Research on Power Data Storage Scheme and Settlement Scheme Based on Blockchain

Lihua Zhang, Jingjing Li and Fan Lan

June 22, 2019

Research on Power Data Storage Scheme and Settlement Scheme Based on Blockchain

Lihua Zhang[1], Jingjing Li[2]and Fan Lan[3]

[1] East China Jiaotong University , Nanchang 330000，China
[2] East China Jiaotong University , Nanchang 330000，China
[3] East China Jiaotong University , Nanchang 330000，China
lncs@springer.com

Abstract. In view of the privacy leakage problem existing in the centralized storage mode of power grid data, the blockchain technology can protect user privacy to a certain extent, but the blockchain has insufficient data storage capacity. In this paper, the blockchain technology is used to design the electricity data storage chain model, and the existing blockchain scalable capacity scheme is improved. Through comparison experiments, it can be shown that the scheme can minimize the total amount of storage under the premise of ensuring certain redundancy of data, and can achieve self-backup after being attacked and restore the original transaction verification performance. After that, this paper proposes a block-based power settlement model. This model uses the MPT tree + Electricity Consumption Information Storage Blockchain (ECISB) structure to settle the power consumption and verify the correctness. At the same time, the feasibility of the model is proved by experiments.

Keywords: Data Storage, Blockchain, Electricity Billing, Data Recovery

1 Introduction

1.1 A Subsection Sample

In recent years, cloud storage technology dominated by cloud computing technology has made data storage and application more convenient and efficient, but at the same time, security problems caused by personnel operations, software and hardware failures, attackers and other attacks are not to be underestimated. The emergence of the idea of blockchain distributed ledgers provides a new way of thinking about data privacy protection storage, which avoids the problem of data privacy leakage caused by centralized data management. In this paper, we will use the electricity data as an example to develop the research on blockchain storage [1]. In the smart grid, the alliance blockchain stores all the data in the Electricity Consumption Information Storage Blockchain (ECISB) after collecting the electricity data. As time goes by, the data on the chain gradually increases. However, the storage node responsible for maintaining the consensus verification node of the ECISB chain has a storage capacity limit. Therefore, solving the storage problem of the blockchain becomes crucial.

2 Related work

Centralized data storage is very detrimental to protecting user privacy. Therefore, many scholars began to use the decentralized idea of blockchain to store data for storage research. Qiao Rui et al [2] proposed a dynamic data storage mechanism based on blockchain technology. Through the requirements of each consensus node, the consensus mechanism used in the data storage process is improved, and the computational power is effectively utilized to avoid waste. Jia Dayu et al. [3] proposed an expandable blockchain storage model that can adjust the storage capacity according to the size of the data. In its scheme, the replica data of the blockchain is sliced and stored in multiple replica nodes. The stability of the node is verified by the POR method, and the node with higher stability is used to store the copy to ensure the secure storage of the data. Zhang Guochao et al. [4] used the method of secret sharing of thresholds to realize fragment storage for blockchain data. When nodes want to read transaction data, they request data from each node and recover data by Lagrange interpolation algorithm. Fei et al. [5] proposed a blockchain-based log storage system, which combines cloud storage and blockchain technology to achieve secure release and storage of log files. This paper will continue to study the blockchain storage problem and propose another solution to the storage problem of the blockchain.

3 System model

3.1 Model hypothesis

In the application of the uplink storage management application after the user's power consumption data collection[6], the design of the storage model needs to formulate corresponding assumptions:

(1) In order to reduce the redundancy of data storage in each node, it is necessary to reduce the number of full nodes in the original model, that is, to retain part of the whole node, and to set the remaining nodes as heavy nodes and light nodes;

(2) In order to ensure the availability and reliability of each storage node in the ECISB chain to meet the requirements of data storage and query on the chain, a special verification node should be set up for storage node data integrity and presence verification[7].

3.2 Model Structure Description

This paper will make some improvements to the capacity scalable blockchain storage model proposed by Jia Dayu et al. There are three types of nodes in the model: user node, verification node and storage node. Each node can act as more than one node role. Each storage node is distributed in various locations in the network. The stored copies are not the same size. They can store the entire blockchain and store one part of it.

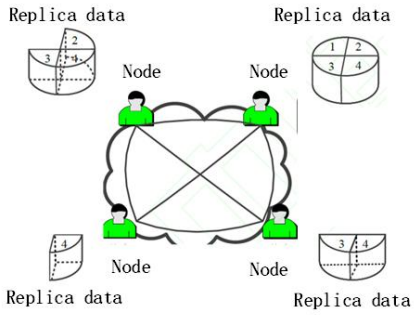

Fig.1. scalable storage model framework

The model used in the ECISB chain storage application (shown in Figure 2) of this paper makes some improvements to be applied to the distributed storage of the entire power account blockchain data after the smart grid power data is uplink. The details are as follows:

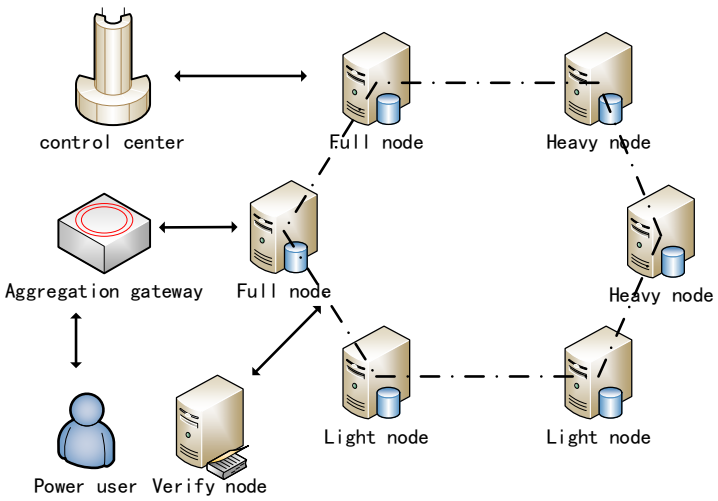

Fig.2. electricity storage model data

(1) The user node (smart meter) is only responsible for encrypting its own power consumption data through the Paillier encryption system, and uploading it to the aggregation gateway to charge the power consumption data of all users in the area under the jurisdiction;

(2) The ECISB chain maintenance node participates in the consensus process of the block uplink and is responsible for storing the copy in the different split mode of the ECISB chain book data, that is, acting as a storage node (full node/heavy node/light node).

(3) The verification node[8] is used for verifying the reliability of each storage node, and the verification node can also be completed by the storage node, and the

verification node group[9] is responsible for maintaining a reliability evaluation chain.

(4) Storage nodes are divided into three categories: full node, heavy node, and light node due to different storage capacity, computing performance, network state, security strength, etc, which are used for storage tasks of different blockchain ledger data.

(5) The blockchain storage capacity scalable model designed in this paper, in addition to the main chain for storing data, also designed two auxiliary chains, which are respectively P (Position) storage location chain and node reliability evaluation chain R chain. Used to record the storage location of the data copy and the evaluation value after verifying the node storage reliability, making full use of the non-tamperable characteristics of the blockchain to ensure secure data storage for persistence.

4 The specific programs

4.1 Power data storage

After the power data collected from the user is encrypted[10] by the aggregator, a block is constructed by the aggregation node to load the aggregated power data into the transaction verification tree (Merkle tree) of the block. The power consumption block structure is shown in Figure 3. The hash value of the previous block records the hash value of the previous block, and the block hash in the block header is the hash value assigned to the block when the block is generated[11]. The timestamp field is used to record the time the block was generated. The power usage total field is used to record the aggregated encrypted value of the plurality of pieces of power usage information collected in the current block. The Merkle root field stores the hash value of the root node of the Merkle tree generated by the aggregation node after the collected power consumption is calculated by Hash.

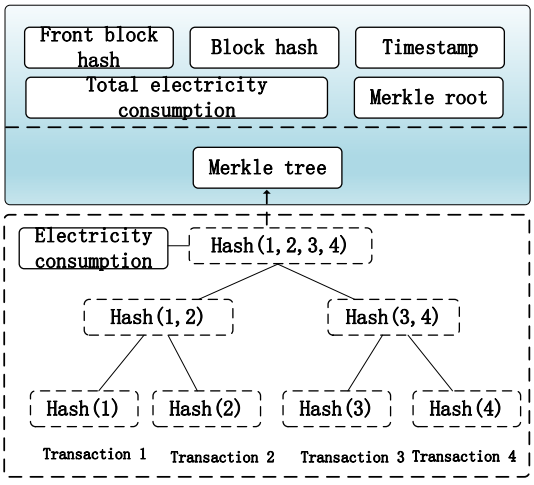

Fig.3. power block structure

Because the data received by the aggregation gateway node may exceed the amount of data that can be accommodated in the block body each time the power consumption data is collected[12]. Therefore, it is necessary to generate a plurality of blocks for the data, and apply for the uplink to the consensus maintenance node of the adjacent ECISB chain in turn, and then add them to the original ECISB chain according to the timestamp order after the consensus verification.

The storage process of the ECISB chain is shown in Figure 4 and is divided into three main steps:

Step 1: After the smart meter installed on the user side, the user periodically uploads the power consumption information to the nearby aggregation gateway node according to the demand of the smart grid power data collection, performs aggregation and encryption, and generates a data block;

Step 2: The aggregation gateway node queries the verification node for the storage node with higher reliability in the current network, and the verification node returns the information of the candidate storage node.

Step 3: According to the candidate node returned by the verification node, the aggregation gateway node sends the block data replication multiple copies to different storage nodes for backup storage, and records the storage location in the P chain.

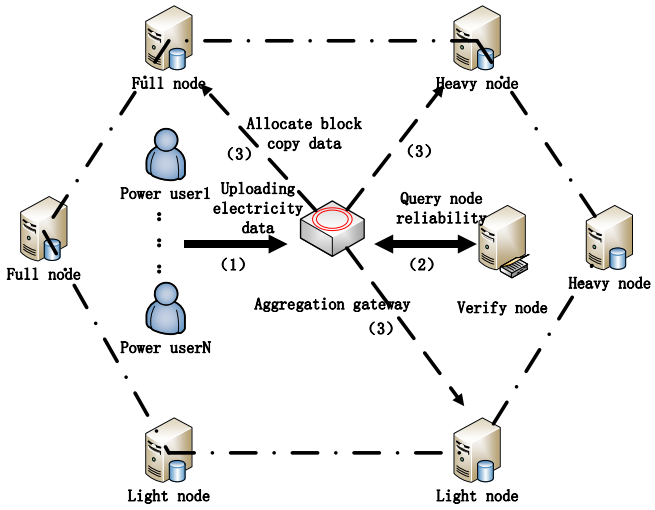

Fig.4. block copy stored procedure

4.2 Reading with electricity data

When the power data set stored in the storage node is obtained, the corresponding storage node may be found at a location stored in the P chain maintained by the aggregation gateway node, and a data request is sent to the storage node, and finally, the required data is decrypted, and the process is shown in Figure 5.

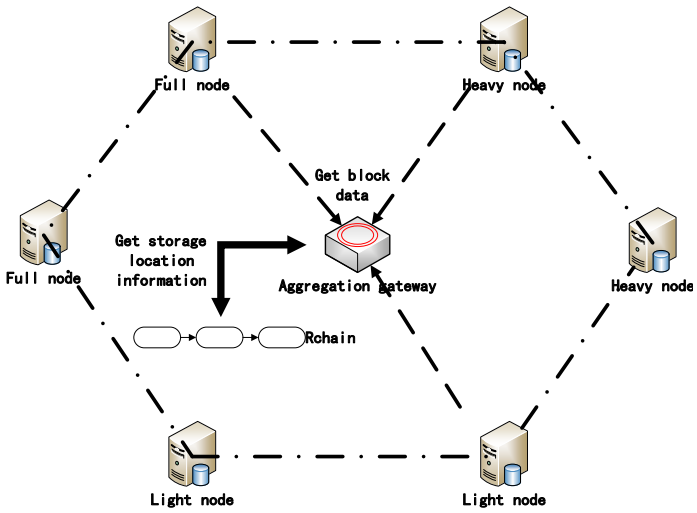

Fig.5. data reading process

The aggregation node contains the following main steps when getting data:

Step 1: Query the allocation location of the required data copy in the P chain;

Step 2: Initiate a complete backup request of the required data to the adjacent storage node, and request the Merkle root of the block where the data is located to the remaining copy storage nodes;

Step 3: The storage node returns the required data and the corresponding Merkle root information, and compares the Merkle root generated by the data in the complete backup of the data block with the block Merkle root returned from other storage nodes to verify the integrity of the data ;

Step 4: Decrypt the block data using the POR protocol to recover it, and then obtain the original data block.

4.3 Data Storage Reliability Verification and Recovery

In order to ensure the storage reliability of each storage node, it is necessary to perform integrity and presence verification on the data in each storage node from time to time, and the work is completed by the verification node. The storage node reliability verification process is shown in Figure 6. The main steps include:

Step 1: Verify node obtains the storage location of the corresponding data block from the P chain;

Step 2: Applying data to a plurality of nodes storing the same block data copy for verification, and performing integrity verification and presence verification on the data returned by the plurality of storage nodes to obtain reliability evaluation thereof;

Step 3: Store the newly calculated storage node reliability evaluation in the R chain and update it.

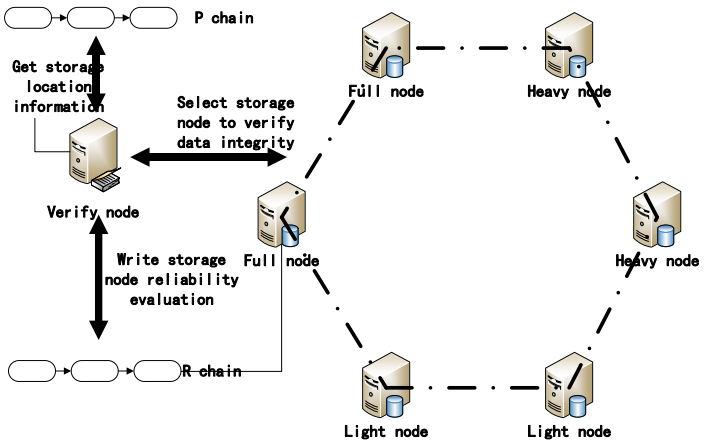

Fig.6. node storage reliability verification process

The reliability value of the node is measured by the computing performance, the network performance, the credibility, and the number of times the node is selected. whose calculation formula is:

$$RV = \frac{CP * NP * CRE}{\sqrt{T}} \tag{1}$$

The node credibility CRE value is set to the same value at the beginning. Once the node data is lost or the data is falsified, the verification node reduces its value. The higher the number of times a node is selected, the better its reliability is than other nodes. However, as the number of times of storage increases, its capacity is continuously consumed. In order to balance the storage space of the storage node, it is necessary to appropriately reduce each node to be used all the time, so the RV value is inversely proportional to T.

In this paper, a Dynamic Data Recovery Scheme (DRS) is proposed for data loading and recovery when new nodes join or node data is behind. The storage node data recovery process is shown in Figure 7.

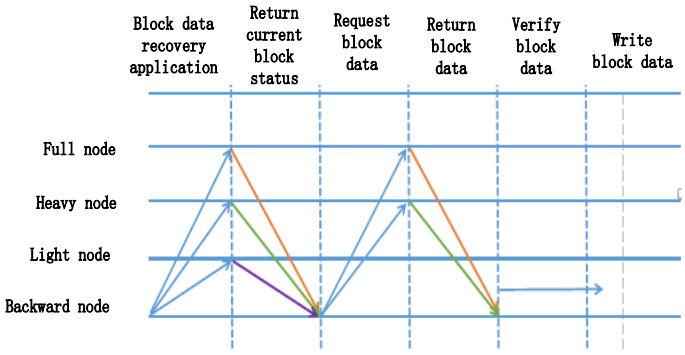

Fig.7. storage node dynamic data recovery process

Step 1: The backward node applies to the node with the latest backup in the network to synchronize the block copy data of the node, and prepares to start the recovery update procedure;

Step 2: Each accessed storage node is responsible for returning the status of the block copy currently owned by itself;

Step 3: Select the required copy data by the backward node, and apply for block data synchronization to not less than f+1 nodes having the same copy data;

Step 4: The storage node applying for the block replica data returns the data required by the backward node;

Step 5: The backward node first verifies whether the block data sent by each node is consistent. If it is consistent, it stores it in its own storage medium. If it is inconsistent, it obeys the majority, even if the data in most nodes is wrong. That is, the backward node obtains the wrong data and is also eliminated when the verification node performs data integrity check.

Step 6: Verify node adds reliability evaluation information to it in the R chain.

4.4 Experimental analysis

This section will test the improved block storage model applied to the storage process of the post-block data of the smart grid. In the laboratory, due to the existing 5 servers, the experimental conditions are limited. Therefore, VMware virtual machine software is installed on each server. Each server has 8 virtual servers equipped with Cent OS 6.7 operating system, and 40 virtual server nodes are added to further perform storage performance testing.

In the experiment, a total of 10 virtual machines installed in the No. 3 server and the No. 4 server are used to form a block data storage cluster, wherein 2 full nodes, 4 heavy nodes, and 4 light nodes. In order to test the reliability of data storage, the network service of one virtual full-node server is artificially closed, and the remaining nodes of the whole network can be observed whether the transaction verification, data storage and recovery work can be completed normally.

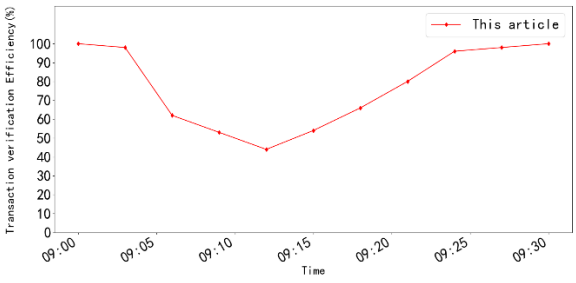

Fig.8. transaction verification efficiency changes

Figure 8 is an image of the transaction verification efficiency change of the remaining nodes after the network service of one full node is cut off. After cutting off the entire node at 09:03, the transaction verification efficiency immediately dropped to about 60%. Since the remaining nodes do not have all the block data, only one full node

provides the transaction verification service, and the rest of the nodes will slow down the storage response to the block. In the node network, it is set to check the survival state of the whole node once for 2 minutes, so that the node is found to have lost all nodes after the time of 09:02. Therefore, the heavy node as the backup full node starts to request all the data to be backed up to the only one full node, and replaces the whole node. After about 10 minutes, the data backup is completed, and two full nodes appear in the whole network. The transaction verification efficiency starts to recover. Until 09:30, the original transaction verification performance is basically restored.

4.5 Summary

In this paper, three types of nodes are set, which are full node, heavy node, and light node. The full node needs to allocate data to the storage node each time it is uploaded. The light node only stores part of the fragmentation data, as well as the block header information of all blocks on the entire chain, for subsequent transaction verification. The heavy node is between the full node and the light node, and the partial copy can be deleted as a light node, and the entire ECISB chain book block data can be updated to become a full node, that is, used as a candidate full node.

5 Subsequent transaction verification expansion plan

In the settlement process of electricity consumption, all the electricity consumption data generated during the settlement period shall be collected and aggregated, thereby generating a corresponding electricity tariff settlement report. In addition, it is necessary to verify the correctness of each historical electricity data transaction content.

In the traditional blockchain system, such as the bitcoin and Ethereum blockchains, the data is retrieved by traversing all the blocks to find the target data, which makes the retrieval efficiency relatively poor. At the same time, the data correctness verification also causes the performance of the consensus process to be weakened.

5.1 Search and settlement of electricity consumption

This paper combines the advantages of Ethereum's account-based MPT model and improves the proposed schemes in [13]. Applying it to the settlement process of power consumption data, the model of "MPT structure + Electricity Consumption Information Storage Blockchain (ECISB)" is used to realize the power consumption settlement and the quick query and correctness check in the retrieval process.

MPT-ECISB Tree Structure Model.

In the MPT-ECISB tree, the MPT directory index tree + ECISB power data storage chain is used, and the account ID stored in the MPT is associated with the power consumption data recorded in the ECISB. In the MPT, the advantages of the Patricia tree and the Merkle tree are combined. One path represents an account address and has the advantage of quick query and verification data. The MPT-ECISB tree includes a root node, a branch node, an extended node, and a leaf node. The structure of each node is as shown in Figure 9.

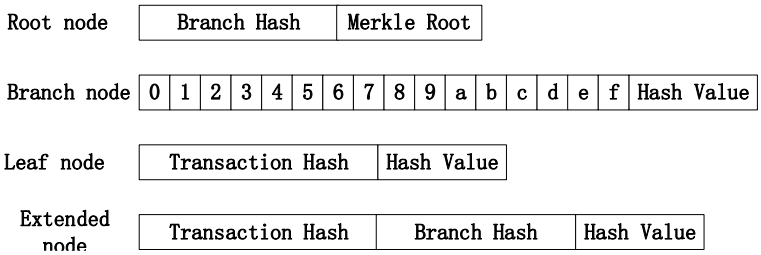

Fig.9. structure diagram of each node

(1) Root node. Contains a branch hash index field Branch Hash, which can be used to find its child branch nodes. The Merkle root hash value is stored in the field Hash Value. When a new transaction is generated, this field value needs to be updated to ensure the integrity of all data.

(2) Branch nodes. The address identifier of the account is implemented by hexadecimal coding, so 16 branches in the branch node represent 0-f, which is used for the composition of the account address, and stores the hash index of the next node therein.

(3) Leaf nodes. The Transaction Hash field in the leaf node points to the latest transaction address owned by the current path. When a new transaction is generated under the leaf node corresponding account, the field is modified to become the Hash index of the latest transaction, and the previous transaction is recorded in the updated transaction through the Parent Hash field carried by the transaction data itself. At the same time, the hash value of the node is stored in the Hash Value field, and the leaf node is the lowest node of the MPT-ECISB tree.

(4) Expansion node. When there is an account address path with the same prefix, since the subsequent account codes may not be exactly the same, it is necessary to perform branching on the branch nodes at the same prefix, and use the extended nodes to open up a new account path.

The power usage data is written to the block in the form of a transaction. Therefore, each time a block is generated, the power consumption transaction contained in the block needs to be updated to the MPT-ECISB tree.

Power Data Query Based on MPT-ECISB

The MPT-ECISB tree can construct a power data as an index directory for transactions based on the accounts, but at this time, the account can only be associated with one of the most recently uploaded power data. In order to obtain the power consumption information in a certain power cycle, it is necessary to obtain the previously generated power consumption data set. Therefore, the "tree + chain" combination method is used to extract the power consumption data set corresponding to the account. Figure 10 shows the complete structure of the MPT-ECISB when performing a query operation.

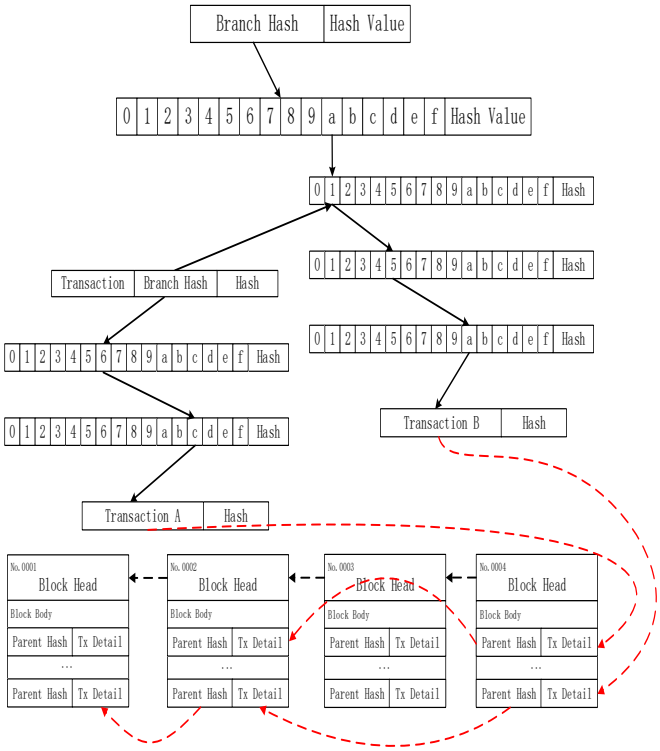

Fig.10. MPT-ECISB power data query process

In Figure 10, the last power consumption data report record of the account is recorded in the leaf node, and the records are stored as transactions in the block of the ECISB chain, and the Parent Hash field in the block is used to store the last power transaction. Where it is. Each time a new power transaction is generated, the address of the previous transaction is written in the Parent Hash field of the transaction. By analogy, a power consumption data record chain is formed, and all data is written in chronological order. Therefore, when the query is needed, only the leaf node corresponding to the current account needs to be found, and the historical historical power consumption data of the entire account can be quickly extracted. It can save a lot of time compared to the way in the existing blockchain system to traverse all the blocks to find the target transaction.

5.2 Search results correctness verification

The Merkle tree feature makes the MPT-ECISB tree configuration impossible to be modified once it is completed. The verification nodes in the ECISB update their own MPT-ECISB trees according to the broadcast transaction information, and then broadcast the updated Merkle root value. The comparison results from the other verification nodes are compared. If they are consistent, the update process of the MPT-ECISB tree is completed. Otherwise, the current transaction data is discarded, and the next round of consensus update is awaited.

In ECISB chain model of this paper, using the Full node, node heavy and light maintenance of the common node, while verifying the transaction data also need to consider performance issues nodes. Therefore, a simplified verification mechanism （SVM） suitable for the electric data query model in this paper is proposed. Take the light node with weak computing power as an example to illustrate the operation mode of the SVM in this paper.

The light node cannot perform the verification of the transaction data, and needs to apply to other nodes to obtain the Hash path segment of the Merkle tree branch portion of the transaction in the block, and start to restore the Merkle tree upward after receiving the Hash segment from the multiple nodes. The Merkle tree in the middle is a binary tree, and its complexity can be as low as $\log_2 N$ in the Merkle tree restoration verification operation of N transactions.

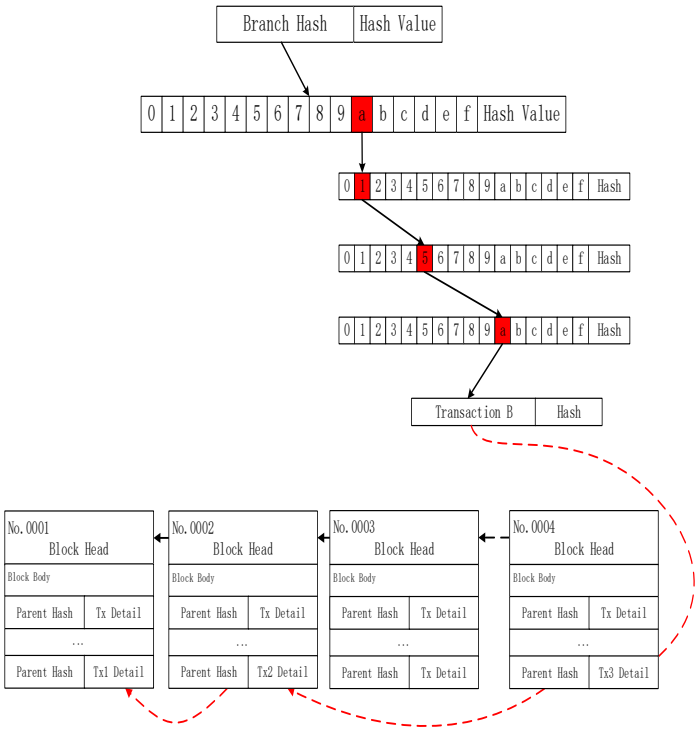

Fig.11. uses the electrical data query verification process

As shown in Figure11, the account 0xa15a completes the data query process of a certain power interval on the light node. The leaf node is found in the MPT-ECISB tree by the user's account code, and its corresponding latest transaction Tx3, continuing to find the transaction Tx2 and the transaction Tx1.

The three power transactions are on the blocks numbered 0004, 0002, and 0001. At this time, the light node applies to the storage node for the hash of the Merkle tree branch path of the block containing the three transactions. After getting the Hash path fragments from multiple nodes, the light nodes themselves complete the construction

of the Merkle tree. If the Merkle root in the block header saved by itself is in one-to-one correspondence, the query is verified, and the data can be used for settlement processing to support the application service in the upper layer.

5.3 Experimental evaluation

In the experiments in this section, the main purpose is to test the query efficiency of the electricity transaction data in the ECISB chain block after the MPT-ECISB model is adopted. This experiment uses 5 servers, each of which runs 5 nodes, a total of 25 nodes, including 5 full nodes, 10 heavy nodes and 10 light nodes, which constitute the maintenance network of the ECISB chain. During the experiment, 10,000 electricity accounts were constructed, and 2000 electricity consumption data were associated with each electricity account, and a total of 20 million electricity consumption data were used to test the query accuracy and efficiency performance of the MPT-ECISB model.

In order to illustrate the query efficiency of the MPT-ECISB model, the performance should be measured from two perspectives: accurate query time consumption and interval query time consumption. As shown in Figure 12, the time spent in accurately querying the account transaction, the two lines are 50% and 100% of the test power consumption data stored in the ECISB chain. As can be seen from the test results in Figure 12, when performing an accurate query, whether it is on the 50% or 100% data chain does not put pressure on the query.

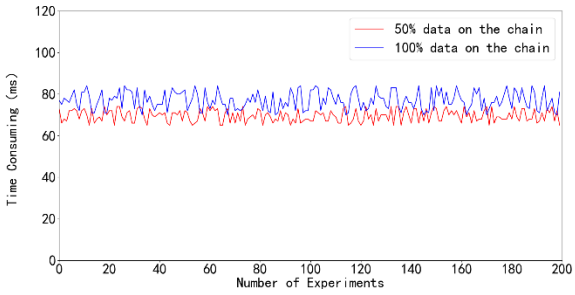

Fig.12. accurate query time-consuming test

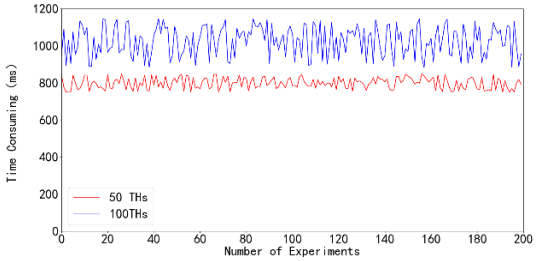

Fig.13. interval query time-consuming test

Figure 13 shows the time spent in querying an account history transaction by interval. It can be seen from the figure that when the magnitude of the query increases from 50 to 100, the query delay increases by 800ms (0.8s) to about 1000ms (1s), and the network delay and verification calculation take a certain amount of time. As a result, the total time taken is greatly increased. However, for query applications in the settlement of power transactions, this delay is acceptable.

6 Summary

This paper improves the existing blockchain scalable capacity model and designs a replica storage model for the block data on the ECISB chain, which solves the storage problem caused by insufficient storage capacity of the blockchain data[14]. At the same time, according to the demand in the actual power consumption settlement, a model that can support the quick query of historical data based on the account is proposed. This model combines the MPT tree in Ethereum, can quickly retrieve data and verify data consistency, and proposes the structure of account-based MPT tree + ECISB chain. It is possible to efficiently query the electricity transaction data[15] distributed in each block without traversing all the blocks, so as to facilitate quick query and settlement of electricity, and at the same time ensure the correctness of the data.

References

1. Xue Tengfei, Fu Qunchao, Wang Wei, et al. Research on medical data sharing model based on blockchain[J]. Acta Automatica Sinica, 2017(09):73-80.
2. Qiao Rui, Dong Shi, Wei Qiang, Wang Qingxian. Research on Dynamic Data Storage Security Mechanism Based on Blockchain Technology[J]. Computer Science, 2018, 45(2): 57-62.
3. Jia Dayu, Xin Junchang, Wang Zhiqiong, Guo Wei, Wang Guoren. Scalable Model of Storage Capacity of Blockchain[J]. Computer Science and Exploration, 2018, 12(04): 525-535.
4. Zhang Guochao, Wang Ruijin. Blockchain fragmentation storage model based on threshold secret sharing [J/OL]. Computer application: 1-9[2019-05-29].http://kns.cnki.net/kcms/detail/51.1307.TP.20190515.1012.002.html.
5. Fei Wei, Jing Jing, Hu Qing. Log storage system based on blockchain [J]. Cyberspace Security, 2018, 9 (06): 80-85.
6. Zyskind G, Nathan O, Alex. Decentralizing Privacy: Using Blockchain to Protect Personal Data[C]// IEEE Security and Privacy Workshops. IEEE Computer Society, 2015:180-18.
7. Ouaddah A, Abou Elkalam A, Ait Ouahman A. FairAccess: a new Blockchainbased access control framework for the Internet of Things[J]. Security & Communication Networks, 2017, 9.
8. A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram. (Dec. 2017). ‘‘LSB: A lightweight scalable blockchain for IoT security and privacy.’’ [Online].Available: https://arxiv.org/abs/1712.02969?context=cs
9. X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, ‘‘ProvChain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and

availability,'' in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, Madrid, Spain, May 2017, pp. 468–477.

10. S. Ma, Y. Deng, D. He, J. Zhang, and X. Xie, ''An efficient NIZK scheme for privacy-preserving transactions over account-model blockchain,'' IACR Cryptol. ePrint Arch., Tech. Rep., 2017, p. 1239.
11. Marco Lakhani Karim R Iansiti. The Truth About Blockchain. Harvard University, Harvard Business Review, January 2017.
12. Marko Vukolic. Rethinking permissioned blockchains. In ´*Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pages 3–7. ACM, 2017.
13. Xu Y , Zhao S , Kong L , et al. ECBC: A High Performance Educational Certificate Blockchain with Efficient Query[J]. Theoretical Aspects of Computing, 2017:288- 304.
14. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[EB/OL][2008-08]. https://bitcoin.org/en/bitcoin-paper, 2008.
15. Hardwick F S, Gioulis A, Akram R N, et al. E-Voting with Blockchain: An E-Voting Protocol with Decentralisation and Voter Privacy[J]. 2018.