Computing (2022) 104:1497–1524
https://doi.org/10.1007/s00607-022-01063-8

REGULAR PAPER

Scalable blockchain storage systems: research progress and models

Xing Fan[1] · **Baoning Niu**[1] · **Zhenliang Liu**[1]

Received: 25 October 2021 / Accepted: 8 February 2022 / Published online: 28 February 2022

Abstract

Blockchain is believed to be able to build trust among multiple parties and improve the operational efficiency of economy and society, which is attributed to its decentralization property. However, the endless growth of Blockchain data keeps challenging the storage capacity of Blockchain nodes, compromising decentralization, and revealing the issue that the state-of-the-art storage systems of Blockchain are not scalable. From the perspective of improving the scalability of Blockchain storage systems, this paper introduces the logical and physical data structure used by Blockchain storage systems, surveys the current schemes of Blockchain storage systems in terms of the approaches to reducing data redundancy, the corresponding influence on its degree of decentralization and data reliability, and conducts quantitative analysis on data redundancy for schemes of Blockchain storage systems. The study finds that the key to realizing scalable Blockchain storage systems is to deal with the contradiction between data redundancy and its decentralization characteristic. Based on the findings, the node-based scalable model for Blockchain storage systems (SMBSS) is proposed, experimental analysis on the prototypes of the SMBSS is carried out to verify its validity, and future directions for scalable Blockchain storage systems are concluded.

Keywords Blockchain · Schemes of Blockchain storage systems · Data redundancy · Decentralization · Node-based scalable model for Blockchain storage systems

Mathematics Subject Classification 68-02

✉ Baoning Niu
niubaoning@tyut.edu.cn

Xing Fan
fanxing0045@link.tyut.edu.cn

Zhenliang Liu
liuzhenliang@tyut.edu.cn

[1] School of Information and Computer, Taiyuan University of Technology, No. 209, Daxue Street, Jinzhong 030600, Shanxi, China

1 Introduction

Blockchain technology has broad development prospects, as it can ensure data transparency, anti-tampering and traceability without the endorsement of a third-party institution [1]. Blockchain falls into two categories, namely, permissionless Blockchain [2, 3] and permissioned Blockchain [4], according to whether a threshold is set for participants. Permissionless Blockchain is also known as public Blockchain, in which anyone can run a node to join or exit the system arbitrarily, access the data on the chain, submitting transactions, and participate in the process of transaction verification. In contrast, in permissioned Blockchain, a node needs the specific authorization and authentication to join the system, issue transactions and participate in recording on-chain data.

Blockchain technology is composed of encryption technology, a peer-to-peer (P2P) network and a consensus protocol, hence, is an integrated innovation of technology. To ensure decentralization, anti-tampering and traceability, Blockchain adopts two measures. (1) Nodes are functionally equivalent, especially in public Blockchain, where the nodes are completely equitable and the system is fully decentralized. (2) Each node possesses three basic functions including data storage and query, transaction validation, and block mining, to maintain data security and consistency of Blockchain.

However, being anti-tampering and the append-only data structure, Blockchain has to maintain its endlessly growing data. Accordingly, a growing storage resource is needed for a node to store Blockchain data. The nodes with a limited storage resource are forced to exit the Blockchain system, which erodes the decentralization characteristic and restricts the application and development of Blockchain. The major reason is the poor scalability of Blockchain storage systems. Scalability [5] is a characteristic of an organization, system, model, or function describing its capability to cope and perform well under an increased or expanding workload or scope. The scalability of Blockchain storage systems means that the nodes can perform the basic functions as the increase of the volume of Blockchain data, to ensure the characteristics of Blockchain remain unchanged. At present, the poor scalability of Blockchain storage systems is reflected in the following three aspects.

- A node with the basic functions, called full node, has to store the entire Blockchain data locally, which results in excessive data redundancy and wastes huge amount of storage resources. As of June 29, 2021, the total amount of Bitcoin [2] data had reached 328.19 GB [6]. Nearly 10,000 online full nodes [7] need to occupy approximately 3.1 PB of storage space to store approximately 330 GB of data. Ethereum, officially released in 2015, has a more serious scalability issue of its storage system than Bitcoin, released in 2009, because of its diversified data and bulky volume. The total data volume of Ethereum reached 232.50 GB [8]. It has undergone several version updates to optimize its storage structure. Hyperledger Fabric [4], as a representative of permissioned Blockchain, sets access restrictions on nodes, and its data volume grows less rapidly than that of public Blockchain. Even so, its data volume continually grows with the increase of business volume and the number of participating nodes.

- Blockchain utilizes a chain structure and a consensus protocol to maintain its anti-tampering and traceability characteristics. However, as data are continuously appended to the chain, the data volume will eventually hit the storage limit of a full node, forcing a large number of full nodes running on general-purpose equipment to exit the system or convert to lightweight nodes [9, 10]. A lightweight node stores only block headers (see Fig. 1) and its transaction verification function downgrades to only verify transactions related to itself, or payment. As a result, the number of full nodes in Blockchain systems drop, which weakens the decentralization characteristic of Blockchain [11, 12] and threatens data security.
- Full nodes with the block mining function can choose to pack any transaction into a block, which deteriorate functional inequivalence among nodes of different types.

This paper compares and analyzes the existing schemes of Blockchain storage systems from the perspectives of types of Blockchain, degree of decentralization and data reliability, and formalizes them. On this basis, the node-based scalable model for Blockchain storage systems (SMBSS) is proposed, which enables a node to have the basic functions with a low-cost storage system while encouraging nodes with adequate storage resources to store more data in order to improve data availability and ultimately break through the storage performance bottleneck and fundamentally solve the scalability issue of Blockchain storage systems. The main contributions of this paper follow.

- We divide the existing schemes of Blockchain storage systems into three types and elaborate them in terms of the approaches to reducing data redundancy, the degree of decentralization and data reliability, and conduct quantitative analysis on data redundancy for the schemes.
- We propose SMBSS, a scalable model for Blockchain storage systems, and carry out experimental analysis on its prototypes to verify its validity.
- We indicate future directions for scalable Blockchain storage systems.

The rest of the paper is organized as follows. Section 2 briefly introduces the logical and physical data structures of Blockchain data. Section 3 elaborates the typical schemes of Blockchain storage systems proposed in recent years. Section 4 conducts quantitative analysis on data redundancy for schemes of Blockchain storage systems and proposes the SMBSS. Section 5 carries out the prototype implementation of SMBSS to verify its validity. We enumerate several open issues regarding scalable Blockchain storage systems in Sect. 6 and conclude in Sect. 7.

2 Blockchain data structures

Blockchain is a single linked list of blocks consisting of transactions identified by their hash value. We introduce the logical and physical data structures of Blockchain in Sects. 2.1 and 2.2, respectively.

2.1 Logical data structure of blockchain

The logical data structure of Blockchain is shown in Fig. 1. A block is composed of a block header and a block body. The block header holds the basic information of the block, including the version, hash value of the previous block, hash value of the current block, encrypted tree root, nonce, timestamp, and parameters of the consensus protocol. The block body maintains all transaction records and the related information, and organizes these data in the form of an encryption tree, usually a Merkle Tree [13] or its variants [14, 15], to quickly verify whether a transaction exists in the current block [16].

Blockchain utilizes two hash structures, namely, a Merkle Tree and a chain structure, to guarantee its anti-tempering characteristic. Merkle Tree and its variants used in Blockchain include the Merkle Binary Tree [13], Merkle Patricia Tree [14] and Merkle Bucket Tree [15]. In essence, they make use of the property of Merkle Tree to verify the data integrity of Blockchain. Blockchain organizes all processed transactions in a chain of blocks, constituting a sequence of valid transactions [16]. All blocks are linked together through the hash value of the preceding block in accordance with the generation order to form a linked list of blocks. Such a chain structure enables any change to the transactions in a block to be propagated to all subsequent blocks, which make it easier to spot any alteration in the transaction history.

There are two types of data models in Blockchain, i.e., the UTXO-based model and the account-based model [17]. UTXO stands for "unspent transaction output", and is an analogue of gold. The UTXO set is the subset of transaction outputs that has not been spent at a given moment. Bitcoin makes use of the UTXO-based model to keep track of output transactions that have not been spent and can therefore be used as inputs to new transactions. Ethereum and Hyperledger Fabric adopt the account-based model, in which each node typically holds the status data generated by the transactions

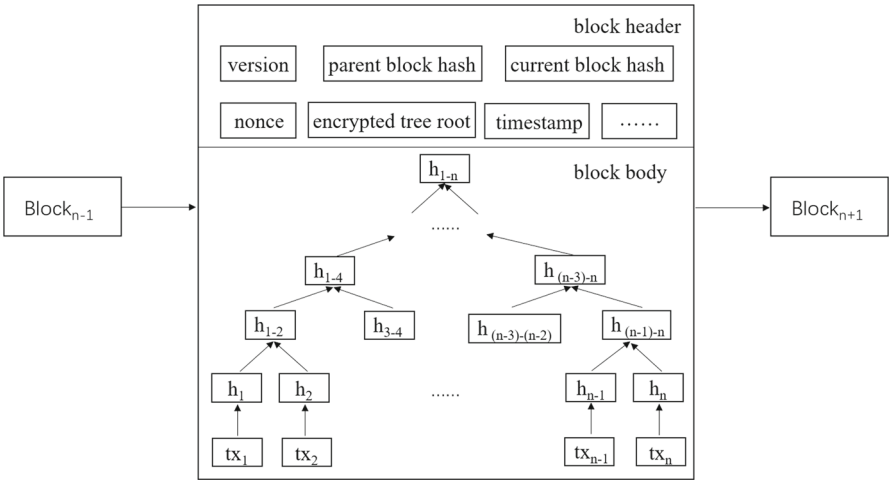

Fig. 1 Logical data structure of blockchain

Table 1 The physical data structure of mainstream Blockchain applications

Blockchain app.	Data structure	Data model	Block storage	Index storage
Bitcoin	Merkle tree/chain structure	UTXO-based	File storage	LevelDB
Ethereum	Merkle patricia tree/chain structure	Account-based	LevelDB	LevelDB
Hyperledger fabric	Merkle bucket tree/chain structure	Account-based	File storage	LevelDB/CouchDB

performed by all accounts, making it easy for nodes to query account balances or status data.

2.2 Physical data structure of blockchain

Blockchain data is stored on disk in the form of a file or database. The former divides Blockchain data into fragments of a fixed size, which is convenient for the append operation. The latter usually uses the LevelDB or another non-relational database [18] to store Blockchain data, which makes it easy to execute queries and modification operations. A key-value database is preferred to store index data to facilitate the block/transaction query operations, where the key refers to the block/transaction hash and the value is a pointer links to the corresponding file. The physical data structure of mainstream Blockchain applications is summarized as follows (Table 1).

3 Schemes of blockchain storage systems

Blockchain is essentially a distributed database, in which scalability is a key barrier, and is being widely used in real business environments. Scalability of Blockchain systems involves three issues: throughput, storage capacity and network communication [19–23]. This paper focuses on the scalability of Blockchain storage systems. The full redundancy of Blockchain data is intended to guarantee the decentralization characteristic, but brings high storage pressures on nodes and causes the scalability issue of Blockchain storage systems, which is detrimental to decentralization as we discussed in Sect. 1. A scalable Blockchain storage system should properly deal with the contradiction between data redundancy and the decentralization characteristic. With regard to this, we divide the existing schemes of Blockchain storage systems into three types, namely, multiple node cooperative storage, data reduction, and sharding, based on their approach to reducing data redundancy and comprehensively evaluate them in Sects. 3.1–3.3, with focuses on the approaches, the degree of decentralization and data reliability.

3.1 Storage schemes with multiple node cooperative storage

The scalability issue of Blockchain storage systems origins from the use of full nodes, and the use of non-full nodes is straightforward to reduce data redundancy. Storage schemes consisting of multiple node types have been proposed.

SSFL: storage scheme with full nodes and lightweight nodes

When Satoshi Nakamoto first proposed the new term "Blockchain", he had already realized that the continuously increasing volume of Blockchain data would cause the storage scalability issue. To address this issue, he proposed a node type which stores only the block header to save storage resource. This type of nodes is called lightweight nodes which need only 80 bytes of storage space for each block. Since there are no transactions stored locally, the transaction verification function is in question. Lightweight nodes are designed to be able to validate payments which are a transaction involving the node itself, by obtaining the payment hash, the Merkle root and the corresponding Merkle tree path from the full nodes and then perform the hash calculation. This kind of verification method is known as Simplified payment verification (SPV) [9] for Bitcoin. Ethereum equivalent to SPV is called the light client protocol [10].

However, the continuous growth of Blockchain data forces a large number of full nodes to convert to lightweight nodes in order to participate in Blockchain with their limited storage resources, which would weaken decentralization characteristic of Blockchain. The lightweight nodes can only verify payments instead of any transactions by themselves. The payment verification relies on full nodes and there is no guarantee of data authenticity when the full nodes are dishonest. With the proposed lightweight nodes storing only block header, SSFL's redundancy is lower than the original storage scheme which has only full node.

SSFL brings new problems. (1) There are two types of nodes in Blockchain, full nodes and lightweight nodes, resulting in the node inequivalence. (2) It is foreseeable that a large number of full nodes with limited storage resources, will be forced to be a lightweight node in order to participate in Blockchain systems, which would increase the retrieval time as well as the considerable communication overheads during the payment verification process.

SSFE: storage scheme with full nodes and enhanced lightweight nodes

The drawback of lightweight nodes is that they can only verify payments, a tiny portion of all transactions for a node. Enhanced lightweight node (enhanced SPV node) or ESPV node [24] is proposed to allow a node to store a small portion of Blockchain data while being able to verify majority transactions. The storage scheme consisting of full nodes and enhanced lightweight nodes is called SSFE. According to the block generation time, ESPV divides blocks into new blocks and old blocks, stores the entire new blocks within a given time window to empower the basic functions, partial old blocks according to the sharding [25] strategy to ensure data reliability and reduce storage requirements. Analogously, virtual block group (VBG) [26] stores partial Blockchain data on each node while guaranteeing the data security and reliability, and VBG storage index is developed to improve query efficiency. While Scalable Storage Model for account-based Blockchain (SSMAB) [27] saves state data in a completely redundant

manner to guarantee its transaction verification function, stores block data in shard storage to reduce redundancy, and adopts an economic incentive mechanism to ensure data availability while reducing storage consumption.

SSFE alleviates the storage pressure on nodes, endows the ESPV nodes with the basic functions, and improves the degree of decentralization compared to SSFL. As more and more ESPV nodes join the Blockchain system and store as much of the old Blockchain data as possible, SSFE can eventually transform into the ideal scalable Blockchain storage systems, which has only one type of node. In this way, function equivalence of nodes and decentralization of the Blockchain systems can be realized. Therefore, developing an effective storage incentive mechanism to adjust the amount of data stored on ESPV nodes is a future research direction.

SSCD: storage scheme with cloud-dew architecture

Dewblock [28] is another scheme trying to endow the nodes deployed on resource limited devices with basic functions by leveraging dew computing [29] and the cloud-dew architecture [30]. Dew computing is an on-premises computer software-hardware organization paradigm in the cloud computing environment where the on-premises computer provides functionality that is independent of cloud services and is also collaborative with cloud services. The goal of dew computing is to fully realize the potentials of on-premises computers and cloud services. Dewblock is composed of a cloud server and a dew server, where the cloud server can be regarded as a full node deploying on a public or private cloud service, and the dew server can operate in one of the two modes: dew mode and full mode. When a dew server is in dew mode, it behaves as a lightweight node which is known as a dew client. The cloud server and the dew client constitute a single Dewblock node. When a dew server is in full mode, it behaves the same with a full node, and we call the dew server a full client. In this mode, the dew server itself constitutes a Dewblock full node, while the cloud server is not involved in the operation.

Essentially speaking, the SSCD consists of the full nodes (cloud server) running on the cloud and the lightweight nodes (dew server). The contradiction between data redundancy and the decentralization characteristic, and the data authenticity problem on SSFL still exists. To make sure that the account in a dew server is consistent with the account in a cloud server, the dew server will periodically send its account information to the cloud server for verification, which incurs communication overhead.

ESS: elastic storage scheme

The difference between SSFE and SSCD is that they take different approaches to endowing a node having limited storage resources with the basic functions. SSFE uses collaborative storage while SSCD leverages dew computing. ElasticChain [31] also takes the collaborative storage approach, but has a much complex mechanism. The storage nodes store parts of the complete chain under the premise of ensuring Blockchain data safety by duplicate ratio regulation algorithm. The earlier a block is generated, the fewer copies are stored, and vice versa. The verification nodes periodically check the reliability of the storage nodes and record them on the proofs of reliability (POR) chain to ensure that the shards are always stored on the storage nodes with good stability, while the locations of the data duplicates are organized as a position (P) chain stored on the user nodes. When a user node queries Blockchain data, it first accesses the P chain on the local disk to get the storage location of the data.

Then, according to the location information, the user node finds the corresponding storage nodes and asks them to return the ciphertext data to the user node. Finally, it reconstructs the ciphertext based on the key, which is saved locally and generated by the POR method, and then obtains the initial data.

Since the data availability and persistence are completely dependent on the storage nodes, and the shard is always stored on storage nodes with good stability, which means that ESS goes against the decentralization characteristic of Blockchain. What's more, ESS stipulates that each shard needs to be stored on more than 50% of the nodes, so the saved storage space and is less than 50% space that the full node only storage scheme uses.

Although ESS reduces the storage pressure on full nodes, the following problems still exist. (1) ESS involves three types of nodesuser nodes, storage nodes and verification nodes and two types of chains–a POR chain and a P chain. Compared with other storage schemes, ESS is more complex, and the node inequivalence is more serious. (2) From the discussions above, the data query process is complex and brings large communication overheads.

CCSS: cluster-based collaborative storage scheme

The collaborative storage in SSFE and ESS is at the system level. CCSS distributes blocks among nodes using Distributed Hash Table (DHT) [32], and allows the collaborative storage at the cluster level. Nodes in a DHT cluster can behave like full nodes without holding the entire of the Blockchain. Each node in the cluster holds the partial Blockchain data assigned according to the DHT algorithm. When verifying new transactions and blocks, a node queries blocks against its DHT cluster. In other words, each node can verify new transactions and blocks by collaborating with other nodes in its cluster. To reduce the storage load on nodes and the propagation delay of the network, a load balancing method based on DHT cluster [33] is proposed, where DHT is implemented by Kademlia. This storage scheme consists of two types of nodes, namely, mining nodes and data nodes, where the mining nodes perform the work of a distributed agreement by PoW (proof of work), the data nodes verify transactions and blocks. The processing procedure consists of two phases: the transaction collect phase, and the block generation and verification phase.

CCSS can reduce storage requirements for nodes by distributing ledger data with DHT. All the nodes in its cluster maintain a complete Blockchain data, which means the more nodes in the cluster, the less storage space required. Thus, its data redundancy depends on the number of the clusters.

However, according to its principle, when a new node joins the cluster, CCSS automatically adjusts the data stored on the original node and redistribute data to the new node which occupies the communication bandwidth in the cluster, and increases the communication overheads.

3.2 Data reduction

Different from storage schemes with multiple node types, which cooperatively store Blockchain data on multiple types of nodes to reduce data redundancy, data reduction compresses or prunes Blockchain data on nodes.

Block compression utilizes coding techniques to compress Blockchain data.

Erasure code storage [34] introduces a new type of node, called erasure code-based low storage node, with the aim of proposing an alternative between light nodes and full nodes. This type of node uses erasure codes in order to allow users with reduced storage capacity to contribute to the Blockchain, without storing the entire Blockchain. Erasure coding [35] decomposes Blockchain data into several fragments, constructs linear combinations among these fragments and then stores these coded fragments in nodes. In order to generate its coded fragments, will split the initial block into fragments. When a node needs to query a specific block, it obtains the corresponding linear combination and performs the inverse operation to reconstruct the block. Compared with storing the uncompressed Blockchain data in a full node, this approach can reduce the number of backups and storage requirements under the same data availability metrics. Erasure codes involve linear combination and its inverse operation, and incur computational overhead. The storage scheme is applicable to the cold data, and the saving on storage is limited. Similarly, Dai et al. [36] propose network coding based distributed storage (NC-DS) for Blockchain to relieve the storage scalability problem. Its basic idea is to divide the Blockchain data into k equal length packets and encode them into n packets, and store each of them in a node. The encoding method ensures arbitrary k of n packets are able to recover the original data. However, there exists shortcomings. (1) Given the characteristics of Blockchain that the number of participants cannot be estimated in advance, it is difficult to get the appropriate parameter values of the encoding method. (2) Manipulation of Blockchain data has to deal with data integrity and security since different nodes store distinct encoded packets.

Blockchain pruning removes unimportant historical data after a predefined time period on the premise of ensuring the data security of Blockchain systems while reducing the storage pressure on nodes.

Based on selective Blockchain transaction pruning, Palm et al. [37] introduce an approach that allow a node to independently select and remove any already applied transactions, in order to reduce ledger size. In contrast to existing erasure approaches attempting to erase data globally from all nodes, Florian et al. [38] propose a pragmatic local erasure solution which empowers node operators to remove problematic transactions from local storage, with minimal impact to their capacity to support the network and autonomously validate further transactions for UTXO-based Blockchain. In addition, Mini-Blockchain [39], Rollerchain [40] and pruneable sharding-based Blockchain protocol [41] also adopt pruning strategy to save storage resources.

Ethereum utilizes reference counting [42] to track leaf nodes in the state trie, treats the data without reference in the latest 5000 blocks as invalid data, and permanently deletes it from the database. Similarly, Monero [43] prunes the unnecessary information including the account addresses, ring signatures and other Blockchain data to save the storage resources according to the predetermined rule, on the premise of ensuring Blockchain data security. Jidar [44], a new data simplification method for Bitcoin, proposes that the nodes only store the transactions of interest and the related Merkle branches. When the transaction submitter verifies a transaction, it should provide the Merkle branches related to the transaction input to the network, together with the transaction information to ensure the transaction verification function of nodes. If the

nodes should obtain a complete block, they need to rely on the incentive mechanism to request data from other nodes and integrate all the fragments.

To alleviate the storage pressure on nodes, Bitcoin proposes block file pruning [45]. Nodes only keep the UTXO index and the latest 288 blocks after synchronizing to the latest Blockchain data. Once the network or the local Blockchain data incurs errors, the nodes have to resync the entire Blockchain data to build the UTXO index. The risk of resync increases with the growing volume of Blockchain data. The node inequivalence still exists and decentralization will be weakened.

To summarize, the above storage schemes related to Blockchain pruning are based on deleting meaningless or insignificant historical data to reduce the storage pressure on nodes. However, these storage schemes fail to meet the requirements of scalable Blockchain storage systems. (1) The Blockchain pruning is applicable to public Blockchain at present. Compared with public Blockchain, the application fields of the permissioned Blockchain are more diverse, the data types are more complex, and it is more difficult to define whether the historical data is invalid data. (2) The node inequivalence exists. (3) The authenticity of transaction verification is in doubt since the transaction information is dependent on the transaction submitter. (4) Full nodes are required to have the ability to maintain the complete Blockchain data, and therefore, the storage schemes can only marginally reduce the storage pressure. (5) It is difficult to define the importance of historical data apart from practical applications. Thus, deleting data blindly to save storage space is not conducive to the maintenance of data reliability of Blockchain systems, which is also contrary to the autonomy of Blockchain.

3.3 Sharding

The basic idea of Sharding [25] is a divide-and-conquer strategy, which partitions the nodes into several units to process transactions in parallel or maintain the disjointed Blockchain ledgers, the former is called transaction sharding [46, 47], while the latter is called state sharding [48, 49]. Transaction sharding implements parallel processing of transactions, and achieves an almost linear extension of transaction throughput. Each node, however, still has to store the complete Blockchain data. State sharding stores a part of the block data of a shard to reduce storage and computing pressures.

Serveral storage schemes [24, 26, 27, 31–33] discussed in Sects. 3.1 and 3.2 involve sort of state sharding as an auxiliary technique. The core idea is that the nodes store partial data [24, 26, 27, 31] or all the nodes of a cluster cooperate to store a complete Blockchain data [32, 33]. Compared to the general sharding [46–49], these storage schemes do not change the consensus mechanism, and retain the reliability and security of the original Blockchain system.

In conclusion, although the general sharding strategies reduce the storage pressure on Blockchain systems, it has the three limitations. (1) The frequent cross-shard communication overheads will increase the bandwidth pressure, prolong the transaction confirmation time and ultimately reduce the data availability and persistence of Blockchain systems. (2) The main challenge with sharding is to ensure consistency among multiple shards, especially in the case of cross-shard transactions, which need

a large communication overhead. (3) It is also very difficult to choose the appropriate shard size; a large shard size may affect the performance of Byzantine consensus algorithms, while a small shard size not only has a bad effect on scalability due to the large number of inter-shard transactions but also threatens the security of Blockchain systems.

4 Scalable model of blockchain storage systems

In the previous section, we have discussed the state-of-the-art storage schemes with the focus on the scalability of Blockchain storage systems. These storage schemes have two major shortcomings. (1) To the best of our knowledge, there is not a mechanism controlling data redundancy which leads to either the poor scalability of Blockchain storage systems or the detriment to decentralization. (2) As full nodes running on general-purpose devices with limited storage resources have to give up the basic functions to become non-full nodes, a large number of non-full nodes depend on a few full nodes, which cause node inequality and compromise decentralization. The key to solving these problems is to resolve the contradiction between data redundancy and its decentralization characteristic.

To build a scalable model of Blockchain storage systems, we start by conduct data redundancy analysis on the schemes in Sect. 4.1, find the key factors which has to do with the scalability of Blockchain storage systems by comparing the schemes in Sect. 4.2, and finally propose the node-based scalable model for Blockchain storage systems (SMBSS) in Sect. 4.3.

4.1 Data redundancy

The scalability of Blockchain storage systems means reducing data redundancy without compromising decentralization. In the ideal Blockchain storage system, the number of types of nodes should be as small as possible. In the optimal case, there is only one type of node in the Blockchain system, to ensure the functional equivalence of nodes. We expect that the minimum redundancy of Blockchain systems could be as small as possible, and it is determined by all the types of nodes in order to ensure the decentralization characteristic of Blockchain systems.

Definition 1 (*Redundancy*) The redundancy of the i^th block ($block_i$) is the number of nodes that store it and is denoted as K_i. The average redundancy refers to the average value of redundancy of all the blocks and denoted as K , while the maximum and minimum redundancy refer to the maximum and minimum values among all the redundancy of blocks, denoted as K_{max} and K_{min}, respectively.

With this definition, we now discuss the data redundancy of **SSFL**. There are two types of nodes in SSFL. The full nodes store the entire blocks. The lightweight nodes only store block headers, which is negligible. The minimum redundancy of SSFL is determined by the number of full nodes, namely, n_f.

SSFE SSFE involves two types of nodes, the full nodes and the ESPV nodes. Both of them store blocks. Its minimum redundancy is jointly determined by them. Its

minimal data redundancy is $n_f + (n_{ESPV} * a/M)$, where n_{ESPV} is the number of ESPV nodes, a is the number of blocks that the ESPV nodes store, M is the number of blocks, and the number of ESPV nodes is much larger than that of full nodes, i.e., $n_{ESPV} \gg n_f$. In the optimal case, there is only one type of node in the Blockchain system, i.e., ESPV nodes, to ensure the functional equivalence of nodes, its minimal data redundancy is $(n_{ESPV} * a/M)$.

SSCD Nodes of SSCD are a combination of the full nodes which are running on the cloud server and the lightweight nodes which are deployed to personal computers and mobile devices. Its data redundancy depends on the full nodes and is n_f.

ESS ESS stipulates that each shard needs to be stored on more than 50% of the nodes, so its minimal data redundancy is $n_s/2$, where n_s is the number of storage nodes.

CCSS According to CCSS protocol, all the nodes in its cluster maintain a complete Blockchain data which means its minimal data redundancy depends on the number of clusters, i.e., k.

Block compression usually have a compressing rate, and the minimal data redundancy depends on the full nodes and the erasure code-based low storage nodes and is $n_f + n_c * k/r$, where n_c is the number of coding nodes.

Blockchain pruning schemes are based on SSFL, and its minimal data redundancy is same as SSFL, or n_f.

Sharding According to the analysis on sharding, the minimal data redundancy of transaction sharding is n_n, where n_n is the number of nodes in the entire Blockchain systems. While the minimal data redundancy of state sharding is $\sum_{i=1}^{k} n_i s_i / M$, where n_i represents any node which stores s_i blocks.

To sum up, although there are many schemes of Blockchain storage systems aimed at scalable storage, most of them are optimized on the basis of SSFL, and their minimum redundancy is n_f, which indicates that the data availability is still largely dependent on the number of full nodes in Blockchain systems. However, the minimum redundancy of SSFE, ESS, CCSS and state sharding is different from that of those other storage schemes. The minimum redundancy of ESS, CCSS and state sharding are $n_1/2$, k and $\sum_{i=1}^{k} n_i s_i / M$, respectively, which is lower than that of SSFL. This goes against the original intention of improving the scalability of Blockchain storage systems. While the minimum redundancy of SSFE and Block Compression are $n_f + n_c * k/r$ and $n_f + (n_{ESPV} * a/M)$, respectively, which is higher than that of SSFL.

4.2 Comparison among various schemes of blockchain storage systems

Based on the discussions above, we find that the scalability of Blockchain systems is mainly reflect on three aspects: minimal data redundancy, degree of decentralization and data reliability.

Decentralization, the key characteristic of Blockchain is measured by node equivalence and the level of transaction verification. The level of transaction verification can be classified into four levels: L1, unable to verify payments; L2, verify all the payments; L3, verify partial transactions; and L4, verify all the transactions. The ideal

type of nodes should maintain the basic functions and all the nodes should jointly maintain a complete ledger to achieve Blockchain systems autonomy.

Blockchain utilizes a P2P network to realize communication and data sharing among nodes, which means that in order to ensure high data reliability, a node should store as many Blockchain data as possible. The data reliability of a Blockchain system involves two aspects, namely, data availability and retrieval time.

Definition 2 (*Data Availability*) Data availability is the probability that data can be retrieved at a given time and denoted as P_a^i.

The probability that data on a node can be retrieved can be measured by the online rate of the node, which is the mean time to failure (MTTF) divided by the total observation time (MTTF+MTTR, mean time to repair).

Definition 3 (*Retrieval time*) Retrieval time of data is the time interval from the data request submitted to the data received, and is denoted as T.

The best data availability and retrieval time can be simply achieved by storing all the Blockchain data, which cause excess redundancy and scalability issue. On the contray, less data redundancy usually means lower data availability, which leads to increase object retrieval times. Unfortunately, longer retrieval times could compromise data maintenance processes and penalize user's retrieval times [50]. It is crucial issue for P2P storage systems to handle the contradictions among them.

Based on the above discussions, we conduct the comparison among the schemes of Blockchain storage systems (Table 2). The retrieval time is the negative indicators; that is, their evaluation is negatively correlated with performance. We expect the degree of decentralization as well as data reliability could be as high as possible while maintain the higher K_{min}. The high data reliability requires high data availability and a short retrieval time. We choose the general storage scheme, i.e., SSFL as the baseline for comparison.

We can see that regarding the degree of node equivalence, although there exist two types of nodes in SSFL and SSFE, ESPV nodes in SSFE have more complete function than the lightweight nodes, so its node equivalence is higher than that of SSFL. Moreover, since more than two types of nodes are involved in ESS and Block Compression, the node asymmetry problem is more serious than that of SSFL. In addition, there are two types of nodes in SSCD, CCSS and Blockchain pruning, so the degree of node equivalence in those three storage schemes is same as that in SSFL. Considered that nodes of Sharding is more complicate since a node can take on multiple roles, so we do not discuss it in this part.

The evaluation results of the level of transaction verification are based on different types of nodes of various storage schemes. In SSFL and Blockchain Pruning, lightweight nodes can verify all the payments (L2), full nodes can verify all the transactions (L4). In SSFE, ESPV nodes can verify partial transactions (L3), full nodes can verify all the transactions (L4). In SSCD, depends on the mode, dew servers can verify all the payments (L2) or verify all the transactions (L4), full nodes can verify all the transactions (L4). In ESS, storage nodes and verification nodes cannot verify payments (L1), user nodes can verify all the transactions (L4). In CCSS, mining nodes

Table 2 Comparison among the schemes of Blockchain storage systems

Storage scheme	Instance of storage schemes	Reference	K_{min}	Degree of decentralization		Data reliability	
				Node equivalence	Level of trans. verification	Data availability	Retrieval time
Storage schemes with multiple node types	SSFL	[9, 10]	n_f	Fair	L2, L4	Fair	Fair
	SSFE	[24, 26, 27]	$n_f + (n_{ESPV} * a/M)$	High	L3, L4	Fair	High
	SSCD	[28]	n_f	Fair	L2, L4	Low	High
	ESS	[31]	$n_f/2$	Low	L1, L1, L4	Low	High
	CCSS	[32, 33]	k	Fair	L1, L4	Low	High
Data	Block compression	[34, 36]	$n_f + n_c * k/r$	Low	L1, L2, L4	Low	High
Reduction	Blockchain pruning	[37–45]	n_f	Fair	L2, L4	Low	High
Sharding	Transaction sharding	[46, 47]	n_n	–	L4	Low	High
	State sharding	[48, 49]	$\sum_{i=1}^{k} n_i s_i / M$	–	L3	Low	High

cannot verify payments (L1), data nodes can verify all the transactions (L4). In Block Compression, coding nodes cannot verify payments (L1), lightweight nodes can verify all the payments (L2), full nodes can verify all the transactions (L4). In Transaction Sharding, nodes can verify all the transactions (L4). While in State Sharding, nodes can verify partial transactions (L3).

As for data availability, ESPV nodes in SSFE have basic functions than the lightweight nodes in SSFL, and their data availability can reach 100% in both SSFL and SSFE (see Sect. 5). SSCD relies on cloud servers, and the high coupling between systems and services may lead to a decrease in data availability. In ESS, its data availability depends on the storage nodes and P chain, and the resulting communication overhead will reduce the data availability. According to the rule in CCSS, the nodes joining or exiting the system will cause a reallocation of data, which will reduce data availability. Block Compression adopts coding method, the data reconstruction process would decrease data availability. Although Blockchain pruning is based on SSFL, deleting data blindly only to save storage space is not conducive to the maintenance of data reliability. In addition, the local data reconstruction operation will waste time and computing resources. The frequent cross-shard transactions in Sharding lead to a decrease in data availability.

Ideally, the data retrieval time should be as short as possible, so the retrieval time is a negative indicator. Compared with SSFL, ESPV nodes in SSFE store only part of the Blockchain data, so the retrieval time is increased slightly (see Sect. 5). While the retrieval time of SSCD is affected by cloud server and network communication, its robustness is poor. The data acquisition in ESS depends on the storage nodes and P chain, which is complicated to operate and will bring a huge communication overhead. As with the analysis of data availability, the retrieval time in CCSS is longer than that of SSFL. Data reduction involves data reconstruction operations, and the cross-shard transactions in Sharding will increase the retrieval time.

Based on above, we can conclude that by implementing enhanced lightweight nodes and equipping them with the basic functions as full nodes, the minimum redundancy of SSFE has changed to $n_f + (n_{ESPV} * a/M)$. Considering the limited storage capacity, the number of ESPV nodes in Blockchain systems is much greater than that of full nodes. Therefore, if we could increase $(n_{ESPV} * a/M)$, then the storage pressure on the full nodes would be transferred to the enhanced lightweight nodes to reduce the threshold of the nodes. Compared with SSFL, SSFE can reduce the storage pressure on the full nodes while reducing the data redundancy as well as improving the scalability of Blockchain storage systems.

4.3 The approach for scalable blockchain storage systems

Since Blockchain data grows linearly with time, the increasing storage pressure on full nodes will force the full nodes to switch roles to non-full nodes or eventually exit the system and will ultimately weaken the decentralization characteristic of Blockchain. As a P2P system, Blockchain should take advantages of all participating nodes, encourage more nodes to participate in data storage, and jointly maintain high data availability and reduce the retrieval time. We discuss each data model in Blockchain separately.

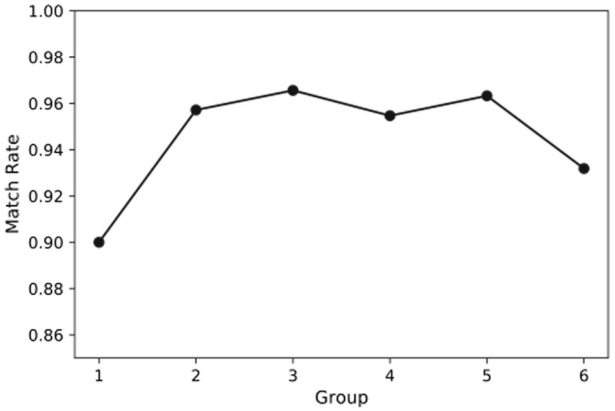

Fig. 2 Overall transaction verification rate distribution

We analyze the UTXO of Bitcoin and find that most UTXOs are generated in a few recent blocks. We take 100,000 blocks as a group, extract 100 blocks of them at random, and calculate the ratio of verified transactions for them in the last 3000 blocks; the mean value is obtained and shown in Fig. 2.

It can be seen that 90% of the transactions can be verified by referring to the most recent blocks within the time window of the current block containing the transactions [24, 51]. The experimental results provide a new idea: compared with full nodes that need to store the entire Blockchain data in order to have the transaction verification function, saving only the latest 3000 blocks can independently verify more than 80% of the transactions of a block, which is obviously more in line with the scalability requirements of Blockchain storage systems.

Unlike Bitcoin, in addition to the original Blockchain data, Ethereum, the representative of the account-based Blockchain, maintains the current state of all accounts in the Merkle Patricia Tree, i.e., state trie. The approach seems highly inefficient at the first glance, because it needs to store the entire state with each block. The efficiency, however, is comparable to that of Bitcoin in reality. The reason is that all of the state information is part of the last block, hence, there is no need to store the entire Blockchain history. As long as the node saves the complete state trie, it can have the transaction verification function. We can conclude that the transaction validation rate for both SSMAB nodes and full nodes achieve 100 percent, so there is no experiment on transaction validation rate for account-based Blockchain in Sect. 5.2.

4.4 Node-based scalable model for blockchain storage systems

The current schemes of Blockchain storage systems might temporarily relief the storage pressure on nodes. To find a permanent solution, quantitative analysis of the relationship between data redundancy and scalability is necessary. In the ideal Blockchain storage system, the number of node types is as small as possible, all the participated nodes adaptive storage of data based on their storage conditions, to ensure

Table 3 SMBSS for blockchain systems

Storage scheme	Instance of storage schemes	Description using SMBSS
Storage	SSFL	$\{(0, L2, remote), (100\%, L4, local)\}$
Schemes	SSFE	$\{(a/M, L3, local), (100\%, L4, local)\}$
with	SSCD	$\{(0, L2, remote), (100\%, L4, local)\}$
Multiple	ESS	$\{(50\%, L1, local), (0, L1, remote), (0, L4, remote)\}$
Node types	CCSS	$\{(0, L1, remote), (100\%, L4, local)\}$
Data	Block compression	$\{(0, L1, remote), (k/r, L2, remote), (100\%, L4, local)\}$
Reduction	Blockchain pruning	$\{(0, L2, remote), (100\%, L4, local)\}$
	Transaction sharding	$\{(100\%, L4, local)\}$
Sharding	State sharding	$\{(s_i/M, L3, local)\}$

the autonomy of Blockchain systems while equipping the nodes with the basic functions.

Based on the above conclusions, we propose the node-based scalable model for Blockchain storage systems (SMBSS), which stores the new blocks within the time window for UTXO-based Blockchain or the entire state trie for account-based Blockchain in order to guarantee its transaction verification function, as well as partial old blocks to reduce the consumption of storage resources.

Definition 4 (*Node-based scalable model for blockchain storage systems*) $M = \left\{\left(c^i, D_T^i, P_r^i\right) \mid i = 1, 2, \ldots m\right\}$, M consists of m node resource types which constitute the Blockchain systems. A node resource types is a triple $\left(c^i, D_T^i, P_r^i\right)$, where c^i is the percentage of data Backup of the i^{th} node resource type, D_T^i is its level of transaction verification, and P_r^i represents its data reliability which manifests in two ways for a node to obtain data, i.e., remote and local.

The schemes of Blockchain storage systems discussed in Sect. 4.1 can be collectively described by SMBSS with different parameter combinations, as shown in Table 3.

The system architecture for SMBSS is shown in Fig. 3. A node stores two types of data, the new blocks within the time window for the UTXO-based Blockchain or the state trie for the account-based Blockchain, and multiple shards according to their storage capacity. The storage strategy allows all the nodes have the transaction verification function by storing the first type of data, while share storage loads by storing the second type of data.

In a P2P system, the system reliability is given by [52]:

$$r = \sum_{i=m}^{n} C_n^i p^i (1-p)^{n-i} \tag{1}$$

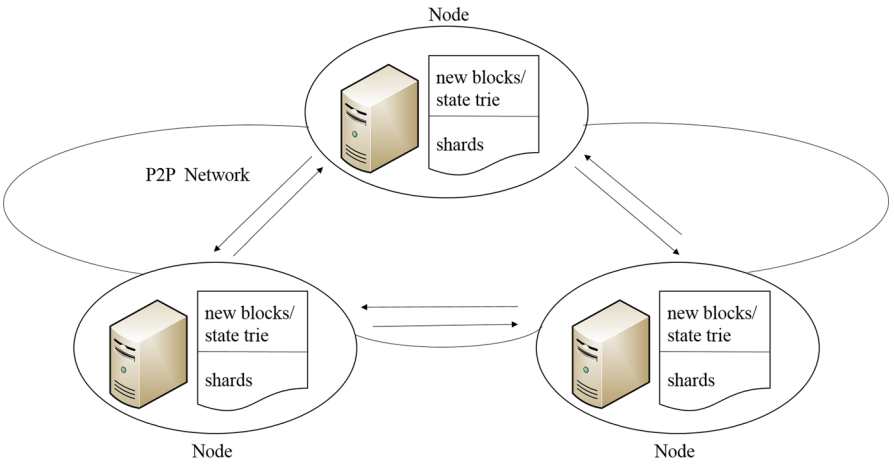

Fig. 3 The system architecture for SMBSS

where r is the system reliability, p is the node reliability, and k is the redundancy of the Blockchain system; then, the redundancy k is given by:

$$k = log\,(1 - r)\,/log\,(1 - p) \tag{2}$$

According to Borel's Law [53], any probability below 1 in 10^50 is automatically zero. According to Formula (2), we set $r = 10^{50}$.

5 SMBSS prototype implementation and validation

According to the theory of the SMBSS, we propose endowing the nodes with the same functions as full nodes on the premise of making full use of their storage resources in order to increase the data redundancy while reducing the nodes' requirements for storage resources, ensuring decentralization characteristic. In this section, we implement an SMBSS prototype on both UTXO-based Blockchain and account-based Blockchain to verify the validity of SMBSS by comparing its performance with that of SSFL.

5.1 Performance evaluation on UTXO-based blockchain

In this section, we take SSFE as the prototype implementation of SMBSS on UTXO-based Blockchain, i.e., Bitcoin, and compare its performance with that of SSFL.

The experiment is conducted on a server with an Intel (R) Xeon (R) E5-2609 v4 1.70 GHz CPU and 16 GB memory. Simulation experiments are performed on Bitcoin data from the origin block up to block number 640091, which is 300 GB of data. We use Bitcoin-ETL [54] to convert the Bitcoin block files to JSON files for further processing.

According to Bitnodes statistics, as of December 19, 2019, the maximum, minimum and average number of full nodes online simultaneously in Bitcoin are 12,335, 9488

Table 4 Node properties

Item	Distribution	Reliability	Bandwidth
Full node	0.3	0.9	10 MB/s
ESPV node	0.7	0.1	[0.2 MB/s, 5 MB/s]

Table 5 Evaluation criteria of SMBSS

Evaluation criteria	Description
Transaction validation rate	The percentage of verified transactions within the given set of transactions
Data availability	The probability that data can be retrieved at a given time
Retrieval time	The time interval from the data request to obtain the data
Storage cost	The required storage space for the participated node

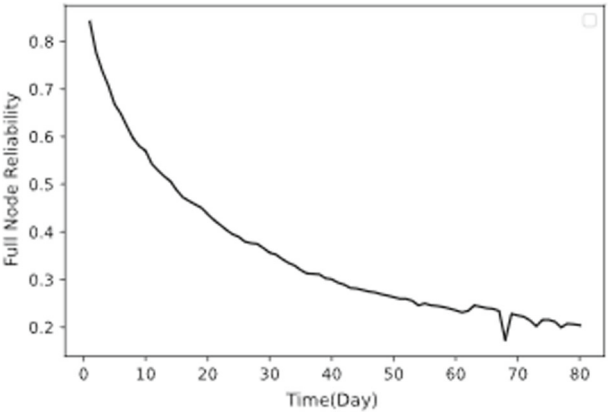

Fig. 4 Full node reliability of Bitcoin

and 10,174, respectively. The number of full nodes online simultaneously has a certain stability, since Bitcoin has become a relatively mature application. Therefore, we monitor the online full nodes to obtain the number of full nodes online simultaneously within 80 days, and the experimental results are shown in Fig. 4.

From the figure above, we set the node reliability $p = 0.1$; then, using Formula (2), we can obtain the redundancy $k \approx 1000$, which means that as long as there are more than 1000 full nodes online at the same time, data availability can be guaranteed.

The performance evaluation indexes are the transaction verification rate, data availability, retrieval time and storage space. The experiment establishes 10,000 nodes in total, and the node properties are shown in Table 4. Suppose that one round of trial and error incurs a delay of 10 ms.

The details of the evaluation criteria of SMBSS for both UTXO-based Blockchain and account-based Blockchain are shown in Table 5.

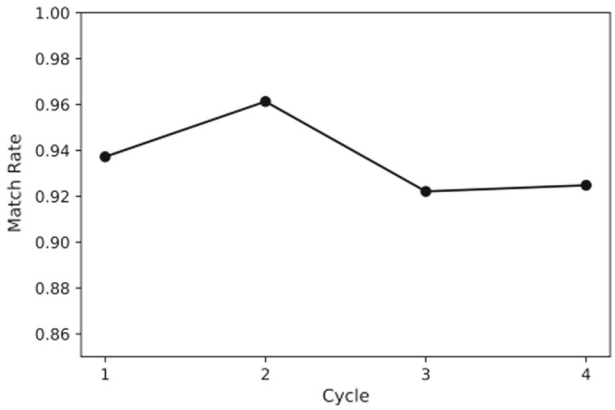

Fig. 5 Transaction verification rate

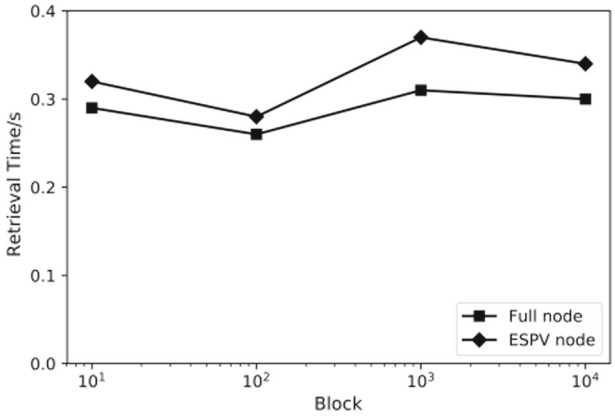

Fig. 6 Retrieval time

Transaction validation rate The experimental dataset is a set of Bitcoin blocks from the height of the 56,8201st block to the 640,091st block. We organize 2016 blocks as a cycle and randomly select 10% of the dataset for sampling to verify the ESPV nodes' transaction verification rate.

As seen from Fig. 5, the ESPV nodes' transaction verification rate is greater than 0.92 in each case. This means that the ESPV nodes will obtain a fraction greater than 0.9 of the transaction verifications' function on the premise of occupying approximately 3 GB of storage space; this kind of storage scheme greatly alleviates the storage pressure on the full nodes, reduces data redundancy, and conforms to the design purpose of the SMBSS.

Data availability and retrieval time Although the nodes in a P2P network are free to join or exit, considering that Bitcoin is a relatively mature Blockchain application, the number of participating nodes is basically constant at any time. Since SPV nodes do not store Blockchain data, they are not involved in this experiment.

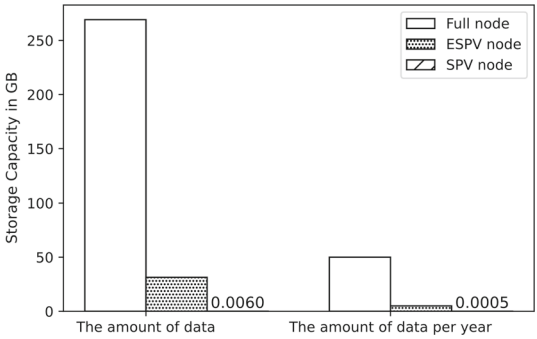

Fig. 7 Storage space

Experimental results show that data availability of the two kinds of nodes remain at 100%, which indicate that the ESPV nodes can obtain the data within the target retrieval time as full nodes. As shown in Fig. 6, the retrieval time of ESPV nodes is slightly longer than that of full nodes when data are available. This is because ESPV nodes do not keep all of the Blockchain data and have to rely on other nodes to synchronize data, which causes a delay. However, the difference in the retrieval time between full nodes and ESPV nodes is very small, which means that the lower latency has little impact on a single node. If millisecond latency can reduce the multiple levels of storage resources, then this cost is acceptable. Additionally, the ESPV nodes encourage more nodes that run on general-purpose equipment to participate in Blockchain systems, so this strategy meets the application requirements, reduces the pressure on full nodes and can be applied in an actual production environment.

Storage cost To quantitatively measure the storage space required by the different types of nodes in the two storage schemes, we calculate statistics on the storage space required by full nodes, ESPV nodes and SPV nodes to store the entire Blockchain data and the Blockchain data within the last year in Fig. 7.

As shown in the figure, the storage space required by full nodes, ESPV nodes and SPV nodes decreases sequentially. However, considering that ESPV nodes need only 31.37 GB of storage space to complete more than 80% of the transaction verifications of the other two types of nodes, they can obviously give better consideration to both saving storage resources and verifying transactions. In addition, through the comparison of the annual Blockchain data for full nodes and ESPV nodes, it can be seen that the growth rate of the ESPV nodes' annual storage data volume is much less than that of the full nodes, which is in line with the hardware conditions of general equipment and is conducive to improving the scalability of Blockchain storage systems. Supposing that there are 10,000 online nodes in Blockchain systems, assuming that 30% of them are full nodes and 70% are ESPV nodes, the whole Blockchain system can save approximately 1.58 PB of storage space.

5.2 Performance evaluation on account-based blockchain

In this section, we select Ethereum as the representative of account-based Blockchain and implement a prototype implementation of SMBSS on it, which is denoted as the

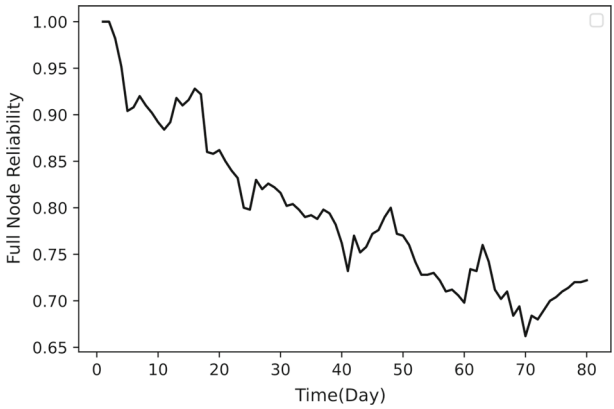

Fig. 8 Full node reliability of Ethereum

scalable storage model for account-based Blockchain (SSMAB). Unlike Bitcoin, in addition to the original Blockchain data, Ethereum maintains the current state of all accounts in the Merkle Patricia Tree, i.e., state trie. As of May 26, 2020, the volume of the Ethereum state trie was 2.8 GB. Based on above observations, we propose a different storage scheme aiming at different Blockchain data; that is, each node stores the entire state trie and part of the Ethereum data based on the sharding strategy.

The experiment is conducted on two servers with an Intel (R) Xeon (R) E3-1225 CPU @3.20 GHz and 64 GB memory. The bandwidth is a random number from 4 Mbps to 100 Mbps, and the experimental data are the original Ethereum data.

According to Ethstats [55] statistics, as of August, 14, 2020, the maximum, minimum and average number of full nodes online simultaneously in Ethereum are 10,295, 4466 and 7418, respectively. The number of full nodes online simultaneously has a certain stability. Therefore, we monitor the online full nodes to obtain the number of full nodes online simultaneously within 80 days, and the experimental results are shown in Fig. 8.

From the figure above, we set the node reliability $p = 0.5$; then, using Formula (2), we can obtain the redundancy $k \approx 47$. We simulate the Kademlia protocol of Ethereum on PeerSim [56]. The experiment deploys 6000 nodes in total, and the reliability of each node is set to 0.5, among which 80 nodes with specific shards are randomly labeled.

Data availability As mentioned above, there are 80 out of 6000 nodes containing specific data. In this experiment, supposing that each node wants to sync the specific data with the random probabilities, we can obtain the data availability by retrieving the specific data of shards (Fig. 9).

The experimental results show that in the simulated network, when the number of nodes online simultaneously with the specific data is greater than 52, the shard availability can reach 100%. The reason that the experimental number of nodes online simultaneously is larger than the theoretical value, i.e., $k \approx 47$, is that the data transmission in the P2P network is influenced by network congestion and other factors, and therefore, the experimental value is higher than the ideal value in the simulated

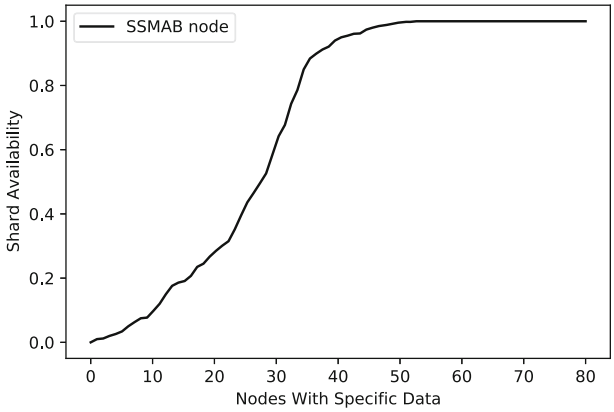

Fig. 9 Shard availability

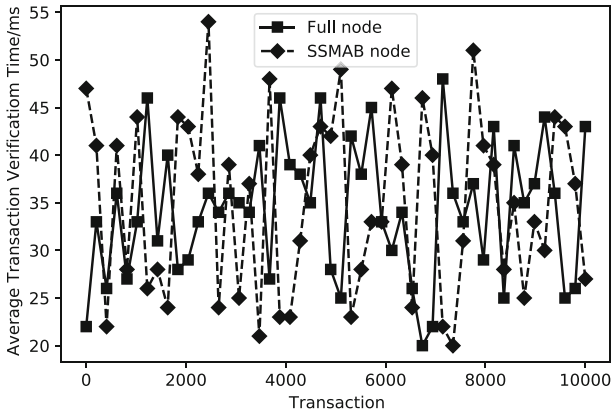

Fig. 10 Average transaction verification time

network. It can be predicted that in a real network, the number of nodes online simultaneously with specific data should be more due to network congestion, routing, network bandwidth and many other factors.

Retrieval time is measured through average transaction verification time and average block generation time. From the Figs. 10 and 11, we can conclude that both the average transaction verification time and the average block generation time of the two types of nodes share the same trend. Although SSMAB nodes spend more time on transaction verification and block generation than those on full nodes, it is acceptable using the small performance loss to gain huge storage saving as shown in Fig. 12.

Storage cost To quantitatively measure the storage space required by different schemes of Blockchain storage systems, we determine the storage space required by Ethereum the full modes and the SSMAB nodes to store all the Blockchain data and the Blockchain data within the last year.

As shown in the Fig. 12, for both the amount of data and the amount of data per year, the storage space required by SSMAB nodes is only 13% that of the full nodes.

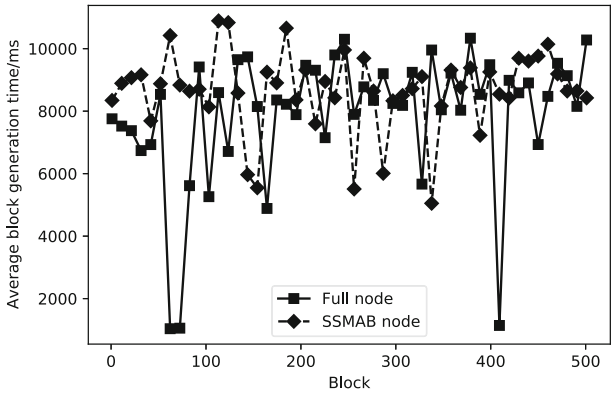

Fig. 11 Average block generation time

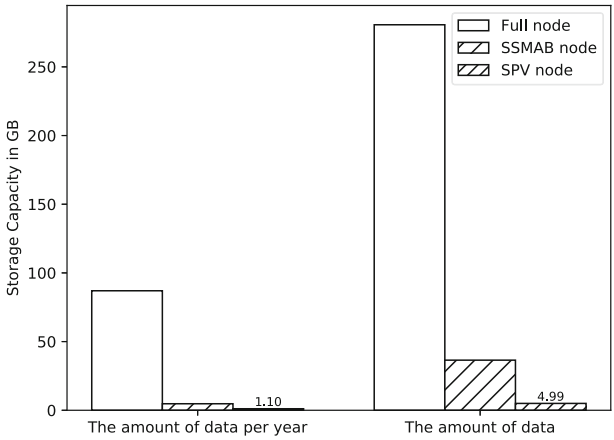

Fig. 12 Storage space

For SSMAB nodes, considering that it saves storage resources and has the complete functionality of full nodes, the additional cost of transaction verification and block generation is acceptable.

6 Research directions

The storage scalability of Blockchain systems is one of the key factors affecting the application and development of Blockchain, and it has become a popular research topic in the field of Blockchain. We discuss possible future directions with respect to three areas.

• Storage incentive mechanisms.

According to the storage capacity of each node, SSMBS aims to encourage nodes to store partial Blockchain data to maintain the basic functions of full nodes. Blockchain systems is ultimately composed of only one type of node, and all the nodes jointly

maintain the Blockchain data. The key to achieving this ultimate goal is to encourage the nodes to store as much block data as possible. It is very important to design a unified storage incentive mechanism, although there exist some difficulties in the design of storage incentive mechanisms: (1) balancing the interest relationship among the nodes, (2) guaranteeing the security of the stimulation reward mode, (3) determining the content of the incentive payment scheme, (4) ensuring the scalability of the reward mechanism, (5) ensuring privacy protection for the nodes, and (6) optimizing the resulting communication overhead during data sharing. In short, there is still much room to explore in this direction.

• Establish a practical quantitative relationship model between the data redundancy and data reliability of Blockchain systems.

An ideal scalable storage model for Blockchain systems should reduce data redundancy while ensuring high data reliability. In theory, reduced data redundancy would lead to reduced data persistence and data availability as well as increased retrieval time. Therefore, it is important to conduct quantitative research on data redundancy, data persistence, and data availability as well as the retrieval time of Blockchain systems and establish a practical quantitative relationship model to perform adaptive adjustment between data redundancy and data reliability.

• The optimization of the dissemination protocol for information in Blockchain.

Nodes broadcast blocks and transactions to the Blockchain network. To maintain the block propagation time while increasing the block size, the average bandwidth of the whole system that determines the block propagation time becomes a performance bottleneck of Blockchain systems. It is meaningful to improve propagation protocols for Blockchain systems, such as finding a better routing mechanism to improve the scalability of Blockchain systems.

7 Conclusion

Blockchain technology has developed rapidly in the past few years, and it can be predicted that Blockchain will be applied in many fields in the future. Focusing on the scalability of Blockchain storage systems, we first introduce the Blockchain data structure. Second, we give a brief survey of recent studies on scalable schemes of Blockchain storage systems with proper depth and sufficient breadth, and then we analyze the storage scalability of Blockchain systems from the perspectives of the minimal data redundancy, the degree of decentralization and data reliability. Finally, we summarize and propose the SMBSS and verify its validity by conducting experimental analysis on both UTXO-based Blockchain and account-based Blockchain. The experimental results show that the prototypes of SMBSS can meet the Blockchain scalability requirements. The potential research directions and open issues include the storage incentive mechanisms, the quantitative models for the relationship between data redundancy and decentralization characteristic of Blockchain systems, and the optimization of the dissemination protocol. With this comprehensive survey, we expect that our analysis and the SMBSS will inspire many further studies dedicated to improving the scalability of Blockchain storage systems.

Acknowledgements This research is supported by National Natural Science Foundation of China (62072326) and the International Cooperation Project of the Major Research, Development Program of Shanxi (201903D421007) and Hubei Key Laboratory of Optical Information and Pattern Recognition, Wuhan Institute of Technology (201903).

References

1. Yuan Y, Wang FY (2016) Blockchain: the state of the art and future trends. Acta Autom Sin 42(4):481–494
2. Nakamoto S (2009) Bitcoin: a peer-to-peer electronic cash system. Preprint at https://bitcoin.org/bitcoin.pdf
3. Buterin V (2014) Ethereum white paper: a next generation smart contract & decentralized application platform. Preprint at https://ethereum.org/en/whitepaper/
4. Androulaki E, Manevich Y, Muralidharan S, Murthy C, Laventman G (2018) Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Paper presented at the 13th European conference on computer systems, 23–26 April 2018
5. Hayes A (2020) Investopedia. figshare https://www.investopedia.com/terms/s/scalability.asp
6. BtcBitaps. figshare https://btc.bitaps.com (2021)
7. Bitnodes. figshare https://bitnodes.io/dashboard/?days=730 (2021)
8. EthBitaps. figshare https://eth.bitaps.com/ (2021)
9. BitcoinJ. figshare https://bitnodes.io/dashboard/?days=730
10. Mckinney J, Light client protocol. figshare https://github.com/ethereum/wiki/wiki/Light-client-protocol
11. Zhang Z, Wang G, Xu J, Du X (2020) Survey on data management in blockchain systems. J Softw 31(9):2903–2925
12. Yuan Y, Wang FY (2020) Editable blockchain: models, techniques and methods. Acta Autom Sin 46(5):831–846
13. Merkle RC (1980) Protocols for public key cryptosystems. In: Paper presented at 1980 IEEE symposium on security and privacy, 14–16 April 1980
14. Buterin V, Merkling in ethereum. figshare https://blog.ethereum.org/2015/11/15/merkling-in-ethereum
15. Xu J, Wei L, Zhang Y, Wang A, Zhou F, Gao CZ (2018) Dynamic fully homomorphic encryption-based Merkle tree for lightweight streaming authenticated data structures. J Netw Comput Appl 107:113–124
16. Shao QF, Jin CQ, Zhang Z, Qian WN, Zhou AY (2018) Blockchain: architecture and research progress. J Softw 41(5):969–988
17. Buterin V, Thoughts on UTXOs. figshare https://medium.com/@ConsenSys/thoughts-on-utxo-by-vitalik-buterin-2bb782c67e53
18. Bitcoin core 0.11. figshare https://en.bitcoin.it/wiki/Bitcoin_Core_0.11_(ch_2):_Data_Storage
19. Xie J, Yu FR, Huang T, Xie R, Liu J, Liu Y (2019) A survey on the scalability of blockchain systems. IEEE Net 33(5):166–173
20. Shuai Z, Yong Y, Xiao-Chun N, Fei-Yue W (2019) Scaling blockchain towards bitcoin: key technologies, constraints and related issues. Acta Autom Sin 45(6):1015–1030
21. Politou E, Casino F, Alepis E, Patsakis C (2019) Blockchain mutability: challenges and proposed solutions. IEEE Trans Emerg Top Comput, arXiv:1907.07099
22. Zhou Q, Huang H, Zheng Z (2020) Solutions to scalability of blockchain: a survey. IEEE Access 8:16440–16455. https://doi.org/10.1109/ACCESS.2020.2967218
23. Sun Z, Zhang X, Xiang F, Chen L (2021) Survey of storage scalability on blockchain. J Softw 32(1):1–20
24. Zhao YL, Niu BN, Li P, Fan X (2019) A novel enhanced lightweight node for blockchain. In: Paper presented at the 1st blockchain and trustworthy systems, 7–8 December 2019
25. Shard wiki. figshare https://en.wikipedia.org/wiki/Shard_(database_architecture
26. Yu B, Li X, Zhao H (2020) Virtual block group: a scalable blockchain model with partial node storage and distributed hash table. Comput J 63(10):1524–1536
27. Zhang XH, Niu BN, Gong T (2021) Account-based blockchain scalable storage model. Preprint at https://kns.cnki.nct/kcms/detail/11.2625.V.20210316.1345.002.html

28. Wang YW (2020) A blockchain system with lightweight full node based on dew computing. Internet of Things 10(3):100184
29. Wang Y (2016) Definition and categorization of dew computing. Open J Cloud Comput 3(1):1–7
30. Wang YW (2015) Cloud-dew architecture. Int J Cloud Comput 4:199–210
31. Jia DY, Xin JC, Wang ZQ, Wei GG, Wang GR (2018) ElasticChain: support very large blockchain by reducing data redundancy. In: Paper presented at the 2018 Asia-Pacific Web (APWeb) and web-age information management (WAIM) joint international conference on web and big data, 23–25 July 2018
32. Jia DY, Xin JC, Wang ZQ, Wei GG, Wang GR (2018) Mitigating bitcoin node storage size by DHT. In: Paper presented at the 2018 Asian internet engineering conference, 12–14 November 2018
33. Kaneko Y, Asaka T (2018) DHT clustering for load balancing considering blockchain data size. In: Paper presented at the 6th international symposium on computing and networking workshops (CANDARW), 27–30 November 2018
34. Perard D, Lacan J, Bachy Y, Detchart J (2018) Erasure code-based low storage blockchain node. In: Paper presented at the 2018 Asian internet engineering conference, 30 July–3 August 2018
35. Li J, Li B (2013) Erasure coding for cloud storage systems: a survey. Tsinghua Sci Technol 3:259–272
36. Li J, Li B (2018) A low storage room requirement framework for distributed ledger in blockchain. IEEE Access 2018:22970–22975
37. Palm E, Schelen O, Bodin U (2018) Selective blockchain transaction pruning and state derivability. In: Paper presented at the 2018 crypto valley conference on blockchain technology (CVCBT), 20–22 June 2018
38. Florian M, Henningsen S, Beaucamp S, Scheuermann B (2019) Erasing data from blockchain nodes. In: Paper presented at the 2019 IEEE European symposium on security and privacy workshops (EuroS&PW), 17–19 June 2019
39. Bruce JD, The mini-blockchain scheme. figshare http://cryptonite.info/files/mbc-scheme-rev3.pdf
40. Chepurnoy A, Larangeira M, Ojiganov A (2019) Rollerchain, a blockchain with safely pruneable full blocks. Preprint at http://arxiv.org/abs/1603.07926
41. Feng X, Ma J, Miao Y, Meng Q, Liu X, Jiang Q, Li H (2019) Pruneable sharding-based blockchain protocol. Peer-to-Peer Netw Appl 12:934–950
42. Buterin V, State Tree Pruning. figshare https://blog.ethereum.org/2015/06/26/state-tree-pruning/
43. Ehrenhofer J, Monero adds blockchain pruning and improves transaction efficiency. figshare https://web.getmonero.org/zh-cn/2019/02/01/pruning.html
44. Dai X, Xiao J, Yang W, Wang C, Jin H (2019) Jidar: a jigsaw-like data reduction approach without trust assumptions for bitcoin system. In: Paper presented at the 2019 IEEE 39th international conference on distributed computing systems (ICDCS) , 7–10 July 2019
45. Block file pruning. figshare https://github.com/bitcoin/bitcoin/blob/v0.11.0/doc/release-notes.md#block-file-pruning
46. Luu L, Narayanan V, Zheng C, Baweja K, Saxena P (2016) A secure sharding protocol for open blockchains. In: Paper presented at the 2016 ACM SIGSAC conference, 24–28 October 2016
47. Kokoris-kogias E, Jovanovic P, Gasser L, Gailly N, Syta E, Ford B (2016) OmniLedger: a secure, scale-out, decentralized ledger via sharding. In: Paper presented at the 2018 IEEE symposium on security and privacy (SP), 20–24 May 2018
48. Xu Z, Han S, Lei C (2018) CUB, a consensus unit-based storage scheme for blockchain system. In: Paper presented at the 2018 IEEE 34th international conference on data engineering (ICDE), 16–19 April 2018
49. Kokoris-kogias E, Jovanovic P, Gasser L, Gailly N, Syta E, Ford B (2016) RapidChain: scaling blockchain via full sharding. In: Paper presented at the 2018 ACM SIGSAC conference on computer and communications security, 15–19 October 2018
50. Pamies-Juarez L, Sanchez-Artigas M, Garcia-Lopez P, Mondejar R, Chaabouni R (2014) On the inter-play between data redundancy and retrieval times in p2p storage systems. Comput Netw 59(11):1–16
51. Segura SD, Perez-Sola C, Navarro-Arribas G, Herrera-Joancomarti J (2016) Analysis of the Bitcoin UTXO set. In: Paper presented at the 22nd international conference on financial cryptography and data security, 26 Feb 2018
52. Xu J (2011) (ed.): Research on data reliability in Peer-to-peer network storage system. Harbin: Harbin Engineering University
53. Borel E (1962) (ed.): Probabilities and life. Maurice Baudin Dover Publications, Inc., New York
54. Bitcoin-ETL developer. figshare https://github.com/blockchain-etl/bitcoin-etl
55. Ethereum Network Status. figshare https://ethstats.net

56. PeerSim. figshare http://peersim.sourceforge.net/