Blockchain periodic storage space recovery method based on blockchain consensus algorithm

Blockchain periodic storage space recovery method based on blockchain consensus algorithm

Hongping Cao*

Nanfang College of Sun Yat-sen University, Guangdong, 510000, China

*Corresponding author: caohongping@hotmail.com

Abstract. Decentralized encryption digital cash has been proved to be a successful application mode, and the users of the application can also be the maintainers of this application. To solve the problem of fast growth of blockchain data on Ethereum platform, this paper proposes a method to recycle the periodic storage space of blockchain deployed on blockchain based on blockchain consensus algorithm. On the principle of decentralization, the problem of mutual trust between nodes is solved. And that stability value of the update storage node is recorded, and the high-stability node is selecte to store the newly generated data copy, so that the stability of data storage is improved. Finally, some bytecode fragments are replaced by a suitable algorithm. Tests show that the massive small file system supporting file deletion has no obvious decline in file reading and writing performance compared with the original one, and the storage space of smart contracts can be saved by nearly 46%.

Keywords: Blockchain consensus algorithm; Periodic storage; Space recovery

1. Introduction

Blockchain first appeared in the paper Bitcoin: A Peer-to-Peer Electronic Cash System published by Satoshi Nakamoto. Its realizability has been proved by Bitcoin which has been running up to now. The decentralized structure of blockchain [1], open and transparent data, and unchangeable features have attracted the attention of governments, corporate giants and startups in various countries, and have become the focus of research and capital. Blockchain technology has effectively solved the consensus problem in Byzantine general problem by means of data encryption, timestamp, distributed consensus and economic incentives [2-3].

Traditional distributed cloud storage systems are represented by GFS and HDFS[4-5]. They are designed and provided by a few large storage service providers. If the storage service providers are malicious, the interests of the participants will be damaged. Due to the lack of specific programming scripts and credible running environment in the initial stage of the proposal, smart contracts have not received wide attention. The emergence of blockchain technology and its natural characteristics for smart contracts redefine smart contracts. In the alliance chain system, once the transaction is packaged, it can be considered irreversible. However, with the increase of nodes in the system, the amount of communication needed in the system will also increase in order to reach a consensus, which does not

have good scalability. In the literature [], the storage space of circular queue is dynamically increased [6]. However, if the storage space is idle after being opened and used, and the user is not sure that the idle space will be used in the subsequent operation, it is necessary to find a way to recycle this part of space.

In order to increase the storage scalability of blockchain technology, this paper proposes a new scheme based on the current design idea of distributed cloud storage system based on blockchain. Based on the blockchain consensus algorithm, this scheme stores data in the local storage space of nodes instead of directly in blocks, and optimizes the model by adopting a data copy allocation strategy. A method of recycling the idle storage space in the circular queue is implemented, which successfully avoids the waste of storage space in the circular queue.

2. Algorithm design

Literature [7] has realized a space-saving and reasonable method to join the queue. These systems only support insert and query based on key value, but do not support relational operation of complex query. At first, ethereum client sends call contract message to ethereum node; Then, ethereum node inputs the message into the contract of local ethereum virtual machine, and executes the message according to the contract rules to get the result; Together with the block information, it is transmitted to the next node, and the subsequent nodes verify the block information first, and then sign the blind information of each node.

Blind signature is blind, which can effectively protect the specific content signed, and is widely used in the implementation of electronic election [8]. Voters first blind their votes, and then let the signer sign the blind information. In the voting system, the blind information is the ballot paper.

Blind processing of signatures:

Blind message: $m^{'} = mr^{e} \left(\mathrm{mod}\ N\right)$

Signature message: $s^{'} = \left(m^{'}\right)^{d} \left(\mathrm{mod}\ N\right)$

Blind removal information: $s\ \ = s^{'} \times r^{-1} \left(\mathrm{mod}\ N\right)$

The principle is as follows:

$r^{ed} \equiv r \left(\mathrm{mod}\ N\right)$

$s \equiv s^{'} \times r^{-1} \equiv \left(m^{'}\right)^{d} r^{-1} \equiv m^{d} r^{ed} r^{-1} \equiv m^{d} r r^{-1} \equiv m^{d} \left(\mathrm{mod}\ N\right)$ Nodes on ethereum platform can freely join or exit the blockchain network, and can only initiate transactions after synchronizing all the data in the blockchain. At the initial stage of ethereum platform operation, the amount of data in the blockchain is very small, so nodes can complete data synchronization in a short time. According to the protocol, a random number meeting the rules is obtained through operation, and the node that calculates the result first obtains the bookkeeping right at this time, and sends out the data to be recorded in this round, which is stored together with other nodes in the whole network after verification. It can be regarded as a distributed database, which only allows data to be modified and inserted by adding, and does not allow deletion.

3. Periodic storage space recovery method

3.1 Opcode definition

A variety of opcodes are defined in ethereum to represent the running process of intelligent contract in EVM [9], and a new opcode COMPRESS is defined in this paper by imitating CALLCODE. Its core idea is to divide a complete blockchain into several parts and store them in the system. Each blockchain is managed by specific members or relevant institutions, and the use of blockchain also has a specific business environment. If the request has been processed at the replica node, the replica will resend the execution result to the client. If the request has not been processed at the replica node, the replica node will forward the request to the master node. Therefore, file deletion in SMDFS can be divided into two

stages: 1) deleting metadata and recording metadata deletion behavior; 2) At a certain time point, the storage space is sorted according to the deletion behavior log, and the storage space occupied by invalid files is released.

Authorization is required to join the license chain, and nodes cannot register and exit at will, even the whole chain is controlled by a single organization, so the license chain is centralized. The user node is the owner of the original data, the storage node is the keeper of the copy, and the verification node is the verifier of the stability of the storage node. When the smart contract is executed, EVM will restore the pointer to the specific bytecode after detecting the opcode COMPRESS, and then continue to execute downward. The fair allocation scheme of deposit after contract destruction is given, and the suggestion of data partitioning is put forward according to the contract test results.

3.2 Data compression

When the blockchain storage capacity can be expanded to store data, the model uses POR data retrievability proof method to encrypt the blocks in the user node blockchain, and obtains the corresponding ciphertext and key [10]. Under the traditional method, we can only ensure that the authority to modify sensitive data is only in the hands of senior managers through authority management [11]. Therefore, the same small file in the data file may occupy multiple storage spaces, so the small file record package uses the combination of "small file pathname _ offset" as an index item to identify each storage space in the data file. Only when the user inquires on the chain, the stored proof is returned, thus reducing unnecessary proof operations.

The problem of computing the same bytecode fragments between adjacent contracts in an intelligent contract sequence can be considered as a simple extension of finding the Longest Common Substring (LCS) of a string. This problem can be defined as follows:

Given an intelligent contract sequence $\{C_0, C_1, \cdots, C_{n-1}, C_n\}$, in order to compress the intelligent contract C_i in it, it is necessary to calculate the common continuous substring LCS of each intelligent contract bytecode in C_i and $\{C_{i-k}, C_{i-2}, \cdots, C_{i-1}\} i-k \geq 0\}$.

Given $\sum$ represents a set of strings, given two strings $s_1, s_2 \in \sum$, their common continuous substring set ω_1, ω_2 can be defined as:

$$LCS = (s_1, s_2) = \left\{ z \in \sum^* \right\} s_1 = x_1 z x_2, s_2 = y_1 z y_2, x, y_1 , y_2 \in \sum^* \left. \right\}$$

Each node in the blockchain network keeps a list, and the address of a node in the alliance appears on the list, which indicates that these nodes are an alliance. In the big data analysis scenario, the data operation to be processed by the system is the continuous reading and writing of block data, and the query optimization strategy of the traditional database management system cannot bring about obvious performance improvement. If the levenshtein between contracts is high, add class tags to smart contracts by manually reading README and other information, otherwise add class tags to smart contracts by reading source code directly; If it is determined that the number of nodes in one's own union has not reached the upper limit, attach a time stamp to the joining information and the public key S of the node, encrypt it with P, and transmit it in the form of a message, indicating that the node has joined the union.

In this paper, a scalable model of blockchain storage capacity is proposed by using distributed storage method. As shown in fig. 1.

In the existing blockchain technology, an attacker who wants to tamper with data needs to control more than 50% of the nodes in the network. After distributed storage of blockchain, the number of copies of blockchain in the network is reduced, and attackers can modify blockchain data while controlling less than 50% of nodes, which reduces the security of blockchain to a certain extent.

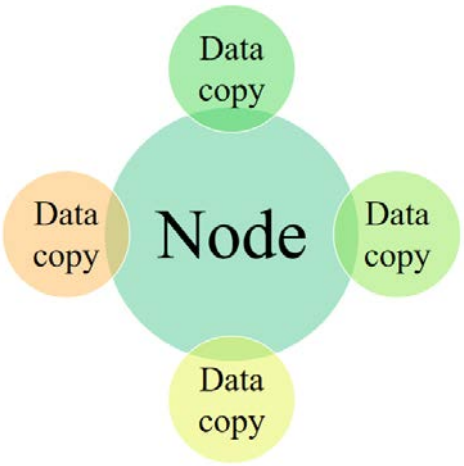

Fig.1 Extensible model storage framework

Data in the blockchain cannot be deleted, and transaction records of accounts can be traced back in the chain. Firstly, each block in the user node is encrypted by POR method, and the corresponding ciphertext and key are obtained. Then, the user node calculates the number of copies to be saved in each block. Then, the model saves the key generated by POR method to the local memory and sends it to the verification node for storage. And storing the encrypted block data in a storage node. The record of small file operation behavior includes file operation type, small file path, offset and length. Operation behavior records of small files in data files are stored in the form of logs, and one data file corresponds to one operation behavior log. Generally speaking, if the distributed modular hierarchical application design is adopted, the fault-tolerant ability can be greatly improved.

3.3 Master-Worker space reclamation

The generation and recycling task is initiated by the metadata node, and the data server judges whether each aggregation unit needs to recycle storage space fragments, and generates a fragment recycling task list in the metadata server, each task corresponds to an aggregation unit. Every message sent by the replica node to the client contains the current view number, so that the client can track the view number and further calculate the number of the current master node. For example, selectively not responding to requests from other nodes, selectively sending different messages to different nodes, or jointly destroying the distributed system with other nodes with Byzantine errors. After the data copy is saved, in order to ensure the user node to read the data, the model returns the address of the storage node to the user node and saves it in the P chain to ensure the security of the address data of the storage node.

Fig.2 Master-Worker space reclamation program framework

Master-Worker distributed task framework is adopted for generating and executing recycling tasks [12] , as shown in Figure 2. Master is located in the metadata server, which mainly completes task management; Worker is located in the data server, and each data node has a Worker process, which is the task performer.

Because blind signature is used for voting, the voting results between nodes are unknown, and the voting results cannot be tampered, which improves the randomness of the voting results. Under asynchronous model, the execution of programs may always be blocked because of waiting for messages to arrive, so the execution time of algorithms is very uncertain. Because Worker may fail to execute tasks, Master is designed as a double list structure of "task list to be executed-task list to be executed". Master responds to the Worker task acquisition request and takes out the task from the queue to be executed and puts it in the execution queue. The implementation of intelligent contract adopts distributed consistent algorithm, which ensures that the storage capacity of the storers is based on consensus and is publicly visible, and users will not establish storage contracts with storers with insufficient storage capacity.

In the POW-type consensus algorithm, the reward mechanism is based on the nature of mining, and the nodes are hashed, and the reward can be obtained if the blocks are successfully mined. When the level 0 storage area reaches the given upper limit, the system sorts and stores all sstables in the level 0 storage area and all sstables stored in the level 1 storage area in the background according to the key value. In this scheme, after receiving the storage certificate every time, the data blocks stored in the contract are not stored cumulatively, so that the contract will not expand because of too many queries on the chain.

4. Performance test

The experimental development environment is Intel Core i5-6500 3.20 GHz CPU and PC with 16 GB memory. Ubuntu16.04 system with 16 nodes, each with 1 GB of memory and 60 GB of hard disk size, was established by using VM ware Work-station 12.5.2. The network environment tested is Gigabit network, and the average size of each file in the test is 100 KB.

Test the stability of the system based on the scalable model of blockchain storage capacity. There are 4, 8, 12 and 16 nodes in the experiment, all of which are storage nodes, and 3 of which are user nodes and verification nodes at the same time. Because of the two-tier format, any instruction from the outside can be intercepted by the master node of the internal system, and because of the traceability and non-tampering of blockchain technology, the authenticity of transaction data is guaranteed. The system should have as high query performance as possible. Therefore, we think there are two ways to expand the query layer on the existing system: the external database method and the built-in index method based on the internal auxiliary index.

Contracts are written in Solidity language. Test parameters are shown in Table 1. To test, T_{trans} (storage contract deposit transfer time limit) and T_{cer} (storage certificate return time limit) are preset as 400 s.

Table 1 Test parameters

Parameter	Value	Unit
C-SER（Service contract）	1.03	KB
C-STO（Storage contract）	5.87	KB
T_{trans}	400	s
T_{cer}	400	s
PG_{store} (Deposit deposit)	0.6	ETH

The experimental data are 1420 intelligent contracts with open source code information downloaded from etherscan.io After classification, the proportion of various types of intelligent contracts is shown

in Figure 3.

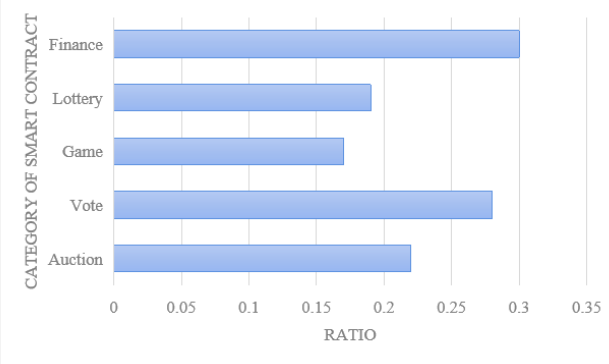

Fig.3 Category statistics

After classifying the smart contracts, sort the smart contracts according to the category order in Figure 2. The longest prefix algorithm based on suffix array is used to calculate the same substring between smart contract bytecodes. The average search time is 1.5s when the number of smart contracts is 10, and 3.8s when the number of smart contracts increases to 40.

Under the condition that all nodes are running normally and are not attacked, when data is fragmented, it is divided into 500 KB groups, and the minimum number of copies of each fragment is 2. When the experiment completed 146 transactions, 720 transactions and 1 077 transactions and generated broadcast data of 1 MB, 4 MB and 6 MB, the total storage capacity of all nodes in the blockchain system based on Hyperledge Fabric v0.6 is shown in Figure 4.

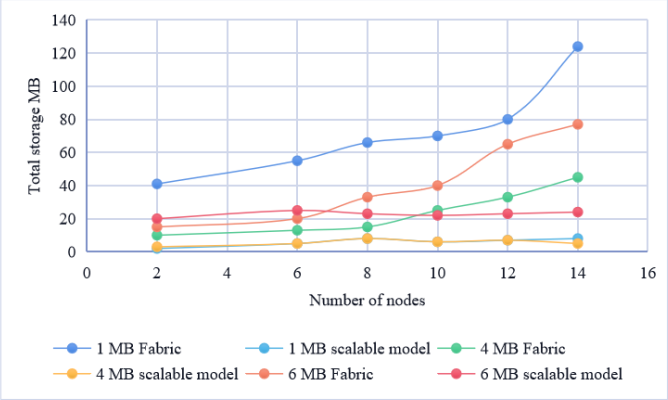

Fig.4 Scalable model and storage space occupied by Hyperledge Fabric

When the number of nodes is small, the total storage capacity of all nodes in the scalable storage model is similar to that of the Fabric blockchain. However, when the number of nodes increases, the storage space occupied by the scalable model is significantly reduced compared with the Fabric blockchain system. When the amount of stored data is small, the total storage capacity of all nodes in the scalable storage model is not much different from that in the Fabric blockchain.

Fig. 5 is a comparison of compression ratio before and after searching for the classification with 10 smart contracts. It can be seen from the figure that the bytecode of the earliest deployed smart contract will not be replaced. With the increase of smart contracts in the blockchain, the number of strings that can be used as pointers for bytecode replacement is also increasing, and the compression ratio of smart contracts is gradually increased and tends to be stable after a certain number.

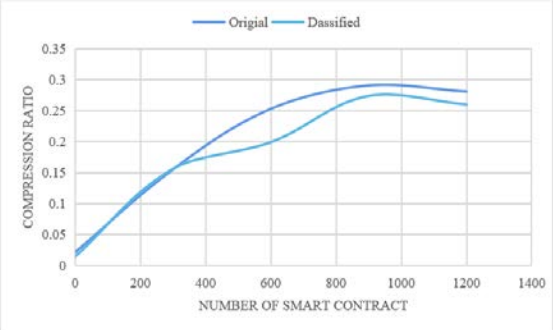

Fig.5 The compression ratio of adjacent contract number is 10

5. Conclusion

It is necessary to publicly review the consensus algorithm of blockchain, and most of the current consensus algorithms of blockchain are also promoted in this way. The consensus agreement of blockchain is improved step by step through the steps of expert proposal, public discussion, agreement realization and agreement supplement. In this paper, the periodic storage space recovery method of blockchain is proposed, which can save the storage space of intelligent contract by nearly 46% by reusing the bytecode of intelligent contract stored in ethereum blockchain. In the model, each block in the blockchain is stored in a certain proportion of nodes instead of all nodes. Moreover, node reliability verification is added to the model, which ensures data security. Future innovation lies in lowering the complexity of consensus algorithm. Consensus algorithms based on work permit will gradually withdraw from the market, while consensus algorithms that do not consume energy will further develop, which is a long-term development trend.

References

[1] Zhai Sheping, Duan Hongyu, Li Zhaozhao, et al. Blockchain Technology: Application and Problems. Journal of Xi 'an University of Posts and Telecommunications, 2018, 023no. 001, pp. 1-13.
[2] Yuan Yong, N I Xiao-chun, Shuai Zeng, et al. Development status and prospect of blockchain consensus algorithm. Zidonghua Xuebao/Acta Automatica Sinica, 2018,44 no. 11, pp. 2011-2022.
[3] Zhou Jie, Li Wenjing. Research on consensus algorithm of logistics blockchain based on cloud computing. Computer Engineering and Application, vol. 054, no. 019, pp. 237-242, 2018.
[4] Yuan Minfu, Elvis Lee, Chen Shengjian, et al. Blockchain networking scheme and data sharing storage mechanism based on cloud platform. Computer and Modernization, no. 9, pp. 46-52, 2019.
[5] Sean, Liu Baixiang, Zhang Ruyi, et al. Overview of blockchain technology. Computer Engineering, vol. 45, no. 05, pp. 7-18, 2019.
[6] None. Top Ten Events Worthy of Expectation in Shaping Blockchain Ecosystem. Science Grand View Garden, no. 2, pp. 28-29, 2020.
[7] Liu Hui. Research and Empirical Analysis on Storage Supply of Blockchain. Industrial Technology and Economy, vol. 039, no. 006, pp. 103-110, 2020.
[8] Li Ying, Yu Yaxin, Zhang Hongyu, Li Zhenguo. A highly trusted cloud storage model based on TBchain blockchain. Computer Science, vol.4 7, no. 09, pp. 336-344, 2020.
[9] Cao Shiming. Research on dynamic data storage security mechanism based on blockchain technology. Communication World, vol. 27, no. 05, pp. 110-111, 2020.
[10] Li Xiangming. Blockchain Storage: Data Sovereignty and Strategic Highland in the Era of Data Explosion. Zhangjiang Science and Technology Review, vol. 000, no. 006, pp. 32-35, 2019.
[11] Huang Gen, Zou Yibo, Xu Yun. Research on Merkle Tree Performance in Blockchain. Computer System Applications, vol. 29, no. 09, pp. 241-247, 2020.
[12] Zhao Yulong, Niu Baoning, Li Peng, et al. Blockchain Enhanced Lightweight Node Model. Computer Applications, vol. 040, no. 004, pp. 942-946, 2020.