Rotura Protocol Whitepaper

Rotura Protocol utilizes a high-performance architecture with horizontal scaling to provide developers with a data-driven development framework, helping users monetize their data

larryliu@RoturaLabs.com

Abstract

Blockchain technology needs breakthrough. But present blockchain architectures all suffer from a number of issues not least practical means of performance and scalability, which prevents billions of users from adopting cryptos.

In this paper, we describe a novel approach to utilize scale out architecture to grow blockchain. The focus of this paper is to formalize the scale out architecture based on protocol special data structure, and prove that this approach increases throughput without compromising security and decentralization.

1. Preface

This is intended to be a technical vision summary of one possible direction that may be taken in further developing the blockchain paradigm. It lays out in as much detail as possible at this stage of development a system which may give concrete improvement on a number of aspects of blockchain technology.

It is not intended to be a specification, formal or otherwise. It is not intended to be comprehensive nor to be a final design. It is not intended to cover non-core aspects of the framework such as APIs, bindings, languages and usage. Mechanisms will be added, refined and removed in response to community ideas and critiques. Large portions of this paper will likely be revised as experimental evidence and prototyping gives us information about what will work and what not.

This document includes a core description of the protocol together with ideas for directions that may be taken to improve various aspects. It is envisioned that the core description will be used as the starting point for an initial series of proofs-of-concept. A final "version 1.0" would be based around this refined protocol together with the

additional ideas that become proven and are determined to be required for the project to reach its goals.

1.1 History

- December, 2021： 0.1
- Febuery, 2022： 0.2
- April, 2022: 0.3
- July, 2022: 0.4
- August, 2022: 0.5
- September, 2022: 0.6
- Dec, 2022: 0.7
- April, 2023: 0.8
- July, 2023: 0.9

2. Motivation

The world is undergoing massive change while industry is shifting to the Crypto Economy. It's an economy powered by the blockchain and other 21st century technologies — the cloud, mobile, distributed system, cryptography and big data. At the heart of every Digital Crypto Economy business are its web, mobile, and Internet of Things (IoT) applications: they're the primary way companies interact with customers, and how companies run more and more of their business to move their assets to blockchain. The experiences that companies deliver via those apps largely determine how satisfied — and how loyal — customers will be.

Today's web, mobile, and IoT applications share one or more (if not all) of the following characteristics. They need to:

- Support large numbers of concurrent users (tens of thousands, perhaps millions)
- Deliver highly responsive experiences to a globally distributed base of users
- Be always available — no downtime
- Rapidly adapt to changing requirements with frequent updates and new features

Building and running these web, mobile, and IoT applications has created a new set of technology requirements. The new blockchain technology architecture needs to be far more agile than ever before, and requires an approach to real-time data management that can accommodate unprecedented levels of scale, speed, and data variability.

Blockchains have demonstrated great promise of utility over several fields including "Internet of Things" (IoT), finance, governance, identity management, web decentralisation and asset-tracking. However despite the technological promise and grand talk, we have yet to see signicant real-world deployment of present technology. We believe that this is down to the low performance and scalability.

Throughput limitations of existing blockchain architectures are well documented and are one of the most significant hurdles for their wide-spread adoption. Attempts to address this challenge include layer-2 solutions, such as Bitcoin's Lightning or Ethereum's Plasma network, that move work off the main chain. Another prominent technique is sharding, i.e., breaking the network into many interconnected networks. However, these scaling approaches significantly increase the complexity of the programming model by breaking ACID guarantees (Atomicity, Consistency, Isolation, and Durability), increasing the cost and time for application development.

This applies equally to both proof-of-work (PoW) systems such as Bitcoin and Ethereum and proofof-stake (PoS) systems: all ultimately suffer from the same handicap. It is a simple strategy that helped make blockchains a success. However, by tightly coupling these two mechanisms into a single unit of the protocol, we also bundle together multiple different actors and applications with different risk proles, different scalability requirements and different privacy needs. One size does not fit all. Too often it is the case that in a desire for broad appeal, a network adopts a degree of conservatism which results in a lowest-common-denominator optimally serving few and ultimately leading to a failing in the ability to innovate, perform and adapt, sometimes dramatically so.

It seems clear, therefore, that one reasonable direction to explore as a route to a scalable decentralised compute platform is to decouple the consensus architecture from the state-transition mechanism. And, perhaps unsurprisingly, this is the strategy that Rotura adopts as a solution to performance and scalability.

2.1 Importance and value of blockchain performance

In light of the popularity of DeFi and the growth of DEXes on Ethereum, users and developers were limited by high gas costs and slow transactions. Users are expecting DEXes and DeFi to have a centralized exchange experience but DEX's security.

The blockchain technology is the bottleneck from driving the global massive adoption of DeFi. In order for DeFi to grow to a billion users, blockchain needs to solve these problems, and improve issues of centralization, capital inefficiency, and liquidity segmentation.

How valuable would that be? Let's just do some really rough calculations.

1. What would NYSE + ARCA + CME + ICE + … make if they were decentralized and charged 0.25bps on each trade? Well, daily trading volume is around $10T. If 0.25bp fees reduced volume by 75% and otherwise 20% of the volume moved to a decentralized venue, the resulting volume would be about $500B/day.That would result in $3B of annual revenue and a valuation likely around $100B.
2. In social media, Say that gas costs of $0.00002, *Twitter:*Daily cost of $10k, *Facebook:*Daily cost of $100k
3. How about credit cards? VISA handles ~2k-50k TPS; scalable blockchains can handle this and has a $500b valuation. So we could have $1T valuation of credit card companies on-chain.

2.2 Performance Analysis

The slow transaction speed of public blockchains has always been criticized by everyone. In recent years, several mainstream public blockchains have experienced congestion. Now many projects advertise themselves as one million TPS, thus solving the problem of insufficient public blockchain performance. But is the blockchain performance problem really solved? Let's analyze it together to see where the bottleneck of the performance of the public blockchain is.

Blockchain technology was first born around 2008, when the distributed systems just appeared such as Hadoop, MongoDB etc. The architecture of blockchain is distributed. But it doesn't perform at scale given the distributed system is not mature at that time. So the blockchain was designed not to be high performance at the beginning. And it is not practical now.

2.2.1 Traditional POW consensus

As the originator of blockchain technology, Bitcoin's performance has long been unable to meet the current transaction needs, so congestion often occurs. Let's first look at the accounting process of Bitcoin:

Bitcoin adopts POW consensus (Proof of Work), and there are two main steps that take a lot of time. The first is to do proof of work, that is, to calculate the hash value of the block that meets the requirements. The second is the process of network-wide broadcasting, waiting for other miners to confirm the block.

At the time of design, the block time of Bitcoin is set to 10 minutes (that is, 10 minutes of proof-of-work is required), the size of each block is about 1MB, and a block can pack up to several thousand transaction data. Although a single block can contain thousands of transaction information, it takes at least 10 minutes to complete a block, which leads to low transaction information recorded per unit time, that is, low TPS.

TPS (transaction per second) is a commonly used indicator to describe the performance of the blockchain, which refers to the number of transactions processed per second, and is an important parameter to measure the throughput of a system.

Due to the slow block generation, the TPS of Bitcoin is only about 7. Therefore, Bitcoin's congestion is often blamed on the POW consensus mechanism it uses.

Some people will say that TPS can be increased by increasing the size of the block or increasing the speed of the block, so that there will be no congestion. It's a nice idea, but it's not actually that simple to increase transaction throughput. Increasing the block size will lead to an increase in block confirmation time (longer broadcast time), because a single block contains more bytes, and the required transmission time will be correspondingly longer; although increasing the block speed will The proof-of-work time is reduced, but when the block production speed is too fast and the broadcast speed

cannot keep up, a new block will be mined before a block is confirmed by the miners of the whole network, which will cause a fork. Forks reduce the security of the network.

The cost of doing evil without forking is not less than 51% of the computing power of the entire network. Forks reduce the cost of doing evil and make the network insecure. Therefore, the 10–minute block time is a compromise between Bitcoin's security and efficiency.

Therefore, under POW, it is not possible to arbitrarily increase the block capacity or increase the block generation speed. This increases the possibility of forks and reduces the security of the network.

2.2.2 POS consensus improves transaction speed

Due to the slow transaction speed and high energy consumption of POW, POS consensus (Proof of Stake) is gradually being more widely adopted. For example, an important step in Ethereum 2.0 is to switch from POW consensus to POS consensus. Different from POW, POS obtains the right to package blocks and vote by staking Token.

In POS, the blocks produced by miners are directly candidate blocks, eliminating the proof–of–work step in POW. Because the right to package is assigned to miners by default when the tokens are pledged to become miners. POS allocates the packaging right before the block is produced, and POW has to compete for the packaging right after the block is produced. Therefore, POS will be more efficient than POW, and it can even be confirmed in seconds. It can be seen that the transaction performance of POS has been significantly improved compared to POW.

Under the existing POW consensus of Ethereum, the TPS is about 30 to 40. Although much higher than Bitcoin, it is still congested from time to time, and transfers sometimes take hours to complete. In order to solve this problem, Ethereum chose to abandon POW and adopt a more efficient POS mechanism. ETH2.0's phase0 beacon chain (Bacon Chain) is launched, and POS will gradually replace POW. At that time, only 32 ETHs need to be pledged to become a miner.

2.2.3 Higher speed DPOS

EOS's DPOS consensus belongs to POS, and token holders elect some representatives, and these representatives take turns to keep accounts. The essence of DPOS is to sacrifice decentralization to obtain high throughput. The EOS network has only 21 super nodes to produce blocks.

Compared with the hundreds or thousands of nodes of POS, EOS only needs to be broadcast in 21 nodes, so the time required for Time2 will be further shortened, and the TPS will also be greatly improved.

EOS claims in the white paper that it has millions of TPS, but there was still congestion during the EIDOS airdrop in 2019. Although the consensus algorithm has been improved to obtain a super high TPS, why does even an airdrop project cause congestion? It can be seen that there are other factors that restrict the performance of the blockchain.

The performance of a blockchain network is not only related to TPS, that is, throughput; it is also related to the performance of a single full node. There are roughly two types of common nodes in a blockchain network: full nodes and light nodes. Each full node in the network is responsible for broadcasting blocks and confirming transactions. Therefore, the full node must save the transaction history of the entire blockchain, as well as the status of all users and DAPPs in the entire network to verify this information during transactions. The light node is only responsible for querying the information of the full node, and does not participate in the accounting process, so the light node will not make any contribution to the network performance. When executing a transaction or running a DAPP, it will rely on the full node to complete, so the performance of the full node determines the processing speed of transaction execution. However, the performance of the blockchain network is not the superposition of the performance of all full nodes, but there is a barrel effect. The water storage capacity of the barrel depends on the shortest board in the barrel, and the performance of the entire network is also subject to the performance of a single full node: the performance of the entire network will not be higher than that of a single full node.

The congestion of EOS is limited by the performance of a single full node. EOS divides the performance of the full node into RAM, CPU and other resources. These resources are then tokenized, allowing users in the network to maximize the utilization of full nodes by leasing. When a large number of processing requests occur in a short period of time, the performance resources of the full node will be quickly exhausted, just like a computer running too many programs, and then opening a web page will be very stuck. The congestion caused by the EIDOS airdrop in 2019 was officially put on hold due to the rapid shortage of CPU resources in the EOS network, and the CPU required for a transfer exceeded the CPU that the network could allocate to it.

2.2.4 Where is the bottleneck of public blockchain performance?

As can be seen from the above examples, changing the consensus algorithm can improve throughput and increase TPS to a certain extent. But no matter how the consensus algorithm changes, this step cannot be escaped: the entire network is fully broadcast, and the block is confirmed after the node has verified it. The processing efficiency of this process is mainly related to the transmission efficiency of broadcast and the ability of nodes to process information.

The higher the transmission efficiency of the broadcast, the greater the throughput of the network and the higher the TPS. The physical upper limit of throughput is the network bandwidth between each full node, and the network bandwidth is limited by communication technology and hardware devices. Many projects are now advocating tens of millions of TPS, which requires a very high network bandwidth, and it is possible

to trap all full nodes in a local area network, but such a network has almost no decentralization at all.

The better the performance of a single full node, the higher its ability to process information, and the stronger the performance of the public blockchain. Blindly increasing the requirements for full nodes will also lead to centralization. For example, only some high-configuration servers can be full nodes, and these nodes may be controlled by some large companies.

Taken together, the performance of the blockchain is mainly limited by the network bandwidth and the performance of a single full node. Nowadays, many projects only blindly pursue high TPS, which makes people mistakenly think that this improves the performance of the public blockchain. In fact, they only focus on network bandwidth and ignore the impact of full node performance on the overall network performance. The key to improving performance for public blockchains is to ensure that these two limitations are broken on the basis of a certain degree of decentralization.

2.3 Essence of performance

The booming blockchain technology has a bottleneck that has yet to be broken through: with the scale and volume of today's digital world, any online system without a large-capacity, high-throughput infrastructure cannot host even an Internet-level application.

It's a pity that Satoshi Nakamoto's paper did not consider this issue at all. Maybe it was not easy to take the first step, and he didn't think too much about the future. Maybe it was such a high-performance design. Too difficult in a centralized system. In short, nearly 10 years have passed, in order to improve the performance of the blockchain system, a lot of projects have appeared one after another, but so far, there is no solution that can carry Internet-level applications. This is a worldwide problem that the brightest scholars and developers in the world are trying to solve.

2.3.1 Throughput

In the current situation where finance is the mainstream application scenario, the most important performance bottleneck of the blockchain system is caused by the broadcast delay of block and transaction data, which is essentially limited by the bandwidth and communication delay of the Internet, which directly restricts the throughput.

Consensus algorithms can't actually help solve the bottlenecks of performance and capacity. Trying to improve the performance of the blockchain system based on unconventional consensus algorithms will basically not substantially improve system performance. In short, solving the bottleneck problems requires ingenuity in the design of distributed systems, which is related to consensus algorithms and cryptography, but the essential starting point is not consensus algorithms and cryptography.

2.3.2 State capacity

State capacity is to a blockchain what memory size is to a computer. But this is not like the TPS of throughput is concerned by everyone. The state capacity is rarely hyped by various projects, because this capacity is not very easy to measure, and it is extremely difficult to expand the state capacity.

The state specifically refers to the use of the blockchain to represent the state of each address (ie, user) and each application on the chain. The sum of all the information that needs to be prepared to verify transactions that come at any time is the state of the blockchain. Typically, for example, this state contains the account balance of each address. When there are abundant applications on the chain, each address will have more information to represent the status of each address in each application.

State capacity refers to how much effective memory space a blockchain system can have to represent the state of the entire chain. The state on the chain must reside in memory at any time, ready to be used to verify transactions that come at any time. This part of the information cannot be placed on the hard disk, otherwise the throughput of transaction verification will be greatly reduced, thus greatly constraining the overall blockchain throughput.

In the blockchain network, each participating node needs to be ready to verify and update the next incoming block at any time, that is to say, each node completely stores the status of every address and every application on the chain. Then the state capacity is essentially limited by the memory capacity of each participating node. In this sense, as long as the memory capacity of each participating node is increased, the effective state capacity of a blockchain network can be increased.

However, increasing the memory capacity of a single node is first of all a very limited improvement. The bigger problem is that it increases the entry threshold for participants, which seriously hurts the decentralization of a blockchain network. If you want to essentially increase the state capacity without increasing the memory pressure of a single node, the only way out is full sharding, at least state sharding. However, this is still a very forward-looking academic research direction.

Here, by the way, another piece of stored information: transactions. Transaction history is the sum of every confirmed transaction since the genesis block. This part of the information will continue to accumulate, only increase and not decrease.

2.3.3 Transaction confirmation delay

In the current mainstream public blockchain, a transaction is not completed in an instant from issuance to final confirmation on the chain, but takes a long time, which may be ten minutes or even up to several hours. During this period, most of the time is waiting in line. In the case of Bitcoin, most of the time there are close to 5,000 to 10,000 transactions that have been issued but not yet confirmed and are temporarily stored in each Bitcoin node in a pool called the mempool. Roughly every 10 minutes, a batch of

transactions will be confirmed on the chain, each batch of about 2000 to 3000 transactions. The rest are just waiting in line in the mempool.

At the same time, the queue of these transactions is not confirmed on a first-come-first-served basis. When each transaction is sent, a transaction fee is attached, and the transaction confirmation will give priority to the one with the higher transaction fee. Therefore, usually when there are too many transactions in the network, the more anxious transactions can jump the queue by attaching more transaction fees. This is why, when the network is congested, transaction fees will soar.

When the throughput of a blockchain system is lower than the demand of the application on it, the delay of transaction confirmation is mainly composed of transaction queuing. This is what everyone thinks, increasing the throughput and increasing the TPS will make the blockchain feel faster. Throughput is indeed where the current blockchain system needs to improve the most and bring lower transaction confirmation latency.

In addition to the queuing delay, the complete process of a transaction from issuance to the first confirmation on the chain includes the following processes:
1. The whole network broadcast of the transaction is diffused, which usually takes 2,3 seconds
2. Queue the transaction (steps discussed earlier)
3. One round of consensus cycle (requires a block interval, such as Bitcoin is 10 minutes)
4. The block containing this transaction is broadcasted to the whole network, which usually takes 4 to 8 seconds

Among them, steps 1 and 4 are mainly determined by the Internet point-to-point communication delay and the number of nodes in the entire network. Step 2 is determined by the consensus algorithm of the blockchain system and its parameters. It is worth noting that, at a rough level, either large block + large interval or small block + small interval can satisfy a certain throughput, and the confirmation delay of the latter (small block) is smaller. But in fact, the bandwidth utilization of small blocks is much smaller than that of large blocks, because in each block, in addition to the confirmed transaction data, it also contains complete consensus-related computing power proof data or verifier's signature data.

3. Design Principle

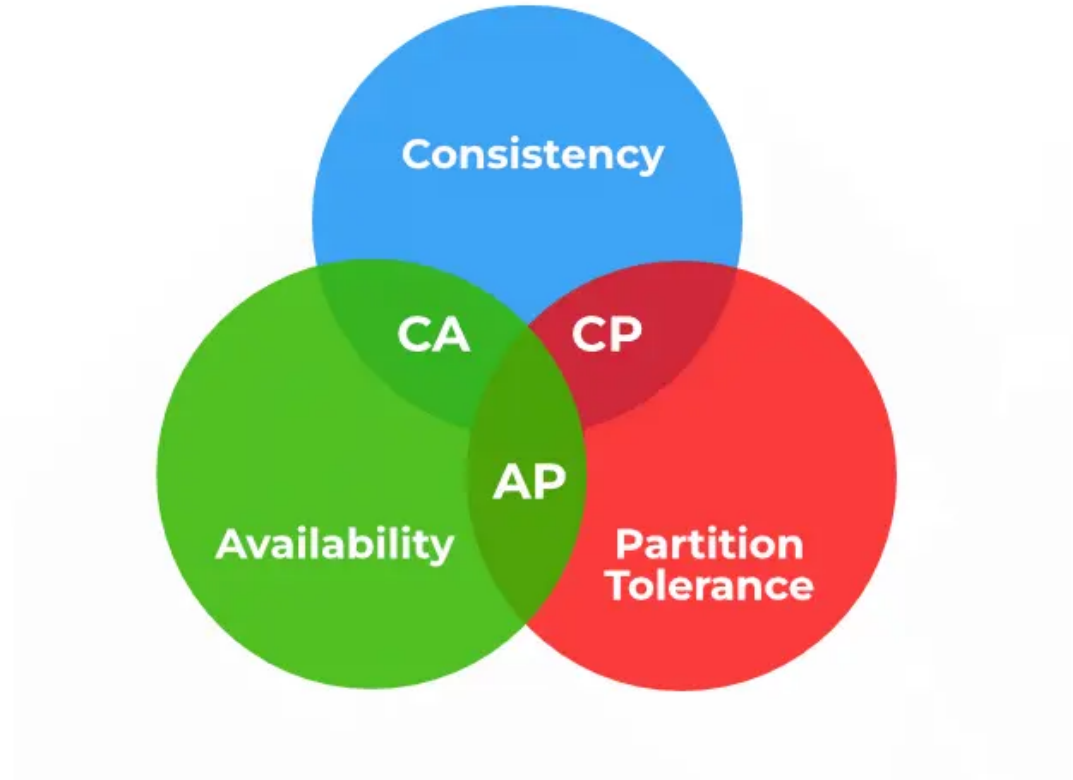
3.1 Performance Theory

3.1.1 CAP Theorem

CAP Theorem states that a distributed system can deliver only two of three desired characteristics: Consistency, Availability, and Partition tolerance. In this article I'll explore what CAP theorem is and how it relates to blockchain technology.

CAP theorem, was introduced by Eric Brewer in 1998 and proven as a theorem in 2002. It states that a distributed system cannot simultaneously
achieve Consistency, Availability, and Partition tolerance.

- Consistency means that all nodes in the system have the same copy of data.
- Availability means that all nodes are up, accessible, and accepting incoming requests.
- Partition tolerance means that the system continues to operate correctly even if a group of nodes is unable to communicate with other nodes due to network failures.



In blockchains, consistency is sacrificed in favour of availability and partition tolerance, achieving eventual consistency over time through validation from multiple nodes.

Bitcoin is “eventually consistent,” which is a polite way to say that it is not consistent. That is, if you send a Bitcoin transaction, there’s no guarantee that it will be received.

3.1.2 Blockchain Trilemma

The blockchain trilemma is a concept that was likely derived from CAP Theorem which highlights the trade-offs that exist between three desirable properties of a blockchain system: *scalability*, *security*, and *decentralisation*.

Like CAP Theorem the Blockchain Trilemma states that it is impossible for a blockchain system to simultaneously achieve all three properties at the same time. This is because each property places different demands on the system, and optimizing for one property requires sacrificing another.

For example, increasing the block size or reducing the block time can improve scalability by allowing more transactions to be processed, but it can also reduce security by making the system more vulnerable to attacks. Similarly, increasing the number of nodes and

their participation in consensus can improve decentralisation, but it can also reduce scalability by slowing down the consensus process.

The trilemma is a challenge for layer 1 blockchain developers who must carefully balance the trade-offs between scalability, security, and decentralisation when designing a blockchain system.

3.1.3 Why scalability is important

Scalability refers to the ability of the blockchain system to handle a large volume of transactions efficiently. As more users join the network and the number of transactions increases, the blockchain must be able to keep up with the demand.

Probably at the early time of blockchain, scalability is not that important given users grow slowly. With more users coming in, more and more applications and transactions demand a new blockchain that can scale, not scale up but scale out.

3.1.4 Scaling Compromises

Different blockchain projects have taken different approaches to this challenge making compromises to meet the principles of their developers and users.
Bitcoin achieves the highest levels of decentralization and censorship resistance at the cost of scalability and high electrical consumption costs. This could potentially cause security issues in the future if declining block rewards can not incentivise a large enough network to prevent a 51% attack.
Ethereum is rolling out upgrades that prioritise scalability at the cost of decentralization and security. Many alternate layer 1 and layer 2 blockchains run centralized sequencers and consensus modules completely compromising on decentralization for the benefit of scalability and security.

3.2 Performance Requirements

What kind of blockchain performance requirements are necessary to bring billions of users to crypto?

1. Ability to process tens of thousands of trades per second and hundreds of thousands of orders per second
2. Ability to process 10 billion social media interactions per day, which is around 100k per second
3. Gas costs of less than $0.001
4. Scaling with the world. As the world's technology grows, the blockchain's throughput has to grow
5. Ability to process all submissions on a human-reaction-time scale. If you click a button you expect it to take ~100ms or so to process on most products — a~1s lag would be really frustrating
6. Products composing. Otherwise you lose a lot of the value-add for companies, meaning each industry will likely want to be roughly single-sharded

7. Ability to be decentralized and open

4. Protocol Design

Rotura utilizes the scale out architecture to create next generation blockchain:

- Scale out up to millions of TPS by adding more nodes
- Distributed Validator Framework makes large scale dApps possible
- SQL+ based Smart Contract framework enables more developers quickly and easily to build future web3 applications

The most important parts of a scale out architecture have to scale the queue and conputation. So the protocol design split into 2 importants scale out framework.

4.1 PoE

PoE stands for Proof of Events.

The PoE is to have a distributed sequence queue which can scale with adding nodes. PoE is built on four major architectural concepts that allow the protocol to be used to build a rich set of applications. Below are the major concepts.

- Order
- Restartability
- Consistency
- Performance

4.2 Rotura Runtime (R2)

R2(Rotura Runtime) utilizes Rotura Decentralized Data Structure(RDD) and PoDAG to allows Tura to scale out with linearity by adding more nodes

The working principle of the R2 is to distribute the code and state of the smart contract to multiple nodes in the network, and these nodes can execute the functions of the contract at the same time. When a node submits a transaction, other nodes will verify and synchronize transaction data and contract status. In this way, each node can independently calculate and confirm the result of the contract function, and submit the result to the blockchain.

The advantages of Rotura Runtime include:

- Higher throughput: Since multiple nodes can execute contract functions at the same time, the R2 can handle more transactions.
- Faster speed: R2 distributes the computing tasks of the contract to multiple nodes, which can speed up the execution of contract functions.
- Better scalability: R2 can add more nodes to increase processing power as needed.

However, R2 also has some challenges, such as:

- Network communication delay: R2 needs to transfer contract code and state between multiple nodes, which may cause network communication delay.
- State synchronization: Executing contract functions on multiple nodes requires ensuring that the contract state of each node is synchronized, which may require additional overhead and complexity.
- Security: R2 needs to ensure that the execution results of contract functions on multiple nodes are consistent, which may require special security measures.

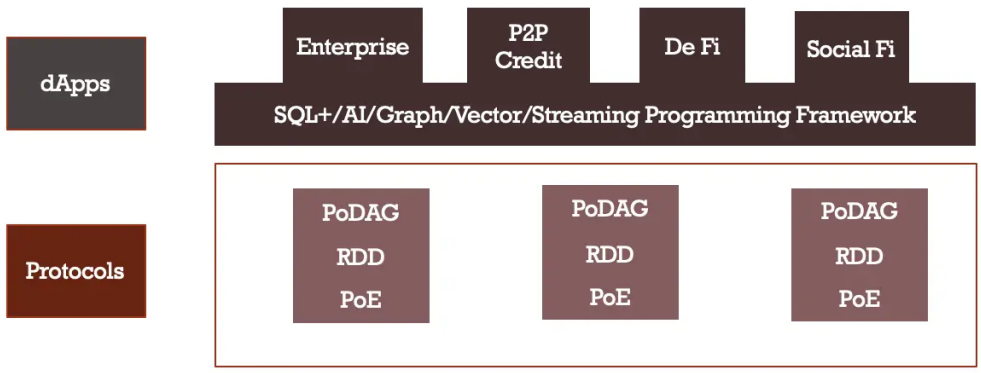
In conclusion, R2 can improve the performance and scalability of blockchain, but it also needs to solve a series of technical challenges.

4.3 Modular design

Modular design, or modularity in design, is a design principle that subdivides a system into smaller parts called modules, which can be independently created, modified, replaced, or exchanged with other modules or between different systems.

5. System Design

This section is intended to give an architecture design of the system as a whole.



5.1 Rotura Decentralized Dataset (RDD)

RDD(Rotura Decentralized Dataset) are the primary data structure in Rotura Runtime. RDDs are reliable and memory-efficient when it comes to parallel processing. By storing and processing data in RDDs, R2 speeds up data process.

5.1.1 RDD Features

The main features of a RDD are:

- In-memory computation. Data calculation resides in memory for faster access and fewer I/O operations.
- Fault tolerance. The tracking of data creation helps recover or recreate lost data after a node failure.
- Immutability. RDDs are read-only. The existing data cannot change, and transformations on existing data generate new RDDs.

- Lazy evaluation. Data does not load immediately after definition – the data loads when applying an action to the data.

5.2 Proof of Events (PoE)

Traditional blockchain architectures require a commitment to the result of each block's state update to be included as part of the consensus process. As a result, every node must reproduce the state-update computation before it can finalize a block. Our finding is that consensus on the order of transactions in the block is all that is required. Once that order is fixed, the resulting computation is determined even though it may not necessarily be known. Thereby, the computational effort of participating in consensus is significantly reduced, even for a very large number of transactions. Once the transaction order is determined, ensuing processes can be delegated to perform the computation itself, without affecting decentralization of the system.

PoE introduces a whole new distributed architecture. The technology brings reliability, performance and scability, which maintains the transactions in an order of insertion. And the PoE is able to handle those transactions in real time with very high speed.

5.2.1 Order

The first major concept is order, and it is the foundation for the rest of the major architectural concepts. Order is important because it allows applications to reason about causality of data – or in other words allows an application to know if an operation occurred before or after another operation. PoE achieves order through the use of sequence numbers, Snapshots and failover logs. Sequence numbers are used to keep order on a single node and failover logs are used to resolve ordering conflicts during failure scenarios.

5.2.2 Restartability

Restartability is an important feature for any replication protocol that often has to deal with large amounts of data. In any distributed system components crash and when you're moving large portions of data applications want to be able to resume from exactly where they left off and not have to resend any data in the case of a dropped connection. Due to the PoE's ordering mechanism restartability is possible from any point and means the server won't send any data that the application has already received even if the application has been disconnected for an extended period of time.

5.2.3 Consistency

Some applications can only make decisions about the data they receive if they have a consistent view of the data in the database. PoE provides consistency through snapshots and allow applications know that they have seen all mutations in the database up to a certain sequence number.

5.2.4 Performance

PoE provides high throughput and low latency by keeping the most recent data that needs to be replicated in memory. This means that PoE connections will normally not

have to read data off of disk in steady-state.

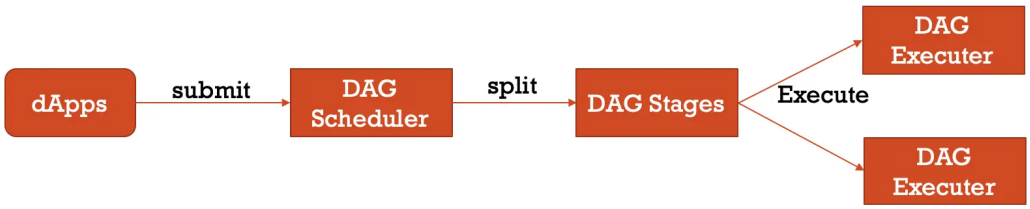
The benefits of PoE are:

1. Scalable. It is designed in a distributed system
2. Fast. It supports realtime application.
3. Interoperable. It is designed to be customizable.
4. Support huge volume of data.
5. In memory.
6. Hash based sharding to ensure the validation of transaction order

5.3 Rotura Runtime (R2)

R2(Rotura Runtime) utilizes Rotura Decentralized Data Structure(RDD) and PoDAG for a distributed execution system.

- DAG (Directed Acyclic Graph) in R2 is a fundamental concept that plays a crucial role in the R2 execution model.
- The DAG is “directed” because the operations are executed in a specific order, and “acyclic” because there are no loops or cycles in the execution plan.
- Each stage depends on the completion of the previous stage, and each task within a stage can run independently of the other.



5.3.1 PoDAG – Proof of DAG

DAG, Directed Acyclic Graph, the abbreviation of Directed Acyclic Graph, is often used in modeling.

PoDAG stands for Proof of Directed Acyclic Graph, is the foundationtal execution system of distributed computing for data driven dapps

Problems DAGs Solve

The emergence of DAG is mainly to solve the limitations of the EVM framework, especially the scalable one. DAG is used in R2 to model the relationship of computations and describe the dependency relationship of computations. This relationship is also called lineage.

The scheduling mechanism of R2 uses DAG to ensure distributed execution to the greatest extent, while ensuring security and traceability, and preventing errors during execution.

So why convert to DAG

The scheduling mechanism of R2 is very suitable for DAG. Because the computation call of R2 is lazy, it is necessary to record the dependencies, so as to prevent errors during execution and trace the source according to the dependencies. Since there is a dependency relationship, it is in line with the concept of DAG (directed acyclic graph).

DAG is a graph structure, a graph connected by edges between multiple vertices

Why is it a directed graph

Since different vertices are dependent, the edges of the lines between vertices are directional.

Why is it an acyclic graph

If there is a loop, it means that there may be a circular dependency between different vertices, such as: A->B->C->A, in this case, it will lead to an infinite loop, and no entry point for execution can be found.

If it is acyclic, you can filter out the initial cut-in vertex by judging whether each vertex has a dependent parent vertex.

5.3.1 Distributed Realtime Validation

Validator is responsible for the real time validation. Validator retrieves the transaction in order which is done by PoE. PoS elected Validator is responsible to distribute the steaming into mempool for other Validators to perform the validation in parallel so accelerate the validation.

5.3.2 Distributed realtime block forging

Once the selected validator finish validating the transaction, it starts to forge the block. And the block is not required to validate all the transaction but to validate the miner's information. Given there are many transaction to be forged into block, Rotura uses distributed sharding stream to handle the forging in parallel.

Once the deadline reaches, the final forging process will collected all forged blocks into the final block and add to blockchain. This is done by the elected validator.

5.4 Development Framework

5.4.1 ChainQL – SQL on Blockchain

SQL is one of the popular structured Query languages. It allows the developer to easily manipulate and access blockchain. ChainQL is the language used to operate the blockchain.

5.4.2 Machine Learning Lib

It contains machine learning libraries that have an implementation of various machine learning algorithms.

Use Cases:

- Label
- Classification
- Prediction

5.4.3 Graph

It unifies ETL, exploratory analysis, and iterative graph computation within a single system.

Use Cases:

- Ranking
- Connected Components
- Social Fi
- Game Fi

5.4.4 Vector Engine

- Chatbots with long-term memory on cloud and short-term memory on devices
- Combine the power of Generative AI models to generate highly relevant, grounded responses
- Extends use cases for broad AI integration

5.4.5 Streaming

Streaming, a scalable and fault-tolerant stream processing engine built on the SQL R2 engine. Streaming queries are processed using a micro-batch processing engine, which processes data streams as a series of small batch jobs thereby achieving end-to-end latencies as low as 100 milliseconds and exactly-once fault-tolerance guarantees.

5.4.6 Pipeline

- Connect dApps in a workflow to create a more complicated and large-scale dApp
- The workflow is represented as a DAG

- It contains individual pieces of work arranged with dependencies and data flows taken into account.

6. Conclusion

We have outlined a direction one may take to author a scalable protocol, which possible is the solution of blockchain trilemma. We have identified a basic design and discussed its strengths and limitations; accordingly we have further directions which may ease those limitations and yield further ground towards a fully scalable blockchain solution.