Computer Systems Science & Engineering
DOI:10.32604/csse.2021.014530
Article

Tech Science Press

A Storage Optimization Scheme for Blockchain Transaction Databases

Jingyu Zhang[1,2], Siqi Zhong[1], Jin Wang[1,3], Xiaofeng Yu[4,*] and Osama Alfarraj[5]

[1]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, 410114, China
[2]School of Systems Engineering, National University of Defense Technology, Changsha, 410000, China
[3]School of Information Science and Engineering, Fujian University of Technology, Fuzhou, 350000, China
[4]School of Business, Nanjing University, Nanjing, 210093, China
[5]Computer Science Department, Community College, King Saud University, Riyadh, 11437, Saudi Arabia
[*]Corresponding Author: Xiaofeng Yu. Email: xiaofengyu@nju.edu.cn
Received: 26 September 2020; Accepted: 10 November 2020

Abstract: As the typical peer-to-peer distributed networks, blockchain systems require each node to copy a complete transaction database, so as to ensure new transactions can by verified independently. In a blockchain system (e.g., bitcoin system), the node does not rely on any central organization, and every node keeps an entire copy of the transaction database. However, this feature determines that the size of blockchain transaction database is growing rapidly. Therefore, with the continuous system operations, the node memory also needs to be expanded to support the system running. Especially in the big data era, the increasing network traffic will lead to faster transaction growth rate. This paper analyzes blockchain transaction databases and proposes a storage optimization scheme. The proposed scheme divides blockchain transaction database into cold zone and hot zone using expiration recognition method based on Least Recently Used (LRU) algorithm. It can achieve storage optimization by moving unspent transaction outputs outside the in-memory transaction databases. We present the theoretical analysis on the optimization method to validate the effectiveness. Extensive experiments show our proposed method outperforms the current mechanism for the blockchain transaction databases.

Keywords: Blockchain; distributed systems; transaction databases

1 Introduction

Blockchain technology has been widely applied and studied with its decentralized characteristics and the shared transaction databases. At present, many scenarios have been applied with the blockchain technologies, including the cryptocurrency systems, electronic ledger systems. The typical blockchain systems are based on different underlying computer theories or technologies, covering the blockchain keys, transactions, Unspent Transaction Output (UTXO), etc.

A pair of blockchain keys normally consist of a private key and a public key. The public key is like the bank account number, which is used to receive digital property. The private key is like the password of the account. The blockchain key encryption is based on the cryptography principle of specific mathematical

functions (e.g., elliptic curve multiplication), which makes the generated digital key irreversible and unchangeable. In a transaction, the blockchain address usually appears as the payee account number. If user A wants to pay a digital coin to user B, A should get the address of user B at first. Fig. 1 describes the relationship between private key, public key, and the generated address in the typical bitcoin-like blockchain system. The private key is a randomly generated number. The system uses elliptic curve multiplication to generate a corresponding public key, and hash functions are used to generate the blockchain addresses.

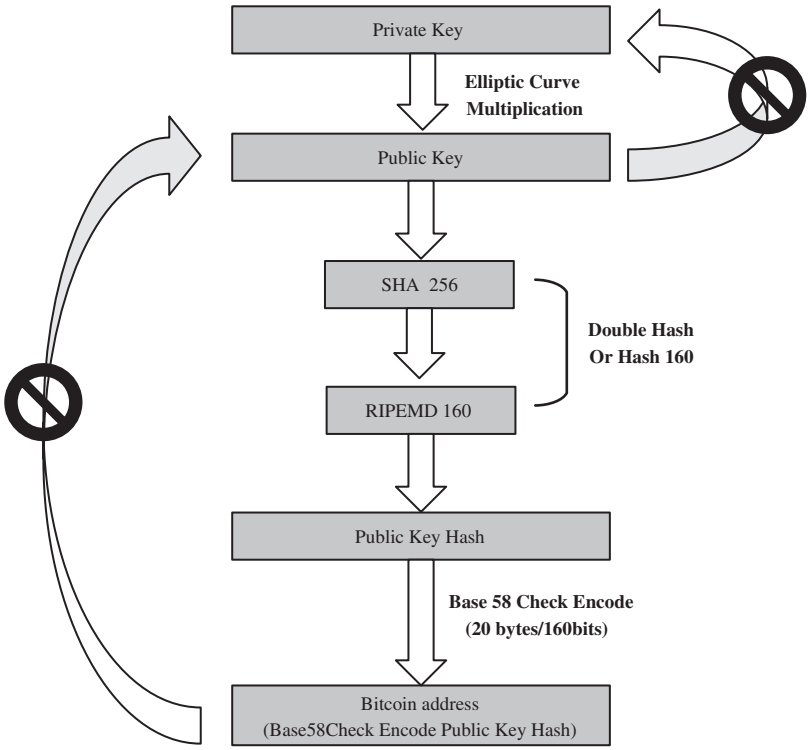

Figure 1: Keys in blockchain systems

Based on cryptography proof, the digital keys and blockchain addresses provide the blockchain system high security. Meanwhile, it also brings an issue to users: the private key must always be kept confidentially, once the private key is lost, it is extremely difficult to recover. Indeed, many private keys have been lost in the bitcoin system. Transaction database is another important part of bitcoin-like blockchain systems. A blockchain transaction consists of input and output. As shown in Fig. 2, the transaction input refers to the UTXOs in the previous transaction. The transaction output represents the ownership transfer status of the unspent digital asset. Blockchain transaction records all digital asset transfers, which forms a traceable transaction chain. The basic unit of blockchain transaction is the UTXO. In bitcoin-like blockchain systems, there is no account balance, and only the UTXOs are distributed in different blocks. The so-called account balance is actually the sum of all UTXOs belonging to the user address.

UTXO is indivisible and can only be consumed as a whole. This feature determines that the total input UTXO value must be equal to the output UTXO value. In most cases, as shown in Fig. 2, the number of input UTXOs is one or more, and at output end, there will be at least three UTXOs: One UTXO paid to the payee, one paid to the miner, and one UTXO change returning to the payer. In order to independently verify transactions, nodes need to track all the UTXOs in the blockchain databases.

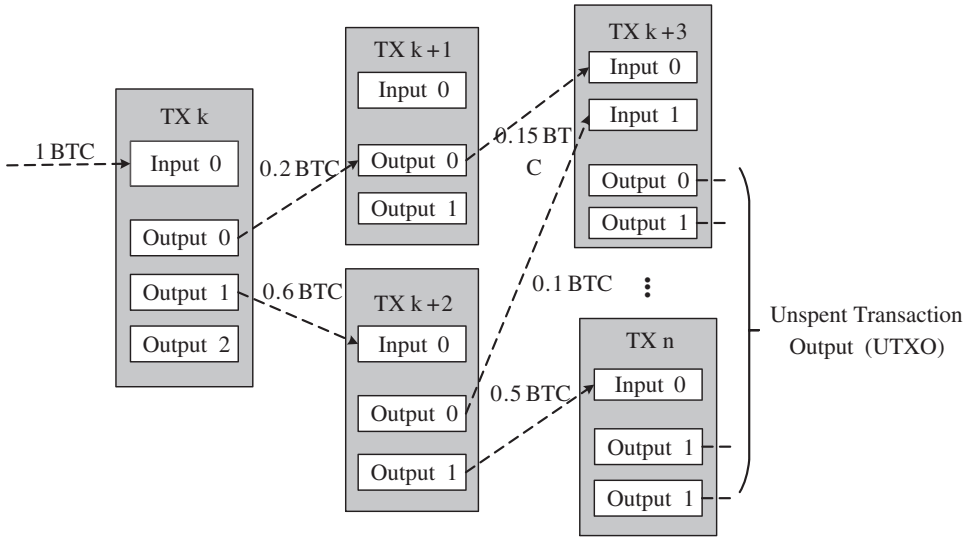

Figure 2: Blockchain transaction chain

Blockchain nodes store all retrieved UTXOs in the in-memory database, and all UTXOs form an UTXO set. Every transaction can generate more UTXOs. Therefore, as the transactions happen, the UTXO set will become larger and larger, and more memory space should be occupied. Therefore, the continuous database growth will require more memory capacity. This paper aims to study the memory storage of blockchain transaction databases, so as to propose an effective storage optimization method.

The organization of this paper is as follows: Section 1 introduces the research background and existing problems of blockchain transaction database storage; Section 2 introduces the related knowledge; Section 3 describes the proposed method and theoretical analysis; Section 4 shows the experimental results and analysis; and last section summarizes the work.

2 Related Work

In this section, we review the related work for the blockchain systems, and the performance optimization for storage and computer systems in various scenarios [1–9].

By using encryption algorithm, time stamp technology, distributed consensus and economic incentives, blockchain technology effectively solves the consensus problem among connected nodes, and realizes decentralized peer-to-peer transaction in a distributed system without mutual trust between nodes. Bitcoin system is the most typical blockchain system, and the structure of bitcoin blockchain system is shown in Fig. 3. In the blockchain system, the miner node uses the block header hash value as the link pointers to link the data blocks together to form a unchangeable chain. Each block includes a block header and a block body. The block header encapsulates the metadata information of the current block, which provides the possibility of tracing historical transaction information. The block body mainly contains the transaction tree status information supported by hash algorithm. Each transaction is permanently recorded in the data block, and anyone can query it. The Merkle tree in the block body will sign each transaction digitally, which can ensure that every transaction is unforgeable and prevent double spend attack.

The blockchain data appending mechanism determines that transaction database size will grow continuously. The blockchain storage mode [10] solves the problem of decentralized trust, but the premise is that nodes must maintain a complete blockchain database to independently verify transactions [11]. This node holding a complete block chain database replica is called “full nodes” [12]. In order to operate the full

node, users need to prepare enough storage space and computing power [13]. Therefore, it is difficult for resource limited devices to become full nodes [14]. For example, mobile devices with limited storage space are not suitable to become full nodes. Imtiaz et al. [15] studied the influence of orphan transaction on the performance of blockchain network. The continuous growth of blockchain database will also reduce the speed of transaction verification by nodes, which will hinder the development of the system.

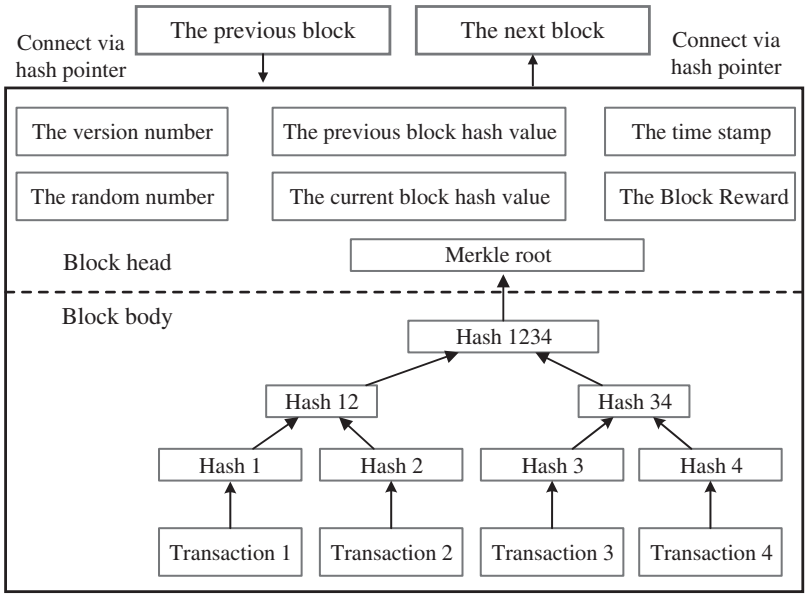

Figure 3: Blockchain structure

To enable nodes with limited performance to operate without saving a complete blockchain database, Nakamoto proposed a Simplified Payment Verification (SPV) method in bitcoin white paper. This type of node is also known as a lightweight node. SPV node has gradually become the most common node form in blockchain system [16–18]. And bitcoin wallet has been successfully implemented in many blockchain applications. SPV nodes can not conduct independent transaction verification because they do not download the complete blockchain database, so they need to connect several nodes randomly. This random connection means that they may be attacked. To enhance the node security in blockchain systems, developers introduced bloom filter [19] to deal with the privacy issues of SPV nodes. A clearer bloom filter will produce more accurate results, but at the cost of exposing the address used in the user's wallet.

The Interplanetary File System (IPFS) is a peer-to-peer version control file system, which not only ensures the security of the storage platform but also solves the single node failure problem [20]. IPFS is used to improve the blockchain system widely, and it combines distributed hash table, incentive block exchange and self certified namespace [21]. Zheng et al. [22] proposed a blockchain data storage model based on IPFS. The miner stores transaction data into IPFS network and packages the returned IPFS hash value into blocks. The scheme greatly reduces the blockchain data by using the characteristics of IPFS network and IPFS hash. According to the characteristics of IPFS, Chen et al. [23] proposed an improved P2P file system scheme based on IPFS and blockchain. At present, the combination of IPFS and blockchain has also been applied by many researchers in the Internet of Things [24,25], and other industry applications [26] for privacy protection.

Section-Blockchain [27] is a blockchain protocol that reduces storage efficiently, the purpose of the protocol is to solve the problem of super large capacity storage without affecting the blockchain. Section-

blockchain runs on an efficient communication protocol, which helps nodes optimize their position in the network, and realizes formatted network layout and faster data broadcast speed. Distributed Hash Table (DHT) [28] is another design to implement hash table among peer-to-peer network nodes. Chord [29] is a new DHT design, which uses a ring overlay network. Abe et al. [30] proposed a distributed storage load balancing scheme based on distributed hash table via Chord. It can effectively reduce the storage space of nodes.

Blockchain technology provides a new cheap, safe and decentralized electricity trading mode for the power sector. However, there is a delay in the electricity trading system based on blockchain. Therefore, Okoye et al. [31] proposed a novel network enhanced transactional micro grid model based on blockchain technology. The optimized blockchain participant permission model improves the transaction speed and convenience. In addition, based on the characteristics of blockchain transaction system, Bai et al. [32] and Shi et al. [33] proposed optimization algorithm and block transaction selection mechanism to improve the stability of blockchain transaction system respectively.

Guo et al. [34] proposed an optimization scheme based on redundancy system, which greatly reduces the storage capacity of blockchain system nodes, and designs a fault-tolerant mechanism based on the scheme. Mbinkeu et al. [35] studied the memory management and access time of bitcoin protocol based on SQLite databases. Mei et al. [36], a memory optimization mechanism based on redundancy system is proposed to reduce the storage capacity of each node. Wang et al. [37] dealt with how to allocate data to each computer in the blockchain network. In this work, authors presented a balanced solution of user input on search time and space occupation. El-Hindi et al. [38] introduced an additional database layer into the blockchain system to improve the performance and scalability of data [39] sharing. In view of bitcoin key management problems, Gennaro et al. [40] proposed a centralized threshold signature scheme for bitcoin systems with higher efficiency.

The most prominent feature of blockchain system is decentralization. In order to realize decentralized systems, there must be enough nodes to independently verify transactions. Most of the existing solutions, such as the most widely used lightweight node scheme, can alleviate the storage problem. As we mentioned before, a large number of resource limited devices can participate in the transaction verification. Lightweight nodes rely on the database storage of full nodes. It is still a problem to be solved to improve the independent verification and storage capacity of blockchain nodes.

In terms of current blockchain node storage schemes, there is no solution for temporarily useless UTXO in the blockchain transaction databases. In order to improve the in-memory UTXO set and storage capacity, this paper proposes an optimized storage scheme for blockchain transaction databases. The main contributions of this paper are as follows:

1. This work investigates the blockchain transaction databases and presents the mathematical models for theoretical analysis.
2. A storage optimization scheme is presented to remove the expired UTXO from the memory. It increases the memory storage space, and improves in-memory UTXO set.

3 Theoretical Analysis

In this section, we theoretically analyze the expansion speed of blockchain transaction databases. Then, we propose the storage optimization scheme, and analyze the proposed method mathematically. Tab. 1 lists the related representations of symbols used frequently.

Table 1: Definitions of related symbols

Symbol	Description
Ω_{in}	Total value spent at the input end
M_r	Total value of UTXOs paid to payees
M_f	Service charge paid to miner
M_b	UTXO change value
Ω_{out}	Total value of UTXOs at the output end
N_{in}	Number of input UTXOs
N_{out}	Number of output UTXOs
θ_1^k	UTXO expansion speed with SIMO
ϕ_i^k	UTXO expansion speed with MIMO
U	Total UTXO number contained in node memory after the new block is generated
U'	Total UTXO number contained in node memory before the new block is generated
U_{out}	Total UTXO number contained in node memory after the new block is generated with proposed method
Γ_{out}	Expiration time threshold for blockchain address
Δ	The increase of UTXO number after the new block is generated
Δ'	The increase of UTXO number after the new block is generated with proposed method
U_{out}'	Total UTXO number contained in node memory before the new block is generated with proposed method
T_N	Average transaction number per block

3.1 The Expansion of UTXO Set

According to the blockchain running principle mentioned above in Section 1, we can divide blockchain transactions into two types as shown in Fig. 4: Single input multiple output (SIMO) and multiple input multiple output (MIMO). In SIMO type, at the input end there is only one UTXO; in MIMO type, more than one input UTXOs are used.

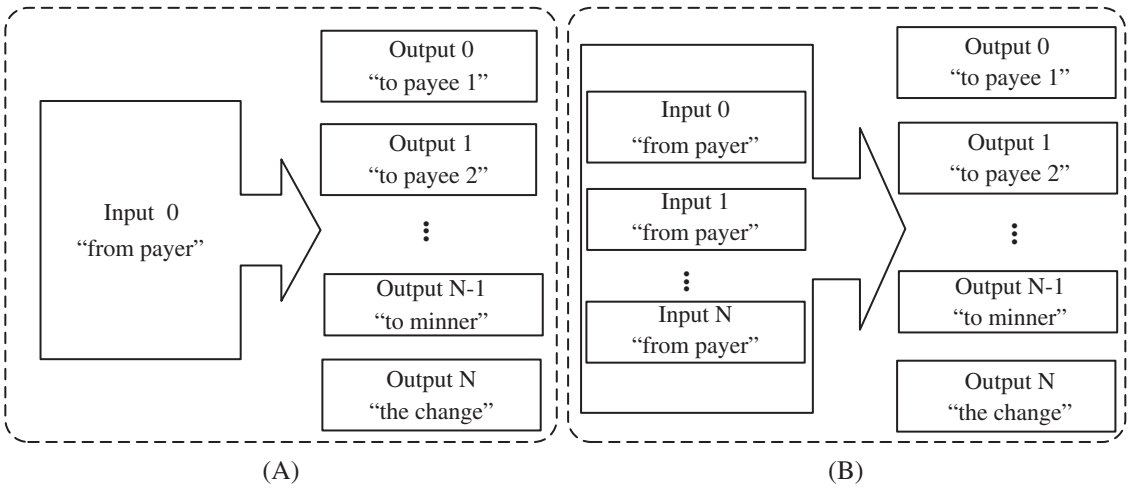

Figure 4: Two different transaction types

Based on the above two transaction types, we theoretically analyze the expansion rate of UTXO set by establishing following mathematical models.

Denote the total input amount as Ω_{in}, the total output amount as Ω_{out}. Assume that in a transaction, *kth* payment value is μ_k, and the total value of UTXOs paid to the payee is M_r, then we have $M_r = \sum_{k=1}^{j} \mu_k$. We denote the UTXO value for the service charge as M_f, the change output as M_b. According to the balancing principle of input and output ends, we have the following model:

$$\Omega_{in} = \Omega_{out} = \sum_{k=1}^{j} \mu_k + M_f + M_b \tag{1}$$

3.1.1 The SIMO Expansion Speed

In SIMO type, a transaction has only one input UTXO. Let the value of input UTXO be v_1, and $\Omega_{in} = v_1$. Total amount paid to payees be $M_r = \sum_{k=1}^{j} \mu_k$. Then, the total output amount is $\Omega_{out} = \sum_{k=1}^{j} \mu_k + M_f + M_b$, According to Eq. (1), there is:

$$v_1 = \sum_{1}^{k} \mu_k + M_f + M_b \tag{2}$$

Based on Eq. (2), the SIMO transaction only consumes one UTXO, so the number of input UTXOs is: $N_{in} = 1$. Meanwhile, the SIMO transaction will generate $k + 2$ UTXOs, and the number of output UTXOs is: $N_{out} = k + 2$. Here, k is the payee number. We can summarize the formula of UTXO expansion speed for SIMO transactions as follows:

$$\theta_1^k = \begin{cases} 2, & k = 1 \\ k + 1, & k > 1 \end{cases} \tag{3}$$

3.1.2 The MIMO Expansion Speed

In MIMO type, one transaction has more than one input UTXOs. Assume that the number of input UTXOs is i, the value of *ith* input UTXO is v_i, then the total amount of all input UTXOs is: $\Omega_{in} = \sum_{i=1}^{s} v_i$. In the MIMO case, according to Eq. (1), there is:

$$\sum_{i=1}^{s} v_i = \sum_{k=1}^{j} \mu_k + M_f + M_b \tag{4}$$

According to Eq. (2), the MIMO transaction will consume $i(i > 1)$ UTXOs, so the number of input UTXOs is: $N_{in} = i$. One MIMO transaction will generate $k + 2$ output UTXOs, and the number of output UTXOs is: $N_{out} = k + 2$, where k denotes the output UTXOs. For MIMO, we can represent the expansion speed as follows:

$$\phi_i^k = \begin{cases} 3 - i, & k = 1 \\ k + 2 - i, & k > 1 \end{cases} \tag{5}$$

3.1.3 The Expansion Speed Per Block

Each transaction can be treated as a random event, and we use $\{X_i Y_k\}$ to represent a transaction event, where X_i indicates this transaction contains i input UTXOs, k indicates that the transaction has k payee addresses. For instance, a SIMO transaction can be expressed as $\{X_1 Y_k\}$. Suppose that the transaction number in a block is T_N, we can express the total UTXO number before the new block is generated as U'. The UTXO number increase after adding a new block is expressed as Δ. Therefore, after a new block is generated, the calculation formula is: $U = U' + \Delta$. The representation of Δ in our model is shown as follows:

$$\Delta = \left(\sum_{k=1}^{j} \theta_1^k \mathrm{P}\{X_1 Y_k\} + \sum_{i=1}^{s} \sum_{k=1}^{j} \phi_i^k \mathrm{P}\{X_i Y_k\}\right) \times T_N \tag{6}$$

By substituting Eqs. (3) and (5), we get the following mathematical model:

$$\Delta = \left(\sum_{k=1}^{j} (k+1)\mathrm{P}\{X_1 Y_k\} + \sum_{i=1}^{s} \sum_{k=1}^{j} (k+2-i)\mathrm{P}\{X_i Y_k\}\right) \times T_N \tag{7}$$

It can be concluded from Eq. (7):

1. There is a positive correlation between the expansion speed of UTXO set and the transaction number;
2. In the normal bitcoin-like blockchain system, the in-memory UTXO set increases linearly with the block chain grows.

3.2 The Expiration Recognition Mechanism Based on LRU

All blockchain nodes maintain an in-memory UTXO set. The purpose of establishing the UTXO set is to generate and verify new transactions. The loss of the private keys means that some UTXOs will never be used again and always resides in the in-memory UTXO set. Although various methods have been proposed to save the private key, there are still many blockchain users lost their private keys. These lost UTXOs are not helpful for the node to verify the transaction, and occupy the precious node memory. The current blockchain system lacks the recognition mechanism, so these UTXOs will still be stored in node memory, occupying additional memory space.

To solve the above issue, this paper introduces a scheme to optimize the UTXO storage based on Least Recently Used (LRU) algorithm. The scheme sets the expiration policy and identifies the UTXO expired regularly. If it searched out an expired UTXO, it moves the UTXO out of running node memory, so as to increase the available memory space.

In this proposed scheme, the UTXO expiration time threshold is set to Γ. When the blockchain system added τ new blocks, the node will identify all current addresses. Assuming that the number of expired addresses each time is d, and the UTXO number under *ith* address is λ_i, the UTXO under the expired address is identified as expired UTXO. Then, the number of expired UTXOs is $\sum_1^d \lambda_i$, and these expired UTXOs will be removed from the in-memory UTXO set. Using the expiration recognition, the average transaction database expansion speed is:

$$\Delta' = \left(\sum_{k=1}^{j} (k+1)\mathrm{P}\{X_1 Y_k\} + \sum_{i=1}^{s} \sum_{k=1}^{j} (k+2-i)\mathrm{P}\{X_i Y_k\}\right) \times T_N - \sum_1^d \lambda_i \Big/ \tau \tag{8}$$

With the proposed expiration recognition mechanism, the number of in-memory UTXOs after adding a new block is:

$$U_{out} = U_{out}^{\cdot} + \Delta - \sum_1^d \lambda_i \Big/ \tau \tag{9}$$

We can get the expansion speed rate of the original blockchain system and the proposed expiration mechanism system:

$$\frac{U}{U_{out}} = \frac{U' + \Delta}{U'_{out} + \Delta - \sum_1^d \lambda_i/\tau} = 1 + \frac{U' - U'_{out} + \sum_1^d \lambda_i/\tau}{U_{out}} \tag{10}$$

As can be seen above, $\frac{U}{U_{out}} > 1$, when using the expiration recognition mechanism. We can see that the expansion rate decreases to $\lambda(\lambda < 1)$: $\lambda = \left(\frac{U}{U_{out}}\right)^{-1}$.

Compared with the original blockchain system, new method slows down the expansion speed of blockchain transaction databases. The proposed method alleviates the storage pressure of full nodes to a certain extent.

4 Evaluation

In this section, we design extensive simulation experiments to validate the new proposed method. The experimental results record the change of UTXO number. The experiments initialize 100 blockchain addresses with 50 digital coins for each. During the transaction process, new blockchain addresses are created randomly. Each generated block contains a coinbase transaction, and the coinbase bonus is set to 25 digital coins. In the following subsections, we introduce the detailed experimental settings and results.

4.1 Experimental Simulation of Original Blockchain System

As mentioned above, two transaction types are studied. In our experiments, the transaction type is randomly selected by the virtual blockchain wallet applications. Two transaction types are with different probabilities. Experiments 1 to 4 respectively simulate the different probabilities of two transaction types.

In experiments (1) to (4), the SIMO transactions are with 40%, 60%, 80% and 100% probabilities respectively. And the four experiments have been carried out with three different trading settings. Under setting 1, 5 transactions are generated averagely per block; under setting 2 and 3, the transactions numbers are 10 and 15. The experimental results are shown in Fig. 5, and we can see the linear growth trends clearly under different settings.

Fig. 5 shows the UTXO changes under four different settings when each block has the different transactions. The UTXO growth trend is basically linear, and there is a positive correlation between the UTXO number and the transaction number. The more the transactions contained in each block, the faster the UTXOs grow.

Fig. 6 shows the UTXO growth trends when each block has almost the same transaction number with different transaction type probabilities (40%, 60%, 80% and 100%). It can be seen from the figure, the higher SIMO probability, the faster UTXO growth rate. This is due to less UTXOs consumed by SIMO transaction type.

4.2 Experimental Results for Expiration Recognition Mechanism

This subsection simulates the UTXO changes, the number of expired addresses and the expired UTXOs in the in-memory transaction databases. To comprehensively test the proposed expiration recognition mechanism, the experiments are under different trading settings. In experiments (1) to (4), the SIMO transactions are with 40%, 60%, 80% and 100% probabilities respectively. In our test, 5 transactions are generated averagely in one block. The experimental results are shown in Fig. 7.

As shown in Fig. 7, if the expiration recognition mechanism is used, a part of the blockchain addresses and UTXOs that may not be used for a long time can be identified. The UTXO number increases with the continuous system running time. The proposed optimization scheme removes the expired UXTOs from the in-memory storage. It reduces the expansion speed of the in-memory database and saves the memory space.

Fig. 8 shows the in-memory UTXO comparisons with the SIMO probabilities at 40%, 60%, 80% and 100% respectively. In the four experiments, per block averagely includes five transactions. As can be seen from the figure, using the proposed optimization scheme, with the block height grows, the proposed method can keep the in-memory UTXO number in a relative low level. In the long run, the proposed scheme can greatly improve the in-memory storage for blockchain transaction databases.

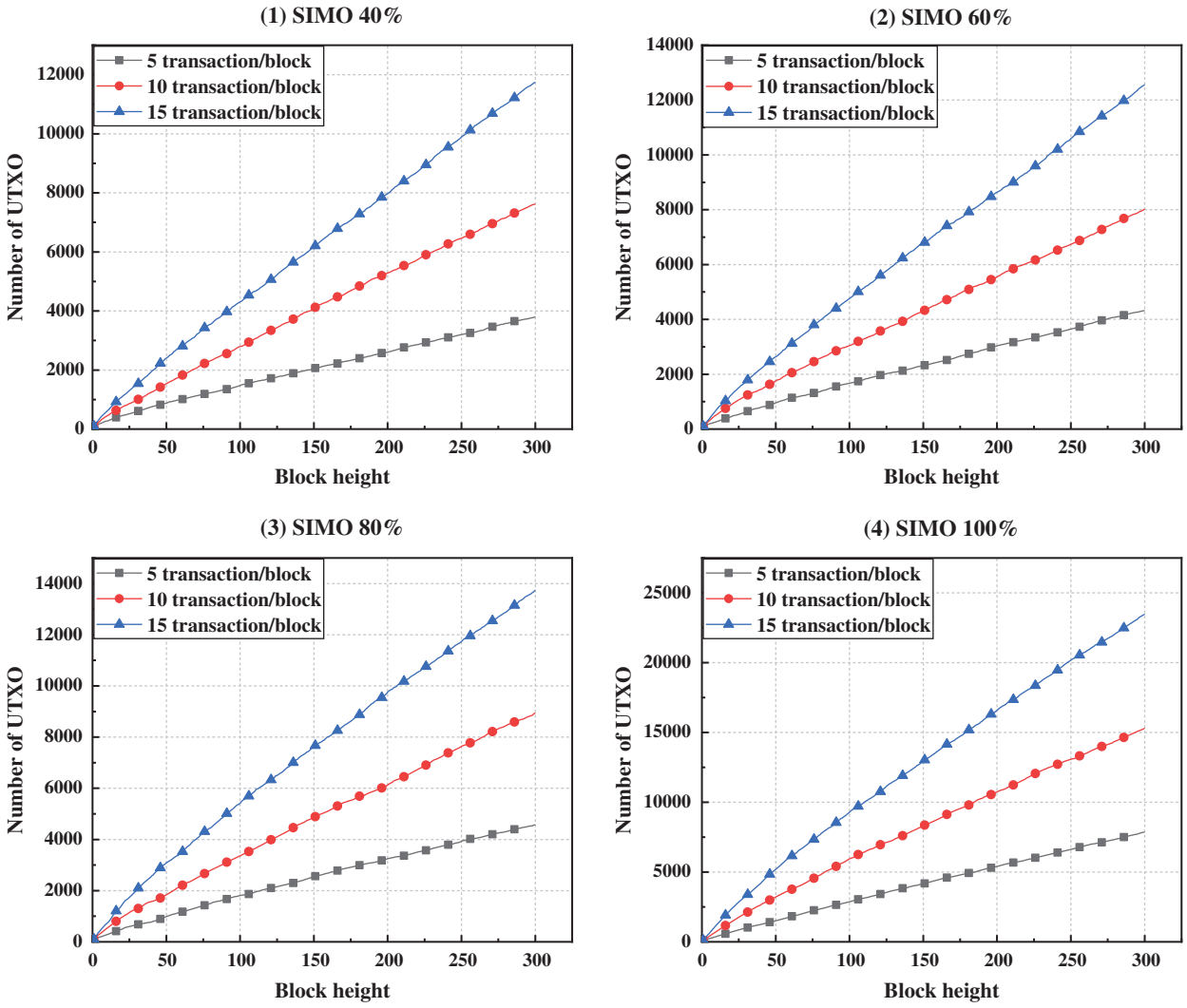

Figure 5: In-memory UTXO changes under different transaction numbers

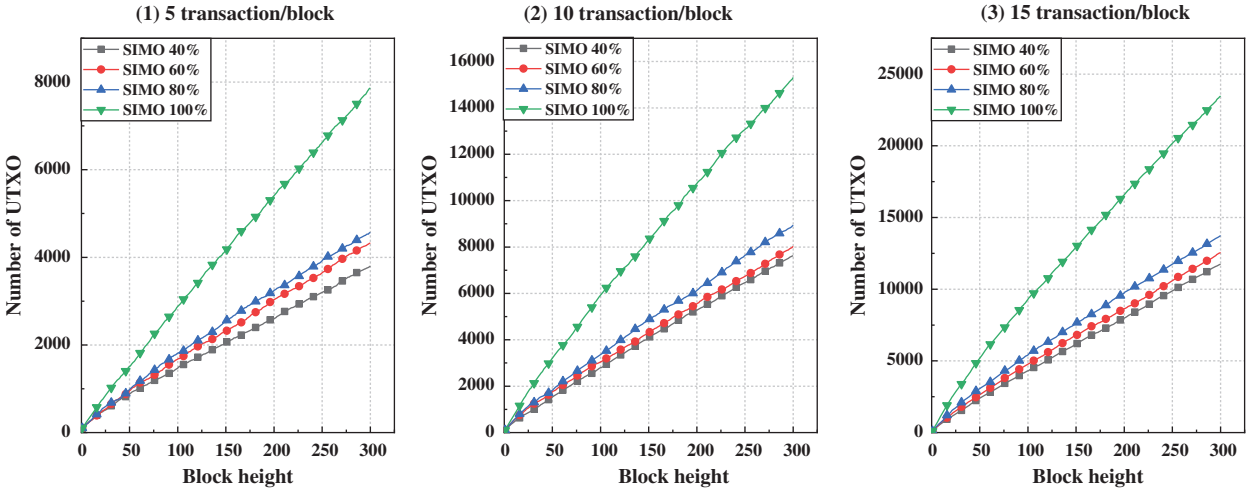

Figure 6: In-memory UTXO changes with different SIMO rates

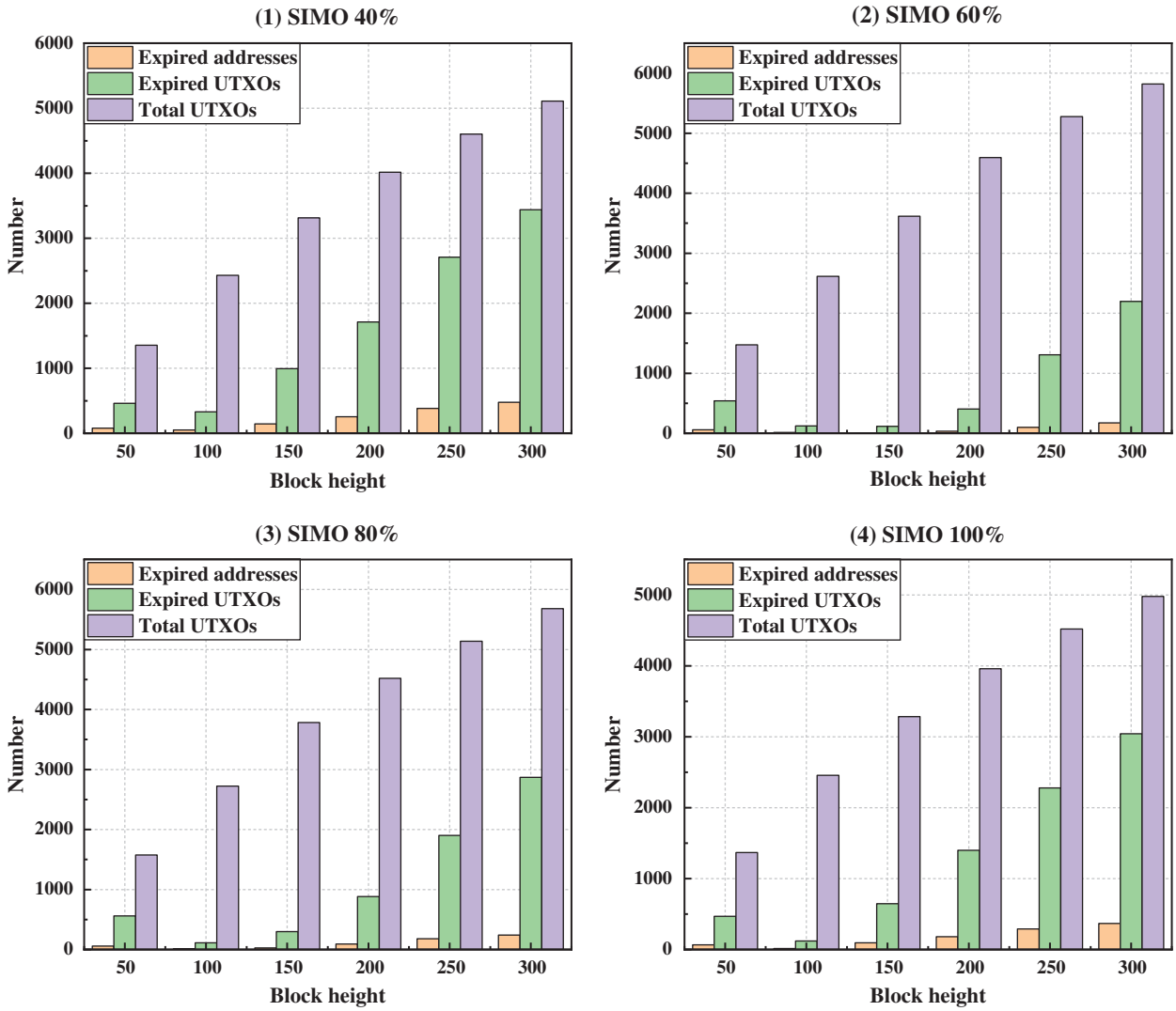

Figure 7: Comparisons for different SIMO rates

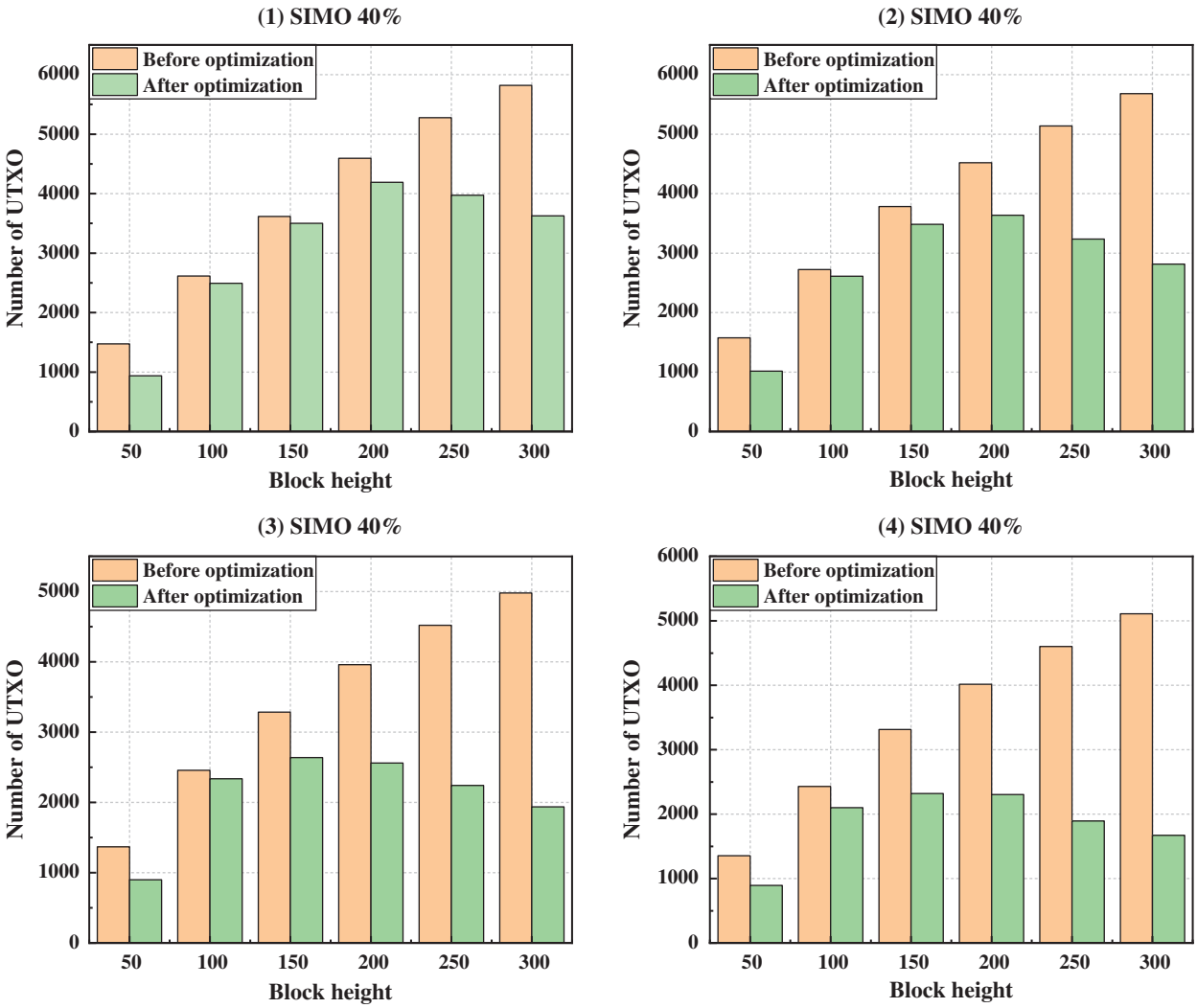

Figure 8: In-memory UTXO comparisons with different SIMO rates

5 Conclusion

The blockchain system requires each connected node to keep a complete transaction database copy which needs to be loaded in memory. That means for full nodes, they will pay more memory capacity to finish the transaction verification. Especially in the big data era, the increase of network traffic will lead to the rapid growth of the connected nodes and the growing blockchain transactions databases. This paper analyzes blockchain transaction databases and proposes a storage optimization scheme. The scheme divides blockchain transaction database into cold zone and hot zone using expiration recognition method based on LRU algorithm. It can achieve storage optimization by moving some UTXOs outside the in-memory transaction databases. We present the theoretical analysis on the optimization method to validate the effectiveness. Designed extensive experiments show our proposed method outperforms the current mechanism for the blockchain transaction databases. The expansion speed of in-memory UTXO set can be improved with the proposed method.

Acknowledgement: We thank Researchers Supporting Project (No. RSP-2020/102) King Saud University, Riyadh, Saudi Arabia, for funding this research. We would also thank the support from the National Natural Science Foundation of China (Nos. 61802031, 61772454, 61811530332, 61811540410), the Natural Science Foundation of Hunan Province, China (No. 2019JGYB177), the Research Foundation of Education Bureau of Hunan Province, China (No. 18C0216), the "Practical Innovation and Entrepreneurial Ability Improvement Plan" for Professional Degree Graduate students of Changsha University of Science and Technology (No. SJCX201971) and Hunan Graduate Scientific Research Innovation Project, China (No. CX2019694). We appreciate the help of the Programs of Transformation and Upgrading of Industries and Information Technologies of Jiangsu Province (No. JITC-1900AX2038/01).

Funding Statement: This work is supported by Researchers Supporting Project (No. RSP-2020/102) King Saud University, Riyadh, Saudi Arabia, the National Natural Science Foundation of China (Nos. 61802031, 61772454, 61811530332, 61811540410), the Natural Science Foundation of Hunan Province, China (No. 2019JGYB177), the Research Foundation of Education Bureau of Hunan Province, China (No. 18C0216), the "Practical Innovation and Entrepreneurial Ability Improvement Plan" for Professional Degree Graduate students of Changsha University of Science and Technology (No. SJCX201971) and Hunan Graduate Scientific Research Innovation Project, China (No. CX2019694). This work is also supported by the Programs of Transformation and Upgrading of Industries and Information Technologies of Jiangsu Province (No. JITC-1900AX2038/01).

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] W. Li, H. Xu, H. Li, Y. Yang, P. K. Sharma *et al.,* "Complexity and algorithms for superposed data uploading problem in networks with smart devices," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5882–5891, 2020.

[2] J. Wang, Y. Zou, L. Peng, L. Wang, O. Alfarraj *et al.,* "Research on crack opening prediction of concrete dam based on recurrent neural network," *Journal of Internet Technology*, vol. 21, no. 4, pp. 1161–1169, 2020.

[3] J. Zhang, C. Wu, D. Yang, Y. Chen, X. Meng *et al.,* "HSCS: A hybrid shared cache scheduling scheme for multi-programmed workloads," *Frontiers of Computer Science*, vol. 12, no. 6, pp. 1090–1104, 2018.

[4] Y. Chen, M. Zhou, Z. Zheng and D. Chen, "Time-aware smart object recommendation in social Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2014–2027, 2019.

[5] Y. Chen, M. Zhou and Z. Zheng, "Learning sequence-based fingerprint for magnetic indoor positioning system," *IEEE Access*, vol. 7, pp. 163231–163244, 2019.

[6] Y. Chen, M. Zhou, Z. Zheng and M. Huo, "Toward practical crowdsourcing-based road anomaly detection with scale-invariant feature," *IEEE Access*, vol. 7, pp. 67666–67678, 2019.

[7] Q. Zhang, S. Yang, M. Liu, J. Liu and L. Jiang, "A new crossover mechanism for genetic algorithms for steinertree optimization," *IEEE Transactions on Cybernetics*, pp. 1–12, 2020.

[8] Q. Zhang, L. Ding and Z. Liao, "A novel genetic algorithm for stable multicast routing in mobile *Ad Hoc* networks," *China Communications*, vol. 16, no. 8, pp. 24–37, 2019.

[9] Y. Chen, J. Tao, L. Liu, J. Xiong and K. Yang, "Research of improving semantic image segmentation based on a feature fusion model," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, pp. 1–13, 2020.

[10] J. Zhang, S. Zhong, T. Wang, H. Chao and J. Wang, "Blockchain-based systems and applications: A survey," *Journal of Internet Technology*, vol. 21, no. 1, pp. 1–14, 2020.

[11] L. V. D. Horst, K. R. Choo and N. Le-Khac, "Process memory investigation of the bitcoin clients electrum and bitcoin core," *IEEE Access*, vol. 5, pp. 22385–22398, 2017.

[12] S. Morishima and H. Matsutani, "Accelerating blockchain search of full nodes using GPUs," in *Proc. PDP*, Cambridge, UK, pp. 244–248, 2018.

[13] P. Ni, H. Li and D. Pan, "Analysis of bitcoin backbone protocol in the non-flat model," *Science China Information Sciences*, vol. 63, no. 3, pp. 60–73, 2020.

[14] S. Park, S. Im, Y. Seol and J. Paek, "Nodes in the bitcoin network: Comparative measurement study and survey," *IEEE Access*, vol. 7, pp. 57009–57022, 2019.

[15] M. A. Imtiaz, D. Starobinski and A. Trachtenberg, "Characterizing orphan transactions in the bitcoin network," in *Proc. ICBC*, Toronto, ON, Canada, pp. 1–9, 2020.

[16] S. N. Mohanty, K. C. Ramya, S. S. Rani, D. Gupta, K. Shankar *et al.,* "An efficient lightweight integrated blockchain (ELIB) model for IoT security and privacy," *Future Generation Computer Systems*, vol. 102, pp. 1027–1037, 2020.

[17] L. Zhou, C. Ge and C. Su, "A privacy preserving two-factor authentication protocol for the bitcoin SPV nodes," *Science China Information Sciences*, vol. 63, no. 3, pp. 130103:1–130103:15, 2020.

[18] A. Kiran, S. Dharanikota and A. Basava, "Blockchain based data access control using smart contracts," in *Proc. TENCON*, Kochi, India, pp. 2335–2339, 2019.

[19] A. Gervais, S. Capkun, G. O. Karame and D. Gruber, "On the privacy provisions of bloom filters in lightweight bitcoin clients," in *Proc. ACSAC*, New Orleans, LA, USA, pp. 326–335, 2014.

[20] J. Sun, X. Yao, S. Wang and Y. Wu, "Blockchain-based secure storage and access scheme for electronic medical records in IPFS," *IEEE Access*, vol. 8, pp. 59389–59401, 2020.

[21] S. Ferretti and G. D'Angelo, "Foreword to the special issue on cryptocurrencies and blockchains for distributed systems," *Concurrency and Computation Practice and Experience*, vol. 155, pp. 1–3, 2020.

[22] Q. Zheng, Y. Li, P. Chen and X. Dong, "An innovative IPFS-based storage model for blockchain," in *Proc. WI*, Santiago, Chile, pp. 704–708, 2018.

[23] Y. Chen, H. Li, K. Li and J. Zhang, "An improved P2P file system scheme based on IPFS and blockchain," in *Proc. BigData*, Boston, MA, USA, pp. 2652–2657, 2017.

[24] J. Wang, Y. Tang, S. He, C. Zhao and A. Tolba, "LogEvent2vec: LogEvent-to-Vector based anomaly detection for large-scale logs in Internet of Things," *Sensors*, vol. 20, no. 9, pp. 2451:1–2451:19, 2020.

[25] J. Wang, W. Wu, Z. Liao, R. S. Sherratt and A. Tolba, "A probability preferred priori offloading mechanism in mobile edge computing," *IEEE Access*, vol. 8, no. 1, pp. 39758–39767, 2020.

[26] M. S. Ali, K. Dolui and F. Antonelli, "IoT data privacy via blockchains and IPFS," in *Proc. IoT*, Linz, Austria, pp. 14:1–14:7, 2017.

[27] Y. Xu, "Section-blockchain: A storage reduced blockchain protocol, the foundation of an autotrophic decentralized storage architecture," in *Proc. ICECCS*, Melbourne, Australia, pp. 115–125, 2018.

[28] S. Ktari, M. Zoubert, A. Hecker and H. Labiod, "Performance evaluation of replication strategies in DHTs under churn," in *Proc. MUM*, Oulu, Finland, pp. 90–97, 2007.

[29] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek *et al.,* "Chord: A scalable peer-to-peer lookup protocol for internet applications," *IEEE ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[30] R. Abe, S. Suzuki and J. Murai, "Mitigating bitcoin node storage size by DHT," in *Proc. AINTEC*, Bangkok, Thailand, pp. 17–23, 2018.

[31] M. Onyeka Okoye, J. Yang, J. Cui, Z. Lei, J. Yuan *et al.,* "A blockchain-enhanced transaction model for microgrid energy trading," *IEEE Access*, vol. 8, pp. 143777–143786, 2020.

[32] S. Bai, G. Yang, C. Rong, G. Liu and H. Dai, "QHSE: An efficient privacy-preserving scheme for blockchain-based transactions," *Future Generation Computer Systems*, vol. 112, pp. 930–944, 2020.

[33] H. Shi, S. Wang and Y. Xiao, "Queuing without patience: A novel transaction selection mechanism in blockchain for IoT enhancement," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7941–7948, 2020.

[34] Z. Guo, Z. Gao, H. Mei, M. Zhao and J. Yang, "Design and optimization for storage mechanism of the public blockchain based on redundant residual number system," *IEEE Access*, vol. 7, pp. 98546–98554, 2019.

[35] R. C. N. Mbinkeu and B. Batchakui, "Reducing disk storage with SQLite into bitcoin architecture," *International Journal of Embedded Systems*, vol. 3, no. 2, pp. 10–14, 2015.

[36] H. Mei, Z. Gao, Z. Guo, M. Zhao and J. Yang, "Storage mechanism optimization in blockchain system based on residual number system," *IEEE Access*, vol. 7, pp. 114539–114546, 2019.

[37] Q. Wang, H. Wang and B. Zheng, "An efficient distributed storage strategy for blockchain," in *Proc. ACM-TURC*, Chengdu, China, pp. 54:1–54:5, 2019.

[38] M. El-Hindi, C. Binnig, A. Arasu, D. Kossmann and R. Ramamurthy, "Blockchain DB—A shared database on blockchains," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, pp. 1597–1609, 2019.

[39] J. Wang, Y. Yang, T. Wang, R. S. Sherratt and J. Zhang, "Big data service architecture: A survey," *Journal of Internet Technology*, vol. 21, pp. 393–405, 2020.

[40] R. Gennaro, S. Goldfeder and A. Narayanan, "Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security," in *Proc. ACNS*, Guildford, UK, pp. 156–174, 2016.