Fractal Ledger: A New Approach towards Scalable Blockchain with Graph-Chain Structure

Tong Zhou
Hefei Institutes of Physical Science, Chinese Academy of Sciences
Anhui Zhongke Jingge Technologies Co., Ltd.
Hefei China
tzhou@hfcas.ac.cn

He Zhao*
Hefei Institutes of Physical Science, Chinese Academy of Sciences
Hefei China
zhaoh@hfcas.ac.cn

Nianzu Shen
Hefei Institutes of Physical Science, Chinese Academy of Sciences
University of Science and Technology of China
Hefei China
nianzu@mail.ustc.edu.cn

Bin Yu
Anhui Zhongke Jingge Technologies Co., Ltd.
Hefei China
yubin@zkjg.com

Xiaofeng Li
Hefei Institutes of Physical Science, Chinese Academy of Sciences
Hefei China
xfli@hfcas.ac.cn

Jinlin Xu
Hefei Institutes of Physical Science, Chinese Academy of Sciences
University of Science and Technology of China
Hefei China
jlxu@hfcas.ac.cn

ABSTRACT

Scalability is often considered the "Achilles' heel" of blockchain technology. With a traditional chain-based structure, blocks cannot be generated concurrently, thus limiting the throughput and slowing transaction confirmation. Recently emerged graph-based blockchains generally excel in on-chain performance but are weaker in structural flexibility, which is essential for off-chain or cross-chain scaling. In this paper, we propose Fractal Ledger, a novel blockchain model to scale out the ledger in a fractal manner. Fractal Ledger is comprised of fractal units that synthesize both chain and graph structure. Through conversions and interactions of fractal units, Fractal Ledger can share the advantages of both types of blockchains and offer scalability in structure and storage. We introduce the deduction mechanisms for flexible structure scalability and a three-directional folding scheme of lossless on-chain data compression for storage scalability. Our experimental results reveal the effectiveness of Fractal Ledger in enhancing structure and storage scalability.

CCS CONCEPTS

• Software and its engineering → Software organization and properties → Software system structures → Distributed systems organizing principles

*Corresponding Author

CF '23, May 9–11, 2023, Bologna, Italy

ACM ISBN 979-8-4007-0140-5/23/05.
https://doi.org/10.1145/3587135.3592203

KEYWORDS

Blockchain, Scalability, Directed Acyclic Graph, Compression

ACM Reference format:

Tong Zhou, He Zhao, Nianzu Shen, Bin Yu, Xiaofeng Li, and Jinlin Xu. 2023. Fractal Ledger: A New Approach towards Scalable Blockchain with Graph-Chain Structure. In *20th ACM International Conference on Computing Frontiers (CF'23), May 9-11, 2023, Bologna, Italy. ACM, New York, NY, USA, 9 pages.* https://doi.org/10.1145/3587135.3592203.

1 INTRODUCTION

Blockchain, the distributed ledger technology (DLT) originally invented for cryptocurrencies, is reshaping various industries such as finance, supply chain management and healthcare, owing to its persistency, anonymity, and auditability [1]. Like other groundbreaking and novel technologies, blockchain still suffers from fundamental issues, among which lies scalability. The ever-increasing demand for including more transactions and data on-chain, calls for efficient, flexible, and sustainable solutions. However, scalability has often been described as the "Achilles' heel" of blockchain [2]. Because the technical limitations of existing on-chain and off-chain solutions such as sharding [3], side-chain [4] and cross-chain interoperation [5], they have not yet been able to successfully tackle the problem.

Currently, mainstream blockchain technology involves single chain of blocks structured in a linear, chronological way. While this structure is simple and easier to maintain, it is also with significant drawbacks: concurrent transactions competing for one valid block with limited capacity each round, resulting in longer confirmation time and lower throughput [6]. Users have to bid for exorbitant fees to bypass the massive transaction backlogs. As a result, on-chain solutions with asynchronous non-blocking transaction have been investigated to lift the global state lock that restricted by linearly structured blockchains. Directed Acyclic

Graph (DAG) structure is a promising underlying architecture aiming to process more transactions concurrently, with transactions or blocks formed in graph topology [7-10]. However, graph-based ledgers are less thoroughly studied. For example, it is challenging to implement smart contracts in DAG ledgers. More importantly, the methods developed to scale traditional chain-based ledgers, such as sharding and side-chain, are generally inapplicable with them [6]. The aforesaid challenges, as we believe, could be solved by designing a graph-chain architecture, which combines the advantages of both types of ledgers, offering a distinct solution to blockchain scalability.

Besides structure aspects, storage is another important factor to take into account when designing a scalable and sustainable blockchain. As of January 31st, 2023, the number of blocks generated by Bitcoin [11] blockchain is 774,468 and the data size is 451.34GB [12]. The number of blocks generated by Ethereum [13] is 16,530,247 and the data size is 1183.74GB [14], and it is still growing at an accelerated rate. The drastically increased on-chain data exerts great burdens onto network participants. Users tend to switch to lightweight clients rather than full nodes, which hinders network decentralization. Currently, projects like Mina adopts recursive ZK-proofs to keep minimal node storage requirement [15]. Yet participants have to rely on a few archive nodes to access ledger contents, which may also contribute to increased centralization. Therefore, it is essential to explore different means to scale node storage that does not impede the availability and integrity of the ledger data, or lowering network decentralization.

The main contributions of this paper are:

- We introduce the basic model of our proposed solution featuring the graph-chain structure. The model supports non-blocking concurrent execution of parallel transactions, and natively supports smart contracts.
- We present a scaling solution with the deduction mechanism based on the designed graph-chain structure to form a fractal ledger. The proposed approach can scale out the ledger in fractal manner, that is, the ledger structure can be made up of parts that are the same structure as itself and are at different scales. The scalability mechanisms serve as a possible complement to existing graph-based systems and enables Fractal Ledger to flexibly adapt to different application scenarios, thereby realizing the blockchain structure's scalability.
- To improve storage scalability, this paper develops a comprehensive folding scheme to considerably lower blockchain storage demands. The ledger data is losslessly compressed along three directions, contributing to a more sustainable blockchain in the long term.

This paper focuses primarily on the structure and storage scalability, so classical PBFT [16] is adopted as the consensus algorithm of Fractal Ledger. Due to the length limit of the paper, a new consensus algorithm that further exploits the advantages of Fractal Ledger will be reported independently.

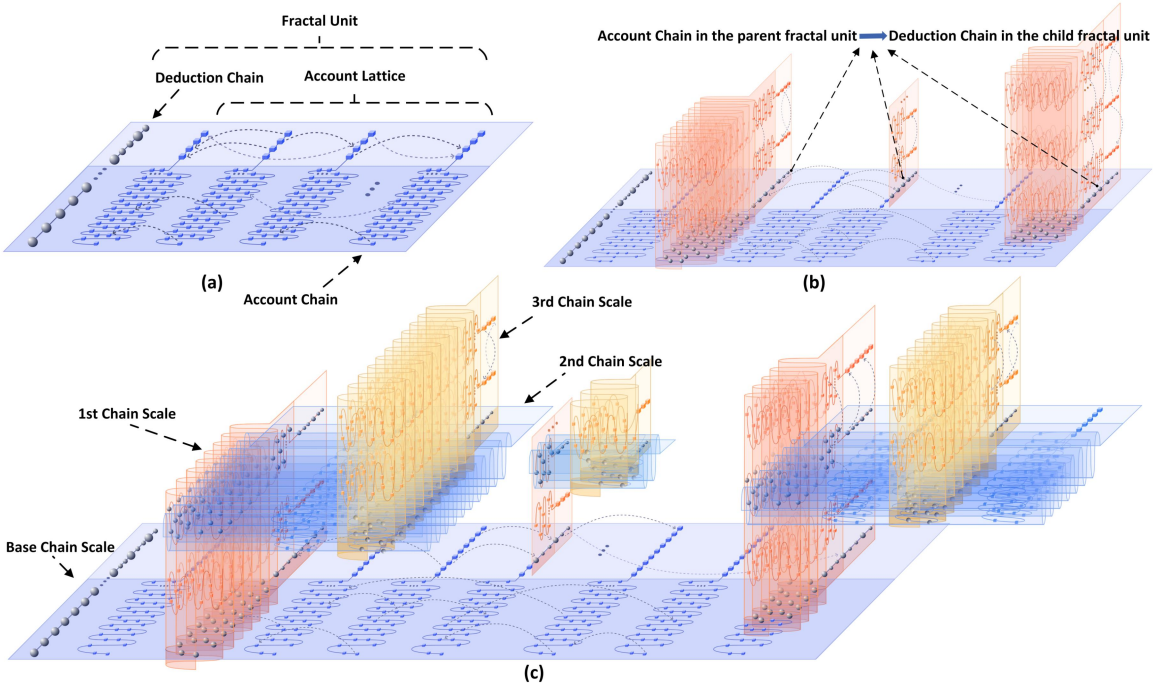


Figure 1: The overview of Fractal Ledger. (a): Fractal Ledger exhibits an initial graph-chain-based fractal unit which includes a deduction chain and an account lattice. The account lattice is comprised of account chains linked together with asynchronous transactions forming a DAG structure. (b): The fractal unit forms three child fractal units in the next chain scale using deduction mechanism (See Section 3.1). (c): After layer-by-layer deduction, Fractal Ledger consists of various fractal units with different chain scales.

2 MODEL DESIGN

As shown in Figure 1, Fractal Ledger is based on graph-chain structure. As the basic component of the system, a fractal unit consists of an account lattice (graph-based) and a deduction chain (chain-based).

2.1 Model Description

Transaction Block and Account Lattice

In Fractal Ledger, each transaction constitutes a single block, so we call transactions as transaction blocks TB. Each account sends transaction blocks on its account chain to transfer funds and execute contracts. All account chains are directly linked together by transactions forming the DAG-structured account lattice. According to the different functions of TB, they are divided into the two types as shown in Figure 2.

1. Transfer transaction for sending and receiving tokens, including send block (TB_{send}) and receive block (TB_{rec}).
2. Contract transaction for supporting smart contracts, including deploy block (TB_{dep}) and execute block (TB_{exc}).

TB are constructed by accounts and trigger transitions of the account state. Besides, the account deploys the smart contract by TB_{dep} and executes the contract by TB_{exc}. Contract transaction rewards both the block producers and consensus nodes contracts execution fees.

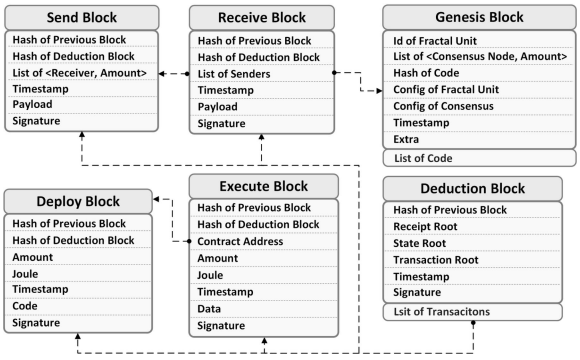


Figure 2: Data structure of the transaction blocks and the deduction blocks.

Deduction Block and Deduction Chain

The deduction chain, separated from the account lattice, is a linear chain used to sequence the contract transaction, witness, and compress the account lattice, and resolve forks. The deduction chain consists of a Genesis Block (DB_0) and deduction blocks (DB_{BH}), which serves as the foundation of the fractal scaling operations.

A Genesis Block DB_0 is the first block in both the deduction chain and the account lattice. It contains necessary initial configuration information for the entire Fractal Ledger, including the id of the fractal unit, total token supply, configuration parameters for the consensus algorithm, and the set of initial accounts, etc.

Deduction block $DB_{BH} \in \{DB_0, \ldots, DB_k\}$ (BH stands for the height of a deduction block) is periodically created by the deduction block producer (producer for short), and witnesses the current state of the account lattice at a given block height, sorts the order of the contract transactions in the account chain and completes the multiple chain scale deduction process. We define "witness" in this paper as a process that a producer bundles transaction blocks into a deduction block and the block consensus can be reached among the consensus nodes.

Chain Scale and Fractal Unit

The transition relationship between deduction chains and account chains is one of the key characteristics of Fractal Ledger, which enables the blockchain to scale out flexibly and support different application scenarios. The chain scale, noted as CS, is checked through each fractal unit's deduction chain. The deduction chain of higher chain scale can be deduced through the account chain of lower chain scale, thus forming a cross chain scale fractal blockchain.

Each fractal unit consists of a deduction chain and an account lattice. A fractal unit is denoted as FU. Multiple fractal units can co-exist at the same chain scale and different child fractal units can originate from a single parent fractal unit using the deduction mechanism described in Section 3.

2.2 Transaction Block State Transition

Figure 3 illustrates the general state transition process for the transfer transactions, specifically shown within the block epoch between $DB_{BH=1}$ and $DB_{BH=2}$. The process primarily consists of an on-chain phase and a witness phase.

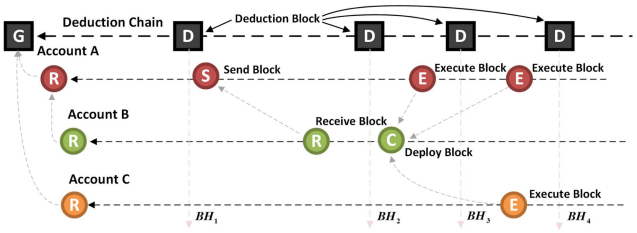


Figure 3: State transition of the transaction blocks.

On-Chain Phase. The account lattice is where the on-chain phase primarily takes place. On its account chain AC_{sender}, the asset sender U_{sender} first creates a send block TB_{send} with a digital signature, and then sends the block to the network. The TB_{send} of agreement is reached among consensus nodes, and asset is subtracted in U_{sender}'s account. The asset receiver $U_{receiver}$ then selects the un-received send blocks to construct the receive block TB_{rec} on its $AC_{receiver}$. Once the agreement of TB_{rec} has been reached, the asset receiving process is finished.

Witness Phase. The witness phase occurs mainly on the deduction chain. The producer periodically groups the transaction blocks into a deduction block during each block epoch. The deduction block will be broadcasted to the whole network after

being signed with a digital signature. The deduction block is confirmed after the consensus has been reached.

2.3 Execution Flow of Smart Contract

The general process of smart contract deployment and execution is also shown in Figure 3 (the exact process is shown within $DB_{BH=2}$ and $DB_{BH=4}$). A contract transaction has the same on-chain and witness processes as those of transfer transactions.

Deployment Phase. The contract deployer U_{dep} creates a deploy block on its account chain as part of the contract deployment procedure. Following the on-chain phase similar to the transfer transactions, the smart contract is deployed and ready to be executed by other accounts.

Execution Phase. In the execution phase, the contract executor $U_{executor}$ constructs the execute block TB_{exc} on its own account chain $AC_{executor}$ to call a contract function, and TB_{exc} will be anchored on $AC_{executor}$ after the same on-chain phase. At this time, the contract function is pre-executed.

In the witness phase, each producer individually sorts all execute blocks at that block height referencing to how much Gas [13] they cost. Only when the consensus $of\ DB_{BH}$ is reached can all contracts included in that block can be truly executed.

3 SCALABILITY OF FRACTAL LEDGER

Leveraging the benefits of the graph-chain structure, we utilize different transformation relationships between account chains and deduction chains to realize scaling solutions of Fractal Ledger by the designing deduction mechanism and folding-compression.

3.1 Deduction Mechanism

Deduction mechanism refers to the method of forming a multiple chain scales fractal model with layer-by-layer deduction process from a parent fractal unit lower chain scale to child fractal units higher chain scale using Deduction Scale Block (DsB) based on the graph-chain infrastructure. Figure 1(b) shows that a fractal unit forms three child fractal units in the next chain scale using deduction mechanism.

The account lattice in a new child fractal unit formed by the deduction mechanism is witnessed by the same fractal unit's deduction chain. And the child fractal unit as part of the account lattice in the parent fractal unit is also witnessed by its parent fractal unit's deduction chain.

The deduction process that the child fractal unit FU_{child}^{cs+1} is formed from the account chain AC_k in the parent fractal unit FU_{par}^{cs} is as follows:

The account U_k first send DsB on its AC_k and launch the deduction operation. DsB, which includes the set of consensus nodes, total supply of tokens, consensus algorithm and other core parameters required for the new formed fractal unit, is primarily used as the Genesis Block of FU_{child}^{cs+1}.

After DsB's consensus is reached in FU_{par}^{cs}, AC_k is transformed into an account deduction chain ADC_k, which only accepts the deduction block built in FU_{child}^{cs+1} and no longer receives the transaction blocks from U_k. U_k can transfer its funds in advance and lock the tokens for the new fractal unit. At this point, FU_{child}^{cs+1} keeps growing with core parameters defined by DsB and is similar to a child-chain. It is noteworthy that this approach allows the layer-by-layer deduction of different account chains with the same chain-scale as well as different chain-scale.

3.2 Folding Scheme

Mainstream blockchains face high storage demands, which can lead users to abandon running full nodes which stores and validates the ledger data. Instead, they may rely on third-party services, which poses a serious threat to the security and decentralization of the blockchain network. Additionally, the graph-chain structure proposed in this paper adds various auxiliary information to maintain the account lattice structure, which differs from traditional blockchain designs. So, this section suggests a folding scheme based on Fractal Ledger, which will perform X-axis Fold, Y-axis Roll-in and Z-axis Condense respectively as shown in Figure 4 to achieve lossless compression.

X-axis Fold means to optimize the growth process of the account chain and the deduction chain respectively along the growth direction of Fractal Ledger.

For the growth of the account chain, the original transfer process is optimized to receive funds while building the send block. This allows us to send and receive multiple funds in a single transfer transaction. Figure 5(a) depicts the transaction block sent and received in the original transfer process, while Figure 5(b) shows the transfer transactions after applying X-axis Fold to the account lattice. As can be seen from Figure 5, the number of transfer transactions required after folding is significantly reduced.

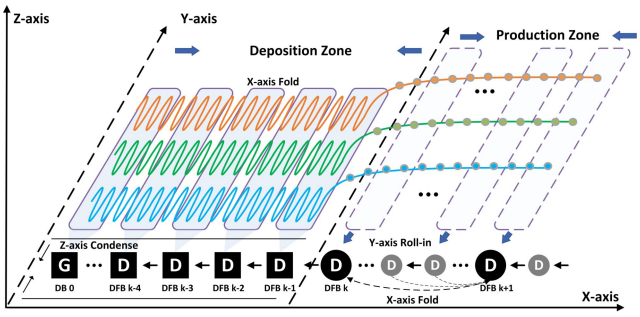


Figure 4: The three-directional folding scheme. The ledger is divided into the production zone and the deposition zone. Storage optimizations are carried out three directions along X-axis, Y-axis, and Z-axis. From DB_0 to DFB_{k-1}, blocks with black square are deduction-folding-blocks in the deposition zone. From DFB_k to the latest are deduction blocks in the production zone, including the deduction-folding-blocks (black circle) and the deduction blocks (small gray circle).

During the growth of the deduction chain, the deduction-folding-block (DFB) is constructed by defining the folding epoch FE and the producer merges multiple deduction blocks within a DFB. DFB_1 is built at block height BH_3 in the deduction chain for

FE_3 , and multiple deduction blocks in the folding epoch are merged into the DFB_1 as shown in of Figure 5(b).

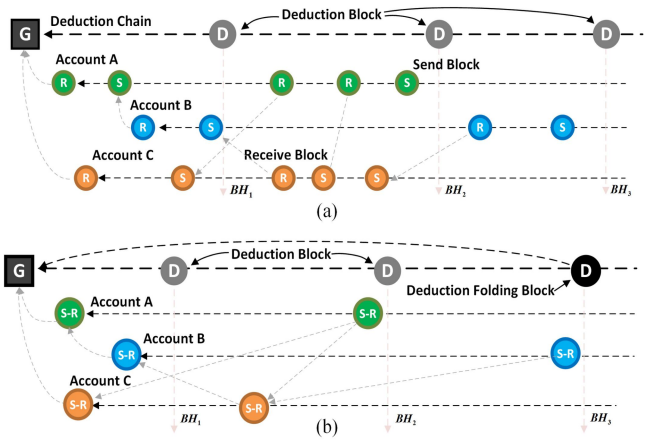


Figure 5: X-axis Fold Scheme

Y-axis Roll-in refers to compressing data in the process of grouping transaction blocks into the deduction chain by using the referencing relationship between the account lattice and the deduction chain. The process is similar to roll account chains into the deduction chain, so it is called Y-axis Roll-in. Inspired by deposition process in geology, the ledger is divided into the deposition zone and the production zone according to deduction block height.

The production zone of the ledger contains latest transaction blocks and deduction blocks. The deduction chain witnesses the new transaction blocks and sorts the contract transactions within the current block epoch when producing the deduction block sequentially.

The deposition zone contains the ledger's historical data. To complete the lossless compression of transaction blocks in the deposition zone, historical account lattice data is no longer organized in the form of the original structure in production zone. All transactions are grouped in the deduction blocks in the form of the tuple $< Addr, Height, Hash >$ as index with their signatures abstracted, and reference information replaced by nonce, which represents the order of the transaction block in its account chain.

Z-axis Condense refers to the data compression of both transaction blocks and deduction blocks. The process is carried out simultaneously when the blocks in the deposition zone are re-organized.

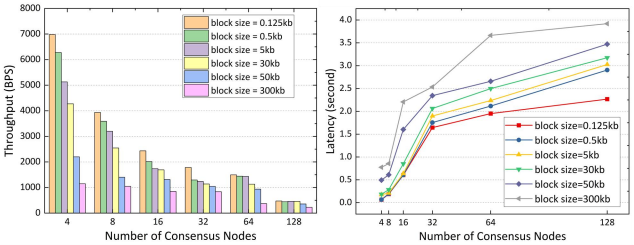
The process mainly involves techniques such as coordinate positioning [17], relative values, extraction of redundant items and optimization of data types. Specifically, in the deposition zone, transaction blocks are re-organized in the account lattice which use the nonce to record the order of transaction blocks in its account-chain, and abstract some auxiliary verification information such as digital signatures and deduction block hash. We reduce the number of bytes of timestamps by means of relative values, and replace the address involved in the transaction block with the coordinates of the initial block wherein deduction chain, such as using the tuple <DBlock No., TBlock No.>.

4 EXPERIMENTAL EVALUATIONS

In this section, we examine Fractal Ledger's performance and storage scalability. Our core system is implemented using Golang. We construct a fully connected P2P network. The system is deployed on a distributed environment that includes up to 128 virtual machines, each of which has 8 cores and 16GB memory. In the test network, we restrict the end-to-end peak bandwidth to 3Gbps.

4.1 Throughput and Latency

We adopt a classic PBFT [16] consensus (See Section 5.2) in a fractal unit to evaluate the performance.



(a)Throughput **(b) Latency**

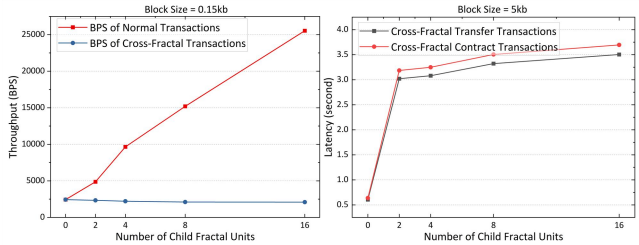
Figure 6: The performance of the fractal unit with various block sizes and consensus node numbers.

We first measure the basic throughput of a single fractal unit by saturating it with transactions and observing the number of confirmed transactions in the steady state. To do this, we broadcast transactions differently based on the number of consensus nodes in a single fractal unit. For example, in a unit with 128 consensus nodes and 16 account chains per node on the Fractal Ledger, each account broadcasts 2 transaction blocks per second, resulting in a maximum of 4,096 blocks being broadcast simultaneously across the network. For 4 consensus nodes, we create 16 account-chains per node, and each account broadcasts 180 transaction blocks per second to saturate the network.

The average size of a transfer transaction without payload data in Fractal Ledger is approximately 128B. The size of a contract transaction, including the deploy and execute blocks, varies depending on the specific contract. And the size of a deduction block varies according to the number of transaction blocks in the current network. Therefore, we conduct experiments in fractal units with 4 to 128 consensus nodes and block sizes ranging from 0.125KB to 300KB. The results of Fractal Ledger's BPS are shown in Figure 6(a).

In traditional blockchain systems, each transaction is transmitted twice to all nodes: it is first broadcast to all nodes, and when a block containing the transaction is mined, it is broadcast to all nodes for the second time within the mined block [18]. In Fractal Ledger, however, for transfer transactions and deduction blocks, the confirmation time only includes the broadcast time of the block and the time when the consensus nodes collect votes. As for the contract transaction, its execution depends on the timing of the deduction block to sort the contract transaction. Figure 6(b)

shows the confirmation latency using the same experimental setup as the throughput experiments.



(a)Throughput **(b)Latency**

Figure 7: (a): The throughput of a parent fractal unit with various number of child fractal units. (b): The average confirmation latency of various number of child fractal units.

In the fractal experiment, we configure the parent fractal unit denoted as FU_0 with 16 witness nodes containing 16 account chains. We adopt the deduction mechanism to form 2, 4, 8 and 16 child fractal units respectively on the same chain scale. Each child fractal unit also contains 16 account chains with 16 witness nodes. For normal transfer transactions, the total BPS of Fractal Ledger grows linearly because the 16 child fractal units are relatively independent, but for cross-fractal transactions, BPS is limited by the worst performing fractal unit. The results are as shown in Figure 7(a).

For the latency of cross-fractal transactions, the transactions from the child fractal unit FU_1 to the child fractal unit FU_2 needs to be sent in the account chain in FU_1 first and are witnessed by the FU_1's deduction chain, and then consensus is reached in FU_0. After the transfer transaction is confirmed by FU_2's deduction chain, the transaction can be received by the target account chain. For the cross-fractal contract execution, the contract execution transaction should be constructed in FU_1's account chain, and the consensus can be reached in the FU_0's deduction chain. After the deduction chain of FU_0 witnesses the deduction block constructed by the deduction chain of FU_2 to sort and execute the contract, the execution result is finally fed back to the executor. The results are as shown in Figure 7(b).

4.2 Folding Compression Ratio

We replay 600 blocks of Ethereum main-net in Fractal Ledger (block height from 15,201,001 to 15,201,600), to evaluate the proposed three-directional folding scheme.

According to the average block generation time of Ethereum blocks, the block interval of the deduction chain is set to 12s. Figure 8 shows the experimental results of the deduction blocks, which includes transaction blocks processed with Y-axis Roll-in, after using Z-axis Condense and the proposed folding scheme respectively. The proposed folding scheme includes X-axis Fold, Y-axis Roll-in, and Z-axis Condense. The average compression ratio is about 70.69% after only using Z-axis Condense to each deduction block (including collected transaction blocks). If we only apply Z-axis Condense to the transfer transactions (ignoring the contract transactions), the average compression ratio is about 28.89% as shown in Figure 8(a), which shows that Z-axis Condense proposed in this paper has a significant compression effect on transfer transactions.

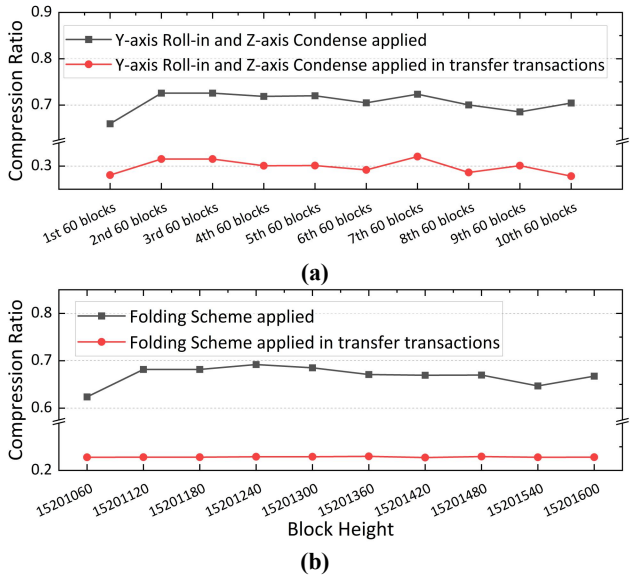
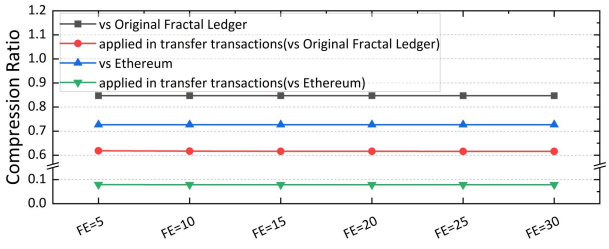


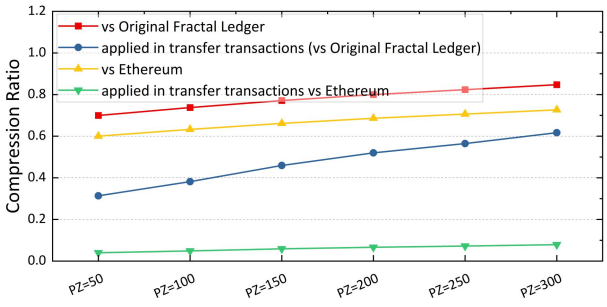
Figure 8: (a): The average compression ratios per 60 deduction blocks, each of which includes transaction blocks by Y-axis Roll-in, after using Z-axis Condense compared with Original Fractal Ledger; (b): The compression ratios of these deduction blocks after using the folding scheme compared with Original Fractal Ledger.

In the further folding experiment, we set the folding epoch FE_{60} to carry out the folding scheme in the deduction blocks. When the folding scheme is used to compress transfer transactions, the compression ratio decreases to 22.83% compared with the original Fractal Ledger as shown in Figure 8(b).

In order to observe the influence of different folding epochs and production zone sizes on the three-directional folding scheme, we first fix the production zone size (short for PZ in the figure 9(a)) to 300 deduction blocks, and evaluate the folding scheme by setting different folding epochs. The experimental results are shown in Figure 9(a). It can be seen that the longer the folding epoch, the slightly higher the compression ratio. When the folding epoch is fixed to 10, we set the different production zone sizes to observe the compression ratios. The smaller the proportion of the production zone (*i.e.*, the larger the deposition zone), the less storage space is required.



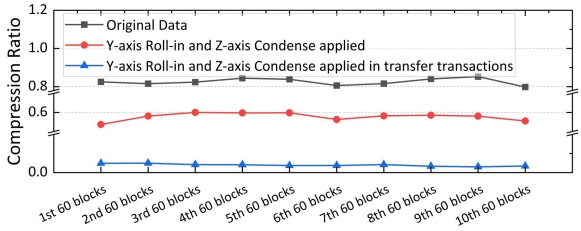
(a) with different folding epochs



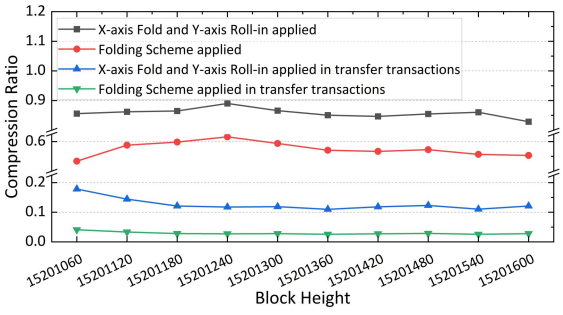
(b) with different production zone sizes

Figure 9: Compression ratios after using three-directional folding scheme.

In the comparison experiment with Ethereum, we replay the complete historical transactions of Ethereum from the height 15,201,001 to 15,201,600 in Fractal Ledger, which includes 90,137 transactions. In Figure 10, we can see that the size of the block data generated by each deduction block in Fractal Ledger is slightly smaller than the block size of Ethereum. In terms of block content, Fractal Ledger has more data than Ethereum block. However, since there is no need to store receipt data and state data for transfer transactions in Fractal Ledger due to account lattice structure, the overall data volume is reduced. Compared with Ethereum, after Y-axis Roll-in and Z-axis Condense, the compression ratio is 57.81%. If we apply Y-axis Roll-in and Z-axis Condense only to transfer transactions, the compression ratio is about 3.94% in Figure 10(a).



(a)



(b)

Figure 10: (a): The average compression ratios per 60 deduction blocks, each of which includes transaction blocks by Y-axis Roll-in, after using Z-axis Condense compared with Ethereum; (b): The compression ratios of these blocks after using the folding scheme compared with Ethereum.

In the further folding experiment, we set the folding epoch FE_{60} to observe the Fractal Ledger' block data after using X-axis Fold and Y-axis Roll-in. Compared with Ethereum, as can be seen from the Figure 10(b), the compression ratio is about 85.79%. While the compression ratio of using the folding scheme is 57.5%. If the scheme as stated are applied only to the transfer transactions, the compression ratios are about 12.62% and 2.92% respectively.

5 ANALYSIS AND DISCUSSION

5.1 Scalability Analysis

The scalability of Fractal Ledger is mainly reflected in following aspects: From the structure scalability aspect, in contrast to the traditional linear blockchain design, the proposed model reduces transaction latency and improves transaction throughput by adding transactions on its own account chain without blocking the entire system. Compared with the conventional graph-based systems, a fractal blockchain model can be achieved by utilizing different interactions between deduction chains and account chains. The deduction mechanism can vertically stratify the blockchain structure to meet multi-level application needs and can also be horizontally fractal by different application types, which solves the problem of the insufficient off-chain scaling solutions in existing graph-based systems, thus realizing the structure scalability of Fractal Ledger. Based on the flexible structure scalability, it is also possible to construct the side-chains, or create the application-specific child-chains and realize the transaction sharding through the deduced fractal units.

Comprehensive measures are also taken to improve storage scalability. A report from Infura[1], an influential blockchain service provider, demonstrated that 70 percent of the block service pressure is focused on the top of the blockchain, while the existing techniques tend to treat the old and new blocks equally. In this regard, we design a three-directional folding scheme. We divide Fractal Ledger into the production zone and the deposition zone. Along X-axis, the streamlining of the asset trading procedure is implemented in growth process of the account lattice. And the deduction chain is folded by grouping deduction blocks into a deduction-folding-block. Along Y-axis, in the production zone, the newer deduction blocks record the index of the transaction blocks, whereas in the deposition zone, the original structure of the account lattice is reorganized, and transaction blocks are bundled into the deduction chain. Along Z-axis, to further improve the storage scalability, lossless compression is applied to transaction blocks and deduction blocks in the deposition zone by index, relative value, and other techniques to reduce the amount of block data.

5.2 Consensus Analysis

In Fractal Ledger, both the account chain and the deduction chain need to configure consensus protocols. In the deduction chain, contract transactions are confirmed when their deduction blocks are confirmed due to the need for producers to sequence them. In

[1] https://blog.infura.io/post/building-better-ethereum-infrastructure-48e76c94724b

the account chain, each user builds transaction blocks on their own account chain, and these transaction blocks are also confirmed among consensus nodes by a consensus protocol. It is noteworthy that when a fork occurs in the account chain, the fork can be resolved through the consensus of the deduction chain. In the experiment section of this paper, we adopt classic PBFT algorithm both in the account chain and the deduction chain, and a new consensus specifically proposed for Fractal Ledger will be reported in detail separately. The safety and liveness of deduction chain are retained according to PBFT. A transaction block is confirmed when consensus nodes collect more than $2f + 1$ votes within one round. Thus, communication complexity is lower than the classic PBFT and safety of the account chain is guaranteed. Once a transaction block is unconfirmed within one round, the transaction block can be confirmed through the consensus of the deduction chain. Thus, liveness is also guaranteed.

5.3 Comparison with Related Systems

Fractal Ledger's flexible and scalable structure enables it to scale similarly to chain-based system such as Cosmos [19] and Polkadot [20]. A fractal ledger with multiple chain scales can provide flexible structural support for the horizontal and vertical division of the ledger to fit different applications scenarios. Besides, the account-deduction chain acts as the bridge to connect different fractal units. Through the deduction mechanism, Fractal Ledger creates different fractal units, similar to parachains in Polkadot. The difference is that the consensus nodes of a fractal unit are specified artificially during the deduction process, instead of being divided by NPoS algorithm as in Polkadot. And each child fractal unit like Rollup [21-24] is able to perform computation and storage independently, and the results are anchored to the parent fractal unit periodically.

Compared to sharding solutions like Monoxide [25], Fractal Ledger can partition the network into different fractal units with its flexible structure. Various fractal units created by the deduction mechanism can also realize the functions similar to sharding, and the cross-fractal transactions are relayed through the deduction chain. A single fractal unit features high throughput and slow confirmation latency due to the graph-chain structure, so the overall performance is improved with the increased number of fractal units. Meanwhile, Fractal Ledger has more flexible structure scalability thanks to the deduction mechanism.

Compared with Nano [10], which also has a DAG-based structure, Fractal Ledger supports off-chain scaling and smart contracts, allowing the blockchain to accommodate various complex application scenarios.

6 RELATED WORKS

In recent years, a plethora of researches has been conducted to increase blockchain scalability. Generally, the research can be categorized into on-chain and off-chain solutions by whether they focus on or off the main blockchain.

On-chain scaling (Layer1 scaling) enhances processing efficiency by changing the underlying structure of the blockchain, including optimizing blockchain structure, data storage and other aspects.

Lombrozo et al. design the segregated witness (SegWit) [26] to increase the number of transactions carried by a block by migrating scripts and signatures (the witness data) in the extended block, and Wuille et al. [27] proposes to reduce the size of transactions with Schnorr Signature to improve storage efficiency.

Meanwhile, Sharding technology is currently the mainstream research direction of on-chain scaling, having been explored in both permissionless systems, *e.g.*, OmniLedger [28], Monoxide [25], Ethereum2 [29] and Elastico [30], and permissioned systems, *e.g.*, AHL [31], and RSCoin [32] to improve scalability. However, due to issues such as segmented user groups and the complicated cross-shard communications, sharding is still with low technological maturity and has not yet been widely adopted.

In terms of blockchain structure, DAG-based ledgers have recently emerged as potential solutions to on-chain scalability. For example, Spectre [7] and Phantom [8] aim to increase Bitcoin's throughput by replacing the chain-based structure to the DAG-based structure and merging blocks from different branches to the ledger. Nano [10] proposes a block-lattice structure to realize immediate and asynchronous processing by allowing users to maintain their own lightweight accounts.

To improve storage scalability of blockchain, Jidar [33] is a data reduction approach for Bitcoin system to allow users only store relevant data they are interested in and thus releases the storage pressure of each node. Yu et al. [34] propose Virtual Block Group (VBG), in which each node only needs to store part of block data and saves the VBG storage index to distributed hash table by taking block data as a resource, thus improving the storage and query efficiency of block data. On-chain storage scaling reduces node storage space requirements while ensuring data storage reliability and security on the assumption that block data is still stored on the blockchain, but there is also "time for space", which increases data acquisition time and transmission volume in the network when block data is requested from other nodes.

Off-chain solutions (Layer 2 scaling) are techniques that indirectly improve the scalability of blockchain by using external modules or systems to carry specific services, with the main directions being state channel, cross-chain, child-chain, etc.

State channel is used to process a large number of transactions by opening a channel independent of the chain and performing only critical clearing on the chain, reducing network congestion, fees, and delays, *e.g.*, Bitcoin-based Poon [35] and Ethereum-based Raiden network [36]. Cross-chain is the process of carrying a portion of the current chain's load by interacting with relatively independent external chains by means of notary, side-chain/relay [37], hash locking [35], and so on. Child-chain sets up several independently operating child-chains under the main chain. Plasma [38] and Rollup techniques, the latter of which can be categorized into Optimistic Rollups (*e.g.*, Arbitrum [21], Optimism [22]) and ZK-Rollups (*e.g.*, Loopring [23], Zk-sync [24]). It is worth noting that off-chain scaling techniques are primarily aimed at chain-based ledger systems; while graph-based

chains can achieve high on-chain concurrency performance, their off-chain scalability methods have not been found reported.

7 CONCLUSION

In this paper, we introduce Fractal Ledger, a new model that uses a graph-chain architecture to scale out blockchain system. A basic unit of Fractal Ledger includes an account lattice, which is graph-based and supports parallelized asynchronous on-chain transactions, and a deduction chain, which is chain-based and enables smart contract. This feature is missing in most DAG-based blockchains and provides the possibility for the interaction of different fractal units. In our experiments with up to 128 consensus nodes, a fractal unit achieves over 200BPS within 4s confirmation latency. The total BPS of Fractal Ledger grows linearly with the increased number of fractal units. Compared with Ethereum, using our designed folding scheme, we show that storage space for normal transfer transactions is reduced to below 2%, contributing to a sustainable storage solution in the long run. In future, the optimization of smart contract execution efficiency and the compression of contract data will be key directions for Fractal Ledger.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China No.2021YFB2700800.

REFERENCES

[1] S. Davidson, P. Filippi, J. Potts, "Disrupting governance: The new institutional economics of distributed ledger technology," Available at SSRN 2811995, 2016.
[2] Y. Liu, K. Qian, J. Chen, K. Wang, and L. He. "Effective Scaling of Blockchain Beyond Consensus Innovations and Moore's Law," 2020, arXiv: 2001.01865. [Online]. Available: https://arxiv.org/ abs/2001. 01865.
[3] J. Howell, "What Is Blockchain Sharding?" 2022. [Online]. Available: https://101blockchains.com/what-is-blockchain-sharding/. [Accessed: 2-January-2023].
[4] S. Roth, "An Introduction to Sidechains," 2022. [Online]. Available: https://www.coindesk.com/learn/an-introduction-to-sidechains/. [Accessed: 22-January-2023].
[5] PPIO,"Cross-Chains: How Blockchains Communicate With Each Other," 2019. [Online]. Available: https://medium.com/@ppio/ understanding-cross- chain-technology-e36b9c0cfaf3.
[6] Q. Wang, J. Yu, S. Chen, Y. Xiang, "SoK: Diving into DAG-based blockchain systems," 2020, arXiv: 2012.06128. [Online]. Available: https://arxiv.org/abs /2012.06128.
[7] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre: A fast and scalable cryptocurrency protocol," *IACR Cryptol*. ePrint Archive, vol. 2016, p. 1159, 2016.
[8] Y. Sompolinsky and A. Zohar, "Phantom: A scalable blockdag protocol," *IACR Cryptol*. ePrint Archive, vol. 2018, p. 104, 2018.
[9] C. Li, P. Li, D. Zhou, W. Xu, F. Long, and A. Yao, "Scaling nakamoto consensus to thousands of transactions per second," 2018, arXiv:1805.03870. [Online]. Available: https://arxiv.org/abs/1805.03870.
[10] C. LeMahieu, "Nano: A Feeless Distributed Cryptocurrency Network," White Paper, 2018. [Online]. Available: https://nano.org /en/ whitepaper.
[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," White Paper, 2008. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf.
[12] Bitcoin blockchain size. 2023. [Online]. Available: https://ycharts.com/indicators/bitcoin_blockchain_size. [Accessed: 02-February - 2023].
[13] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project Yellow Paper, vol. 151, pp. 1–32, Apr. 2014.
[14] Ethereum Chain Full Sync Data Size. 2023. [Online]. Available: https://ycharts.com/indicators/ethereum_chain_full_sync_data_size. [Accessed: 02-February-2023].
[15] J. Bonneau, I. Meckler, V. Rao, and E. Shapiro, "Mina: Decentralized Cryptocurrency at Scale," White Paper, 2020. [Online]. Available: https://docs.mina protocol.com/static/pdf/technicalWhitepaper.pdf.
[16] M. Castro and B. Liskov, "Practical byzantine fault tolerance," *in Proc.OSDI*, vol. 99, 1999, pp. 173−186.
[17] B. Yu, X. Li, H. Zhao. "PoW-BC: A PoW Consensus Protocol Based on Block Compression," *KSII Transactions on Internet and Information Systems*, 2021, 15(4):1389-1408. https://doi.org/10.3837/tiis.2021.04.011.
[18] J. Xie, F. Yu, T. Huang, et al., "A survey on the scalability of blockchain systems," *IEEE Network*, 2019, 33(5): 166-173.
[19] Cosmos. 2022. [Online]. Available: https://cosmos. network/whitepaper.
[20] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," Polkadot, White Paper, 2016.
[21] H. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: Scalable, private smart contracts," *in Proc. 27th USENIX Secur. Symp*. USENIX Secur., 2018, pp. 1353−1370.
[22] Optimism. 2022. [Online]. Available: https://community.optimism.io/. [Accessed: 12-Decemeber-2023].
[23] Loopring. 2022. [Online]. Available: https://loopring.org/#/. [Accessed: 12-Decemeber-2022].
[24] ZK-sync. 2022. [Online]. Available: https://docs.zksync.io/dev/. [Accessed: 12-Decemeber-2022].
[25] J. Wang and H. Wang, "Monoxide: Scale out blockchains with asynchronous consensus zones," *in Proc. 16th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2019, pp. 95−112.
[26] E. Lombrozo, J. Lau, and P. Wuille, "Segregated witness (consensus layer)," Bitcoin Core Develop. Team, Tech. Rep., 2015.
[27] BIP 340, 2022. [Online]. Available: https://github.com/bitcoin/bips/blob /master/bip-0340.mediawiki.
[28] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," *in Proc. IEEE Symp*. Secur. Privacy (SP), May 2018, pp. 583−598.
[29] "The Beacon Chain Ethereum 2.0 explainer you need to read first," 202. [Online]. Available: https://ethos.dev/beacon-chain/.
[30] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," *in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. -CCS*, 2016, pp. 17−30.
[31] D. Hung, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin, and Beng Chin Ooi, "Towards Scaling Blockchain Systems via Sharding," *In SIGMOD Int. Conf. on Management of Data*. ACM, 2019.
[32] G. Danezis and S. Meiklejohn. "Centrally Banked Cryptocurrencies," *In Network and Distributed System Security Symposium (NDSS)*, 2016.
[33] X. Dai, J. Xiao, W. Yang, C. Wang, and H. Jin, "Jidar: A jigsaw-likedata reduction approach without trust assumptions for bitcoin system," *in Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1317−1326.
[34] B. Yu, X. Li, H. Zhao. "Virtual Block Group: A Scalable Blockchain Model with Partial Node Storage and Distributed Hash Table". *The Computer Journal*, 2020,63(10) : 1524−1536.
[35] J. Poon and T. Dryja, "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments," White Paper, 2016. [Online]. Available: https://lightning. network/lightning-network-paper.pdf. [Accessed: 22-January-2023].
[36] Raiden Network. 2022. [Online]. Available: https://raiden-network.Readthe docs.io/en/stable/. [Accessed: 02-February-2023].
[37] J. Nick, A. Poelstra, and G. Sanders, "Liquid: A bitcoin sidechain,". White Paper, 2020, [Online]. Available: https://blockstream.com/assets/downloads/pdf /liquid -whitepaper.pdf. [Accessed: 22-January-2023].
[38] J. Poon, V. Buterin, "Plasma: Scalable autonomous smart contracts," White Paper, 2017, [Online]. Available: https://www.plasma.io/plasma-deprecated.pdf. [Accessed: 22-January-2023].