETH и переводы ERC-20. Отложенные обновления помогают нам параллелизировать переводы с одного и на один и тот же адрес, при этом добавляется лишь незначительная задержка на постобработку для оценки отложенных значений.

В то время как другие интегрируют свои параллельные исполнители прямо в узлы, мы создали отдельный репозиторий, который служит площадкой для дальнейших исследований и разработок pevm. Исходный код открыт и доступен по адресу: <https://github.com/risechain/pevm>.

### *3.2.1 Предобработка Metapool*

В отличие от предыдущих collup'ов, которые упорядочивали транзакции по принципу "первый пришел — первый обслужен" или через аукционы газа, RISE вводит новую структуру Metapool, которая балансирует между задержкой и пропускной способностью. Цель состоит в том, чтобы заранее упорядочить транзакции таким образом, чтобы минимизировать количество общих состояний и максимизировать параллельное выполнение. Это имеет сходный эффект с локальным рынком комиссий Solana, где загруженные контракты и состояния обходятся дороже с точки зрения газа и задержки. Поскольку количество потоков для выполнения транзакций намного меньше, чем предполагаемый TPS, мы можем все еще выделить потоки для последовательного выполнения высоконагруженных контрактов, в то время как другие транзакции будут выполняться параллельно в других потоках.

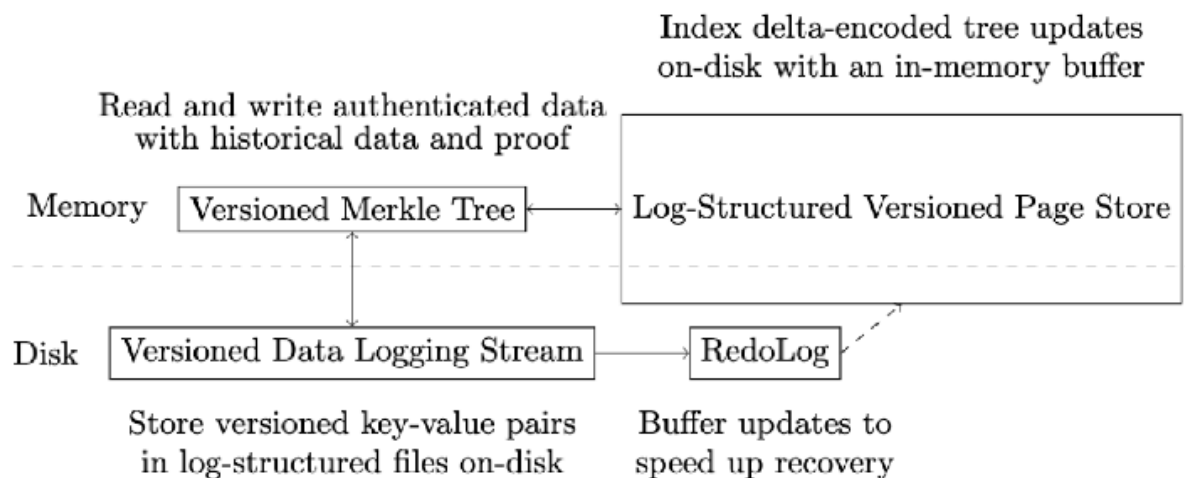
### **3.3 Версионное Мерклово дерево и RiseDB**

Помните, что доступ к состоянию Ethereum существенно влияет на пропускную способность; наше начальное решение заключается в том, чтобы хранить все MPT в памяти, чтобы уменьшить накладные расходы на чтение и запись. Однако по мере того как цепочка обрабатывает сотни тысяч транзакций в секунду, состояние может стать огромным. По мере его роста хранение всего MPT в памяти становится нецелесообразным для полных узлов.

Вместо этого мы заменяем MPT Ethereum на гораздо более эффективное с точки зрения хранения Версионное Мерклово дерево. Версионные Меркловы деревья используют версионный ключ узла, поэтому все ключи сортируются при вставке. Это минимизирует стоимость сжатия данных в подлежащем хранилище "ключ-значение". Версионные деревья также позволяют эффективно и надежно выполнять исторические запросы и проверку данных.

Популярным версионным Меркловым деревом является Jellyfish Merkle Tree[14], разработанное для Aptos. Хотя Jellyfish минимизирует стоимость сжатия данных в хранилище "ключ-значение" с помощью версионных ключей узлов, оно по-прежнему использует универсальный RocksDB, который имеет высокую степень многократной записи. Сжатие также запускается во время обрезки.

Для решения этой проблемы мы применяем подход LETUS[15], заменяя двухслойную архитектуру хранения на упрощенную. Обновления дерева кодируются в виде дельт, а пары ключ-значение хранятся в файлах с лог-структурой (а не в деревьях), что минимизирует накладные расходы на чтение и запись.



Наконец, мы экспериментируем с идеями из NOMT[16], снижая радиус Мерклогового дерева, чтобы уменьшить сложность меркелизации. Что касается увеличенной сложности чтения, мы используем оптимальное представление на диске, чтобы параллельно извлекать все поддеревья для одного ключа за одну операцию чтения с SSD. NOMT может использовать бинарную систему, так как каждый узел занимает только 32 байта. Это не относится к версионным Меркловым деревьям, так как они должны хранить дополнительные версии. Нам также нужно хранить смещение и размер значения для индексации значения в файлах с лог-структурой. Окончательный дизайн может использовать 8-арное дерево для балансировки между многократной записью и чтением.

### 3.4 Дополнительные улучшения

#### 3.4.1 Сеть

Метропол секвенсора будет подключен к тысячам узлов, и каждое соединение будет иметь накладные расходы. Сеть RISE использует протокол QUIC, альтернативу традиционному TCP, который предлагает более быстрое установление соединений и сниженные задержки, среди других функций.

### *\*3.4.2 Компиляция JIT*

Just-in-time компиляция позволяет выполнять виртуальные машины напрямую, а не через интерпретатор. RISE revm является независимым от исполнителя EVM и может переключаться между JIT-компилятором, например <https://github.com/paradigmxyz/revmc>, и интерпретатором, таким как <https://github.com/bluealloy/revm>, в зависимости от блока и среды развертывания.

<https://github.com/paradigmxyz/revmc>

<https://github.com/bluealloy/revm>

## **4 Заключение**

Разработка RISE является значительным шагом вперед в эволюции масштабируемости и производительности блокчейнов. В этой белой книге были исследованы присущие существующим технологиям второго уровня в экосистеме Ethereum проблемы и предложено решение RISE как трансформационное решение, предназначенное для преодоления этих ограничений. Реализуя параллельный EVM, усовершенствованные процессы выполнения и инновационные решения для баз данных, RISE нацелен на амбициозную пропускную способность в 100 000 TPS, устанавливая новый стандарт в технологии блокчейнов и вступая в эпоху Gigagas.

Развитие RISE нацелено на удовлетворение текущих потребностей и предвосхищение будущих вызовов и инноваций в блокчейн-пространстве. Мы приглашаем сообщество присоединиться к нам в этом увлекательном путешествии, направленном на создание более масштабируемой, эффективной и инклюзивной блокчейн-экосистемы.

## **Благодарности**

Мы выражаем благодарность тем, кто проложил путь для достижений в масштабируемости Ethereum и rollup-технологиях. Мы признаем фундаментальную работу и значительный вклад команд Arbitrum и Optimism. Их новаторские усилия в области оптимистичных rollup'ов доказали, что rollup'ы являются жизнеспособным вариантом масштабирования для Ethereum.

Мы также благодарим команду Reth за создание клиента выполнения и SDK, на которых строится RISE. Модульность и производительность Reth имеют важное значение для RISE, и команда является дальновидными лидерами. Мы с нетерпением ждем возможности отблагодарить их, сделав значимый вклад в проект Reth.

## Ссылки

1. Buterin, V. (2014). Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. Retrieved from [https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum\\_Whitepaper\\_-\\_Buterin\\_2014.pdf](https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf)
2. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from <https://bitcoin.org/bitcoin.pdf>
3. Buterin, V. (2021, December 6). Endgame. Retrieved from <https://vitalik.eth.limo/general/2021/12/06/endgame.html>
4. Understanding the Ethereum Development Roadmap. Retrieved April 10, 2024, from <https://www.blocknative.com/blog/ethereum-roadmap-guide>
5. Kalodner, H., Goldfeder, S., Chen, X., Weinberg, S. M., & Felten, E. W. (2018). Arbitrum: Scalable, Private Smart Contracts. Princeton University. Retrieved from <https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-kalodner.pdf>
6. TPS, Max TPS, TTF, Block Time & Governance Model. (2024). Retrieved April 14, 2024, from <https://chainspect.app/dashboard?range=30d>
7. Buterin, V. [@VitalikButerin]. (2024, March 29). What does this mean more broadly? I argue that now that the merge and EIP-4844 are done, we are decidedly on the right side of the S-curve. Further L1 changes will be quite meaningful and significant, but relatively milder. Retrieved from <https://x.com/VitalikButerin/status/1773356271300706507>
8. Value Locked. (2024). Retrieved April 10, 2024, from <https://l2beat.com/scaling/summary>
9. Buterin, V., & Beiko, T. (2024). EIP-4844: Shard Blob Transactions. Ethereum Improvement Proposals. Retrieved April 2, 2024, from <https://eips.ethereum.org/EIPS/eip-4844>
10. Electric Capital. (2023). 2023 Crypto Developer Report.
11. Choi, J., Beillahi, S., Singh, S., Michalopoulos, P., Li, P., Veneris, A., & Long, F. (2022). LMPT: A Novel Authenticated Data Structure to Eliminate Storage Bottlenecks for High-Performance Blockchains. IEEE.
12. Ethereum Optimism. (2024). Optimism: Optimistic Ethereum. Retrieved from <https://github.com/ethereum-optimism/optimism>
13. Gelashvili, R., Spiegelman, A., Xiang, Z., Danezis, G., Li, Z., Malkhi, D., Xia, Y., & Zhou, R. (2022). Block-STM: Scaling Blockchain Execution by Turning Ordering Curse to a Performance Blessing. arXiv preprint arXiv:2203.06871. <https://doi.org/10.48550/arXiv.2203.06871>
14. Zhenhuan Gao, Yuxuan Hu, Qinfan Wu. (2021). Jellyfish Merkle Tree. Retrieved from <https://developers.diem.com/papers/jellyfish-merkle-tree/2021-01-14.pdf>
15. Shikun Tian, Zhonghao Lu, Haizhen Zhuo, Xiaojing Tang, Peiyi Hong, Shenglong Chen, Dayi Yang, Ying Yan, Zhiyong Jiang, Hui Zhang, and Guofei Jiang. (2024). LETUS: A Log-Structured Efficient Trusted Universal Blockchain Storage. <https://doi.org/10.1145/3626246.3653390>
16. Preston Evans. (2024). Nearly Optimal State Merklization. Retrieved from [https://sovereign.mirror.xyz/jfx\\_cJ\\_15saejG9ZuQWjnGnG-NfahbazQH98i1J3NN8](https://sovereign.mirror.xyz/jfx_cJ_15saejG9ZuQWjnGnG-NfahbazQH98i1J3NN8)
17. Hongyin Chen, Amit Vaisman, Ittay Eyal. (2024). SAMM: Sharded Automated Market Maker. Retrieved from <https://arxiv.org/abs/2406.05568>

