Second-degree branch structure blockchain expansion model

Qiang Liu, Baoyan Song and Junlu Wang

Abstract
The blockchain runs in a complex topological network which is affected by the principle of consensus, and data storage between nodes needs to maintain global consistency in the entire network, which causes the data storage inefficient. At the same time, the information exchange between large-scale communication node groups leads to the problems of bandwidth expropriation and excessive network load. In response to these problems, this article proposes a second-degree branch structure blockchain expansion model. First, a ternary storage structure is established. Data use the way of fully integrated storage, multi-cell storage, and fully split storage, and data are classified and stored in parallel between the structures. Second, a second-degree branch chain model is constructed. The main chain forks into multiple sub-chains, and a free competition chain structure and a Z-type chain structure are defined; a two-way rotation mechanism is introduced to realize the integration and transition between chain structures. Finally, a set of malicious attacks is simulated to realize the security constraints of the blockchain, to verify the security of the second-degree branch chain model. Experiment shows that the second-degree branch structure expansion model proposed in this article has great advantages in data storage efficiency and network load.

Keywords
Blockchain expansion, two-degree branch chain, ternary storage structure, two-way rotation mechanism, security constraints

Date received: 2 July 2021; accepted: 28 October 2021
Handling Editor: Peio Lopez Iturri

Introduction
Since the concept of blockchain was put forward, it has attracted widespread attention from the world. In terms of data storage, it is a distributed storage ledger; in terms of protocol, it is a decentralized consensus protocol; and in terms of economy, it is an Internet of value that improves cooperation efficiency. From the perspective of blockchain technology, blockchain data storage uses hash compression, asymmetric encryption, and other cryptographic principles to ensure reliability and to adopt distributed data storage through point-to-point connections. The blockchain ledger is jointly maintained by all nodes and stores data, which is based on a credible consensus mechanism. In recent years, the application of blockchain technology has become more and more extensive such as Bitcoin, Ethereum, and Litecoin, which uses computing power to prove, and it has gradually evolved from encrypted digital currency to a credible service platform applied to various industries in society. But with facing the challenges of complex environments, blockchain expansion problems have become more prominent.
Although the blockchain adopts expansion mechanisms such as Segregated Witness,\textsuperscript{10} Lightning Network (LN),\textsuperscript{11} and data sharding,\textsuperscript{12} as the scale and production speed of blockchain data continue to increase, the rate of storing is increasingly lagging behind the rate of real-time data production, which results in more serious data storage and network load problems in the blockchain. Therefore, how to further improve the storage efficiency and reduce the network load has become a difficult point in the current blockchain research.

In response to these problems, this article proposes a second-degree branch structure blockchain expansion model. The main contributions are as follows:

1. In view of the inefficiency of blockchain data storage, a ternary storage structure is designed to significantly improve data storage efficiency through data shunting and task distribution.

2. In the design of the actual structure of the blockchain, the second-degree branch chain model is proposed, including the free competition chain structure and the Z-type chain structure. The expansion of the blockchain is based on two chain structures.

3. On this basis, in response to the incompatibility of the second-degree chain due to structural differences, a two-way rotation mechanism is proposed to enable smooth switching between structural chains, and the double-chain structure fusion is specifically demonstrated through the second-degree chain fusion transition process.

4. Finally, the security of the model is further analyzed by the second-degree branch structure blockchain expansion model, and on the basis of the security analysis, to find out the security constraints which have instructive significance for the actual application of the blockchain.

Related works

At present, many scholars have conducted in-depth research on blockchain expansion technology and have achieved many research results. In the isolation authentication expansion mechanism,\textsuperscript{13} the block body extracts the signature information from the main chain space and stores it in a new data structure to achieve expansion, but the block space saved by this mechanism is limited. Seres et al.\textsuperscript{11} quantitatively analyze the structural characteristics of the LN and solve the expansion problem by improving the data throughput through multiple payment channels. But the LN topology needs to be improved, and the security needs to be strengthened. Min et al.\textsuperscript{14} propose a multicenter dynamic consensus mechanism in the permission chain, and by optimizing consensus mechanism, it reduces block confirmation delays to achieve expansion. But the dependence on master nodes is risky, and the system reliability is difficult to guarantee. Jia et al.\textsuperscript{15} propose a scalable blockchain model, which can achieve expansion by optimizing the storage structure and reducing the communication cost. But it requires high credibility of data storage nodes, which reduce the stability of the system.

Burchert et al.\textsuperscript{16} propose a new layer between the blockchain and payment channels. To deal with the expansion problem through the new layer of micro-payment channels, which achieves delay-free payment. Kim et al.\textsuperscript{17} propose a distributed storage blockchain (DSB) system, which improves storage efficiency by combining secret sharing, private key encryption, and information dispersal algorithms. But when peer failure occurs due to denial service attacks, DSB will cause serious communication cost. Fadhil et al.\textsuperscript{18} propose the Bitcoin Clustering Based Super Node (BCBSN) protocol, which reduces the transaction propagation delay by a reasonable ratio. But when the super node has a transaction failure, the normal transactions will be seriously affected. Zhao et al.\textsuperscript{19} propose a security strategy for DSBs, which can delete part of blockchains, so that nodes only store a part of blockchain. But due to the coexistence of multiple node modes, the operational burden of the management node is increased. Zhang et al.\textsuperscript{20} analyze blockchain transaction databases and propose a storage optimization scheme. The proposed scheme divides blockchain transaction database into cold zone and hot zone, and achieves storage optimization by moving unspent transaction outputs outside the in-memory transaction databases. But the data query call to the cold zone is inefficient. Shah et al.\textsuperscript{21} propose consensus-ADMM (alternating direction method of multiplier) based distributed optimization algorithm, which decomposes the optimization problem into sub-problems, and solves the sub-problem locally and exchanges information with the neighboring regions to solve for the global update.

In summary, this article conducts research on the basis of existing research, further optimizes the blockchain structure, and proposes a second-degree branch structure blockchain expansion model.

Second-degree branch model

Aiming at the construction of blockchain, first this article designs a ternary storage method to optimize the blockchain storage structure, then builds a second-degree branch chain mechanism for expansion, and uses a two-way rotation mechanism to carry out the transition of differentiated structure chains.
**Ternary storage structure**

After the blockchain is bifurcated into multiple sub-chains, the original blockchain information is distributed among the sub-chains for data storage, and the data storage rate has been significantly improved.

**Definition 1.** Blockchain data flow circulates through multiple channels, and it is stored in parallel at the same time, which is called data shunting. The increased block per unit time before the blockchain fork is \( N_{\text{block}} \), and the effective data amount of a single block body is \( D_{\text{valid}} \). So the data storage rate before the fork is \( N_{\text{block}} \times D_{\text{valid}} \). The blockchain classifies the stored data information \( D_{\text{valid}} \) as \( D_{\text{valid,1}}, \ldots, D_{\text{valid,n}} \). \( N \) sub-chains are formed after \( n \) bifurcation; the number of blocks increased per unit time in the sub-chain is \( N_{\text{block}} \), and the effective data amount from a single block body is \( D_{\text{valid,i}} \). Since data amount \( D_{\text{valid}} \) and \( D_{\text{valid,i}} \) tend to be same, after the blockchain is forked, the data storage rate is \( N_{\text{block}} \times (D_{\text{valid,1}} + \ldots + D_{\text{valid,n}}) \), which is close to \( n \times N_{\text{block}} \times D_{\text{valid}} \), and data shunting significantly improves the efficiency of blockchain storage.

After the blockchain is bifurcated into multiple sub-chains, at least one complete storage unit needs to be stored when the node data on the sub-chains are stored.

**Definition 2.** Each entire single chain that goes back to the genesis block is called a storage unit. When nodes store transaction information, it must be sure that the current block can be traced back to the genesis block of the blockchain. The storage unit composed of this entire single chain reflects the integrity and atomicity of the blockchain information.

A ternary accounting structure is proposed due to the node accounting method and environmental impact:

1. **Full integrated storage structure**: Each node records all the storage unit of the entire blockchain. Under this structure, a global view of the second-degree branch blockchain is obtained. This type of node is mostly the global management and service node, which is used to dynamically monitor the overall operation of the multi-tree blockchain and provides subsequent maintenance and update.

2. **Multi-cell storage structure**: Each node records a limited number of storage units and dynamically updates and stores them. A partial view of the second-degree branch blockchain is obtained. On one hand, this type of node is the local management and service node, and it is used to dynamically monitor and manage the local operation of the blockchain, and report to the blockchain global management service nodes about the current storage status; on the other hand, it is the information storage node of the blockchain and exists in the Z-type chain structure proposed below.

3. **Fully split storage structure**: Each node records its own separate storage unit. In this mode, the node is only responsible for the block storage of the sub-chain and does not care about the secure status of the chain where it is located. This type of node is the information storage node and exists in the free competition chain structure proposed below.

The ternary storage structure is shown in Figure 1:

**Figure 1.** Ternary storage structure.
structure chains use data shunting for effective expansion; on the other hand, they have different characteristics due to structural differences.

**Free competition chain structure.** In the free competition chain, the blockchain forms multiple sub-chains after branching. Sub-chains are independent of each other. When the storage node downloads the sub-chain data, it must store a complete storage unit, which reflects the good traceability and integrity of the blockchain. The sub-chain only packs information on this chain and does not interact with other sub-chains. Due to the competition between the subchains, the distribution of computing power resources will be obviously uneven, which is not conducive to the stability of the blockchain, so introducing the system default allocation to balance the computing power gap.

**Definition 3**

System default allocation. Blockchain computing power distribution is different, which will cause the storage rate of each sub-chain to be significantly different, and then generates security problems due to malicious attacks. The system generally allocates newly added accounting nodes to the weaker computing power chain to maintain the safety and reliability of the blockchain. This process is allocated by the system by default.

The schematic diagram of the free competition chain structure is shown in Figure 2. “Free-competition” represents the free competition chain structure. Each single chain creates blocks in a top-down order, and the child blocks store the hash value of the parent block. The single chain does not affect each other, and the single chain with weak computing power will be allocated by the system to supply the computing power by default, so that the overall blockchain computing power tends to be consistent. The \((x, y)\) form coordinate uniquely identifies the position of block.

In the free competition chain structure, the node which stores the block only needs to download a storage unit to carry out data storage services. In all chain structures, the storage overhead is the smallest, but it will face the secure issues brought by the shunting of computing power, which will be discussed later. The chain type Construct field of the stored block header information is “Free-competition.”

**Z-type chain structure.** After the Z-type chain is branched, information is stored in the blocks on each child chain. In addition to storing the hash value of the parent block, it is also necessary to store the hash value of the adjacent block, which is the pseudo-hash value.

**Definition 4.** Pseudo-hash. In the Z-type chain structure, the sequence of block generation is from left to right, from top to bottom, and the hash value of the block generated in the previous order is the pseudo-hash.

As shown in Figure 3, unlike the free competition chain structure, blocks are generated sequentially from top to bottom on the child chain. All nodes on the Z-type chain structure save the hash value of the parent block, store the hash value of the adjacent block from the left of the same layer (or the rightmost block of the upper layer), and then perform the block data storage process. Therefore, the block generation sequence under the entire chain structure is from left to right and from top to bottom.

In the Z-type chain structure, honest computing power is concentrated to sequentially store and generate blocks, which avoids the advantages of centralized malicious attacks on one block and significantly improves the security of blockchain. The Construct field of the structure header information is “Z-type,” and a new hash supplement field HashExtra is added. This structure HashExtra is “Fakehash.” Finally, a block location field Location is defined.

**Definition 5.** Location field is located by a two-dimensional coordinate \([Xline,Yrow]\). Xline represents
the row index of the block in Z-type chain structure, and $X_{row}$ represents the column index of the block in the Z-type chain structure. Through the Location field of the blockchain, each sub-chain block can get the block where the pseudo-hash is located.

**Inter-chain rotation transition**

Due to the structural differences between the two chain structures, compatibility problems can be caused when the chain structure is switched. This article proposes a two-way rotation mechanism to make the transition between different chain structures smoothly and dynamically to display the state changes of the chain structure.

**Second-degree chain integration transition.** The second-degree branch structure blockchain model is accompanied by the fusion between the two chain structures. When the blockchain transits between the chain structures, the corresponding attributes of the blocks will change. The sub-chain will form another chain after the fork, and the structure will be accompanied by this change.

On the premise that other fields are omitted except for the original block timestamp, random number, and Merkle Root, the new block information in this article is shown in Table 1.

In Figure 4, first, the initial block is bifurcated through a free competition chain structure, and then each sub-chain is further bifurcated through a Z-type chain structure. In this process, the corresponding

---

**Table 1. New block information table.**

| Block field | Meaning |
|-------------|---------|
| Prehash     | The parent block hash on the same child chain |
| HashExtra   | Two additional hash types with chain structure (null, Fake-hash) |
| Construct   | Double chain type (Free-competition, Z-type) |
| Location    | The position of the block in the fork chain $[X_{line},Y_{row}]$ |

**Two-way rotation mechanism.** In the second-degree block-chain, it is inevitable that one chain structure will become another chain structure after chain forks, and there will be the transition between the following chain structure: free competition chain and Z-type chain.

**Two-way rotation between the free competition chain and the Z-type chain.** When the free competition chain rotates to the Z-type chain, the data in the free competition chain are allocated to each sub-chain of the Z-type chain in an orderly manner. The node on the sub-chain cannot only store the storage unit of local sub-chain, because the storage of block information requires pseudo-hash provided by other sub-chains. So, the storage units of all sub-chains after the fork should be stored. The corresponding Construct field changes from “Free-competition” to “Z-type,” and the HashExtra field is from “null” to “Fakehash.” When the Z-type chain is rotated to the free competition chain, the data tasks stored in the Z-type chain are orderly distributed to free competition sub-chain. Nodes on the free competition chain only need to store one storage unit where the sub-chain is located (without interference from the hash value of other sub-chains), and the corresponding Construct field changes from “Z-type” to “Free-competition”; HashExtra field changes from “Fakehash” to “null.”

---

![Schematic diagram of Z-type chain structure.](image3)
attributes in the blocks inevitably change. Table 2 lists the block attribute changes based on Table 1.

In Table 2, the state changes of the fusion transition from the two chain structures are dynamically displayed. With the change of the chain structure, the attribute fields of the corresponding blocks have also undergone necessary changes. It changes the sub-chains structure and achieves the smooth transition.

Model safety analysis and constraints

Based on the two-degree branch structure blockchain expansion model proposed above, the expansion model is further analyzed in terms of security and based on the security analysis to construct a security constraint.

Safety analysis of free competition chain structure

Each sub-chain in the free competition chain structure runs independently, and the system defaults that the computing power among the sub-chains is uniform (which the honest computing power is evenly distributed to each branch sub-chain). Malicious nodes conduct malicious attacks on the forked single chain, as shown in Figure 5. In this case, the security analysis of the free competition chain is carried out.

Supposing the workload of the entire blockchain model is 1, the proportion of malicious computing power is \( q \), and the chain is bifurcated into \( n \) sub-chains. The malicious nodes compete with the honest

Table 2. Three-chain fusion transition table.

| Head parameter | Genesis block header | Free-C chain 1 root block | Free-C chain 2 root blocks | Free-C chain block |
|----------------|---------------------|--------------------------|---------------------------|-------------------|
| Prehash        | Null                | ✓                        | ✓                         | ✓                 |
| HashExtra      | Null                | ✓                        | ✓                         | ✓                 |
| Construct      | Null                | ✓ Free-C                 | ✓ Free-C                  | ✓ Free-C          |
| Location       | Null                | (X1start-F, Y1start-F)   | (X2start-F, Y2start-F)    | (X1-F, Y1-F)      |
|                | Z-type chain 2      | Z-type chain 2           | Z-type chain 2 blocks     | Z-type chain 2    |
|                | tail blocks         | root blocks              |                           | tail blocks       |
|                | ✓                   | ✓                        | ✓                         | ✓                 |
|                | Null                | ✓ Free-C                 | ✓ Free-C                  | ✓ Free-C          |
|                | ✓                   | Z-type                   | Z-type                    | Z-type            |
|                | ✓                   | FakeHash                 | FakeHash                  | FakeHash          |
|                | ✓                   | Z-type                   | Z-type                    | Z-type            |
|                | ✓                   | (X2-Z, Y2-Z)             | (X2-Z, Y2-Z)              | (X2-Z, Y2-Z)      |
|                | ✓                   | (X2end-F, Y2end-F)       | (X2end-F, Y2end-F)        | (X2end-Z, Y2end-Z)|
nodes in one sub-chain, and \( z \) represents the number of blocks that malicious nodes are chasing honest node. At this time, the computing power of the honest nodes attacked by the malicious node is \( (1 - q)/n \), the percentage of the malicious computing power is \( q = q/(1 - q)/n + q\), and the percentage of the computing power from the honest node is \( p = 1 - q \). Through the gambler bankruptcy model, a gambler can gamble countless times, and it tries to fill up the shortfall created. The probability that the gambler fills up the shortfall is the probability that the attacker catches up with the honesty nodes. As shown in equation (1)

\[
\begin{align*}
p(1) &= q1 + p1 \times p(2) \\
p(2) &= p(1)^2 \\
p(z) &= p(1)^z \\
\rightarrow p(1) &= q1/p1 = q/n(1 - q)
\end{align*}
\]

The probability of success of a malicious node filling \( z \) blocks is shown in equation (2)

\[
p(z) = \begin{cases} 
1, & q1 \geq p1 \\
(q1)^z/p1, & q1 < p1 
\end{cases}
\] (2)

where \( q1 = q/(1 - q)/n \) and \( p1 = 1 - q1 \). Using the transfer confirmation chase model, the success rate of malicious node attack is shown in equation (3)

\[
p = \sum_{k=0}^{k-1} \frac{\lambda^k \times e^{-\lambda}}{k!} \times \left\{ \begin{array}{ll} 
1, & q1 \geq p1 \\
(q1)^z/p1, & q1 < p1 
\end{array} \right. 
\] (3)

Because \( \lambda = (q1/p1) \times z \), equation (4) is obtained after equation (3) is organized

\[
p = 1 - \sum_{k=0}^{k-1} \frac{\lambda^k \times e^{-\lambda}}{k!} \times \left( 1 - \left( \frac{q1}{p1} \right)^z \right) 
\] (4)

The description of the functional relationship in the free chain structure is shown in Figures 6 and 7.

In Figure 6, the malicious computing power accounts for a constant 0.1 (i.e. 10%), and the relationship between the two variables \( z \) and the malicious node attack probability \( p \) is obtained. Setting the number \( z \) of blocks confirmed by honest nodes to 1, 3, 5, 7, 9, 11, 13, 15, and 17, and setting the forked number \( n \) in the chain to 1, 2, 4, and 6. When the malicious computing power \( q \) is constant, each function line decreases downward, and the success rate \( p \) from the upper function line is higher than the lower function line.

In Figure 7, the number of bifurcations of the chain structure is set to be constant, and the relationship between two different variables \( q \) and \( z \) and the success probability of malicious node attack \( p \) is studied. The block number \( z \) confirmed by honest nodes is set to 1, 3, 5, 7, 9, 11, 13, 15, and 17 respectively, and the percentage of computing power of malicious nodes is set to 0.2, 0.1, 0.05 and 0.01 respectively. Figure 7 is obtained by law statistics. Each function line downward and the
success probability $p$ of the upper function line is always higher than the success probability of the lower function line. The function analysis is as follows:

1. The proportion of malicious computing power $q$ and the number of forks $n$ remain unchanged, and the probability of successful malicious attack $p$ decreases monotonically with the block $z$ confirmed by the honest chain.

2. The percentage of computing power $q$ and the block $z$ confirmed by the honest chain remain unchanged, and the probability of successful malicious node attack $p$ increases monotonically with the number of forks $n$.

3. The number of forks $n$ and the block $z$ confirmed remain unchanged, and the success rate of malicious attack $p$ increases monotonically with the proportion of malicious computing power $q$.

Combining with the inference, we can get that in order to ensure the safety and reliability of the blockchain, the measures that can be taken are to reduce the forks of the blockchain, increase the proportion of honest computing power, and wait for as many blocks confirmed as possible.

**Safety analysis of Z-type chain structure**

When all the sub-chains on the Z-type chain store the data, the forked sub-chains logically connect into a Z-type pseudo-chain, and the chain structure safety feasibility analysis is carried out on the characteristics of the Z-type chain structure.

The Z-type chain block packing sequence is from left to right and top to bottom, and all honest computing power is concentrated for data storage on the Z-type chain. However, as each block is packaged, the hash value from the parent block and the pseudo-hash value from other sub-chain blocks need to be recorded at the same time. When malicious node attacks a sub-chain, it must wait for the pseudo-hash provided by other sub-chains at the same time and then competes with the honest nodes. In this case, to perform security analysis on the sub-chain. Figure 8 is a diagram of the attack mode from the Z-type chain.

It can be seen from Figure 8 that the blocks on the Z-type chain structure are generated in an orderly manner, which is logically a single chain structure. Therefore, the success rate of the double-spending attack has nothing to do with the number of forks $n$. Only the relationship between the variables $q, z$ and the success probability of malicious node attack is

Combining with the inference, we can get that in order to ensure the safety and reliability of the blockchain, the measures that can be taken are to reduce the forks of the blockchain, increase the proportion of honest computing power, and wait for as many blocks confirmed as possible.
considered, setting the block number $z$ confirmed by honest nodes to 1, 3, 5, 7, 9, 11, 13, 15, and 17, and setting the percentage of malicious computing power in the Z-type chain to 0.2, 0.1, 0.05, and 0.01. Figure 9 is obtained by law statistics. Each function line downward and the success probability $p$ of the upper function lines are always higher than the lower function lines. The analysis of the combination function to the Z-type chain structure is as follows:

1. The proportion of malicious computing power $q$ remains unchanged, and the probability of successful malicious node attack $p$ decreases monotonically with the block $z$ confirmed by the honest chain.

2. The block $z$ confirmed by the honest chain remains unchanged, and the success rate of malicious attack $p$ increases monotonically with the proportion of malicious computing power $q$.

Combined with the inference, it can be obtained that the security of the chain structure has nothing to do with the number of forked sub-chains. Against malicious attacks, the measures which can be taken are increasing the proportion of honest computing power and waiting for as many block confirmations as possible in the transaction.

Second-degree chain security constraints

The safety analysis of the two chain structures is carried out above. Among them, the second-degree chain should meet the safety constraints in practical applications and construct the safety constraints based on the safety analysis in the chain structure.

The relationship model in Figure 10 satisfies the security constraints of the blockchain. Among them, as the confirming number increases, the highest number of forks also increases accordingly. In practical applications, the security between the number of sub-chains and the number of confirmed blocks should be considered before the blockchain forks. For example, with the rapid increase in the scale of data, the second-degree branch structure blockchain expansion is carried out on the blockchain. The system determines to adopt the free competition chain to expand, and then the blockchain is transformed from a master chain into five branches. In order to prevent malicious double-spending attacks, the sub-chains should wait at least two confirmation blocks after the fork to ensure that the transaction will not be double-spending and then conduct normal transactions.
The influence in success probability of double-spending attacks between the block number confirmed and the proportion of malicious computing power can be obtained from the table. For example, chains with weak computing power are vulnerable to malicious attacks, and the system determines to adopt the Z-type chain to expand. If the proportion of malicious computing power is less than 1%, after the chain forks, in order to prevent malicious double-spending attacks, the sub-chains should wait for at least two confirmed blocks (or more confirmed blocks) to ensure that the transaction will not be double-spending and then perform normal transactions.

**Experiment and analysis**

The experimental environment is 20 servers with 32-core CPU, 128G memory, and 10T storage space. Docker virtualization technology is used to deploy three-dimensional chain nodes, Kubernetes is used to manage docker clusters, and the servers use gigabit networks and communicate between containers through flannel technology. Twenty hosts with TestRPC installed are equivalent to 20 master nodes NodeDB. NodeDB is mainly responsible for managing and maintaining the internal container nodes, and the internal nodes exchange information through NodeDB. The experimental design architecture diagram is shown in Figure 11.

In this experiment, there are 20 main nodes NodeDB. Each NodeDB has about 10 container nodes, and the network is composed of 250 nodes. The second-degree branch chain branches into four sub-chains for expansion. The blockchain uses the proof of work (POW) consensus mechanism and does not use random numbers for verification, which reduces the amount of data transmission by broadcasting the block header data, and adds additional information to verify the consistency of transaction information. The experimental comparison mechanisms to the two-degree branch chain (TDBC) are the Segregated Witness Expansion (SWE) mechanism and the directed acyclic graph (DAG) expansion mechanism, to compare and analyze in terms of transaction per second (TPS), communication overhead, confirmation delay, and effective data rate.

**Storage rate**

In the second-degree branch chain TDBC, each node only broadcasts its own block header information to the entire network, and the transaction information can be packaged in the block after verifying the valid transaction identifier. Without the verification process, the transaction volume is compressed and stored in the block body, which expands the upper limit of data throughput. The TDBC mechanism effectively shunts data, and tasks are allocated to each sub-chain for parallel storage. The DAG mechanism uses a network structure in the data structure, and the blockchain consensus is converted from the longest chain consensus to
the heaviest chain consensus, which retains a certain degree of independence and autonomy of the local network, and allows the parallel creation of blocks. The SWE mechanism isolates the transaction signature from the storing block, and the extra space is used for the storage of transaction information. In the experiment, this article deploys an experimental environment with a master node of 20, and tests the data throughput per unit time under the three expansion mechanisms in different periods. The final figure is shown in Figure 12.

Through analysis, due to shunting and paralleling storage in TDBC, TDBC blockchain storage rate reaches nearly 10,000 transactions per second at the peak, which is much higher than the 100 transactions per second in SWE and the nearly 1000 transactions per second in DAG. Therefore, the advantage of TDBC in data storage throughput rate is obvious.

**Communication overhead**

With the increasing of communication nodes, the communication channels have increased exponentially, and the number of communication nodes has become the main factor restricting communication overhead. In the TDBC mechanism, the blockchain information is distributed to each sub-chain in an orderly manner. After the internal container node recognizes that the block belongs to the sub-chain, data communication is carried out; otherwise, it does not respond. The DAG mechanism adopts a directed acyclic structure. Due to the independence of the local blockchain, the blocks are allowed to be created in parallel. So, the blocks can be confirmed without the recognition of all nodes. The SWE mechanism broadcasts block information to the entire network before it is confirmed by the entire network. If a conflict block occurs, it must enter the second confirmation. This experiment sets up an experimental environment of 10 and 20 main nodes. In this experiment, some communication nodes may fail; the number of communication nodes required directly reflects the communication overhead of the blockchain.

As shown in Figure 13(a), TDBC confirms that the average number of communication nodes required is about 30, which is significantly lower than the number that is about 60 nodes in DAG and about 100 nodes in SWE. It can be seen that the data shunting and task classification in TDBC reduce the communication overhead greatly. In Figure 13(b), the number of communication nodes in the TDBC mechanism is still stable and keeps at a minimum level, which reflects the good expansion of TDBC.

In the information exchange of communication nodes, the number of valid transactions carried by the global block in each communication also reflects the communication efficiency. In Figure 14, the size of the global block in the TDBC and SWE mechanisms is relatively stable, because DAG mechanism does not support strong consistency and the block size has obvious volatility, which brings instability factors to the communication of the blockchain.

**Confirmation delay**

In the blockchain, the transaction information is written into the block body after validity verification. The block body generates the broadcast header information,
and the node transmits the block header information to other nodes for verification; finally, it gets verified by other nodes in the entire network. This process is confirmation delay. In TDBC, the information stored in the block is strictly classified, and the storage node only stores the block data on the sub-chain and does not have to be submitted to all nodes in the entire network for verification. In the DAG mechanism, it is affected by the independence of the blockchain local network where blocks are allowed to be verified by nodes in the local network. The SWE mechanism has relatively high requirements for network verification nodes, and blocks need to be confirmed by the entire network. In this experiment, to determine the impact of different blockchain network scales on the confirmation delay, the 5, 10, and 20 NodeDB experimental environments are, respectively, deployed. The abscissa is different periods, and the ordinate is the statistical confirmation delay range at this stage. Three types of average block confirmation delay are obtained under the mechanism.

Figure 15 shows that the average deterministic delay of TDBC is about (8,10,13), the average deterministic delay of SWE is about (10,14,20), and the average deterministic delay of DAG is about (9,12,16). It can be obtained that the confirmation delay of the TDBC blockchain that uses data shunting and task allocation is less than the other two mechanisms, and as the network scale becomes larger, the increase of the confirmation delay from the TDBC mechanism is also slightly less than DAG and SWE mechanisms.

**Effective data rate**

The transaction information generated is validated and stored in the block body. The blocks are verified by the entire network nodes and then stored on the blockchain, which are valid blocks. But the transaction data from generating to the incoming chain inevitably produce data loss such as network instability which leads to data packet loss, network congestion which leads to data cache being cleared, and invalid data caused by double-spending attacks. The blockchain adopts the TDBC mechanism to strictly classify data, and the blockchain network is orderly and not redundant, which greatly alleviates network congestion. The blockchain using the DAG mechanism adopts a directive acyclic graph and allows the parallel generation of blocks. But there will be repeatable valid transaction information in the parallel blocks, and the block storage space is wasted to a certain extent. For blockchains using the SWE mechanism, the degree of network congestion is the main factor that restricts the effective data rate in the blockchain. In this experiment, four blockchain environments including 5, 10, 15, and 20 main nodes are deployed, and the effective data rate is reflected by the ratio of the effective data amount to the total data amount.

Figure 16 is obtained through experiments. Through analysis, the data efficiency of the TDBC mechanism is better than the other mechanisms, and with the increase of the blockchain nodes, the data efficiency is always relatively stable. The DAG mechanism data efficiency is better than SWE mechanism in a long term. The data

![Figure 14. Global block volatility.](image1)

![Figure 15. Determine the delay.](image2)
efficiency of SWE mechanism is higher in the early stage, and with the expansion of the blockchain network, its data effectiveness has significantly decreased.

Conclusion and future work

This article conducts an in-depth study on the deficiencies of the existing blockchain system expansion and proposes a second-degree branch structure blockchain expansion model. In order to improve data storage efficiency, the model optimizes the data structure through ternary storage. A second-degree branch chain model is constructed: free competition chain structure and Z-type chain structure. The model expands the blockchain that alleviates the communication burden on the network. To merge and transit the two structural chains through the two-way rotation mechanism, which ensures the stability of the blockchain expansion. On the basis of the simulative malicious attacks on the blockchain, putting forward the safety constraints to ensure the security of the expansion. And finally, to realize the effective expansion in the blockchain.

The second-degree branch structure blockchain expansion model in this article is mainly researched under the POW consensus. The future work is to study the expansion to other mainstream blockchains such as proof of stake (POS) blockchain and delegated proof of stake (DPOS) blockchain. In addition, with the rapid increase of blockchain transactions, the huge global storage ledger of the second-degree branch chain has problems in the management and maintenance for the blockchain. Another future work is to study the global ledger of the second-degree branch chain and, through distributed services, to further optimize the storage structure.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded by the China Postdoctoral Fund General Project (Grant No. 2020M672134), the National Natural Science Foundation (Grant No. 61502215 and Grant No. 51704138), and the Scientific Research Project of Department of Education of Liaoning Province (Grant No. LJC201913).

ORCID iD

Qiang Liu https://orcid.org/0000-0002-0046-7001

References

1. Yin HHS, Langenheldt K, Harlev M, et al. Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the Bitcoin blockchain. J Manage Inform Syst 2019; 36(1): 37–73.
2. Ahmed M and Pathan ASK. Blockchain: can it be trusted? Computer 2020; 53(4): 31–35.
3. Hamandawana P, Khan A, Lee CG, et al. Crocus: enabling computing resource orchestration for inline cluster-wide deduplication on scalable storage systems. IEEE T Parall Distr 2020; 31(8): 1740–1753.
4. Hu P, Sung CW and Chan TH. Capacity of wireless distributed storage systems with broadcast repair. IEEE T Commun 2019; 67(2): 1004–1017.
5. Wang Z, Yu X, Mu Y, et al. A distributed Peer-to-Peer energy transaction method for diversified prosumers in Urban Community Microgrid System. Appl Energ 2020; 260: 114327.
6. Wu J, Dong M, Ota K, et al. Application-aware consensus management for software-defined intelligent blockchain in IoT. IEEE Network 2020; 34(1): 69–75.
7. Wang Y, Tang C, Lin F, et al. Pool strategies selection in PoW-based blockchain networks: game-theoretic analysis. IEEE Access 2019; 7: 8427–8436.
8. Yaqoob I, Salah K, Uddin M, et al. Blockchain for digital twins: recent advances and future research challenges. IEEE Network 2020; 34: 290–298.
9. Gul MJ, Rehman A, Paul A, et al. Blockchain expansion to secure assets with fog node on special duty. Soft Comput 2020; 24: 15209–15221.
10. Nguyen J. The risks of segregated witness: problems under electronic contract and evidence laws. Comput Internet Law 2017; 34(11): 1–8.
11. Seres IA, Gulyás L, Nagy DA, et al. Topological analysis of Bitcoin’s Lightning Network. In: Pardalos P, Kotsireas I, Guo Y, et al. (eds) Mathematical research for blockchain economy. Cham: Springer, 2020, pp.1–12.
12. Chen H and Wang Y. SSChain: a full sharding protocol for public blockchain without data migration overhead. Pervasive Mob Comput 2019; 59: 101055.
13. Wright CS. The illusion of scale in segregated witness. Rochester, NY: Social Science Electronic Publishing, 2017.
14. Min X, Li Q, Kong L, et al. License chain multi-center consensus mechanism. Chin J Comput 2018; 41(5): 1005–1020.
15. Jia D, Xin J, Wang Z, et al. Blockchain storage capacity scalable model. J Front Comput Sci Technol 2018; 12(4): 525–535.
16. Burchert C, Decker C and Wattenhofer R. Scalable funding of Bitcoin micropayment channel networks. Roy Soc Open Sci 2018; 5: 180089.
17. Kim Y, Raman R, Kim Y-S, et al. Efficient local secret sharing for distributed blockchain systems. IEEE Commun Lett 2019; 23: 282–285.
18. Fadhil M, Owenson G and Adda M. A Bitcoin model for evaluation of clustering to improve propagation delay in Bitcoin network. In: Proceedings of the 2016 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC) and 15th international symposium on distributed computing and applications for business engineering (DCABES), Paris, 24–26 August 2016. Washington, DC: IEEE Computer Society.
19. Zhao P, Cheng H, Fang Y, et al. A secure storage strategy for blockchain based on MCMC algorithm. IEEE Access 2020; 8: 160815–160824.
20. Zhang J, Zhong S, Wang J, et al. A storage optimization scheme for blockchain transaction databases. Comput Syst Sci Eng 2021; 36(3): 521–535.
21. Shah C, Annoni J and Wies R. Consensus-based ADMM for distributed optimization in power systems using private blockchain network, 2021, https://doi.org/10.36227/techrxiv.13507176.v1