This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3179733, IEEE Transactions on Industrial Informatics

A Scalable Two-layer Blockchain System for Distributed Multi-cloud Storage in IIoT

Tianyi Xu, *Member, IEEE*, Tie Qiu, *Senior Member, IEEE*, Dengcheng Hu, Chaoxu Mu, *Senior Member, IEEE*, Zhiguo Wan, Wenyuan Liu

***Abstract*—Blockchain has been utilized to manage distributed multi-cloud storage in the Industrial Internet of Things(IIoT). Existing approaches commonly use trusted third-party servers or middle-wares to search data allocation strategies and use blockchain to enhance security. However, finding a fair data allocation strategy is hard when the third-party brokers are manipulated. Moreover, the complex computing in generating blocks reduces efficiency and heavy communication cost in consensus leads to critical challenges to scalability. To address that, this paper proposes a Scalable Two-layer blockchain System for storage in distributed Multi-cloud(STSM). We design a novel consensus mechanism called Proof-of-Storage-Allocation(PoSA), which integrates data placement problems into leader selection to achieve fair strategy and high QoS of data storage. We also incorporate asynchronous consensus groups into the consensus process to enhance scalability. Extensive experiments verify that STSM gains high scalability and increases efficiency while achieving high QoS in distributed multi-cloud data allocation.**

***Index Terms*—Multi-Cloud Storage, Blockchain System, Consensus Protocol, System Scalability, Industrial Internet of Things.**

I. Introduction

ORGANIZATIONS tend to apply cloud computing techniques into their business for data storing to reduce the overhead of local hardware in Industrial Internet of Things(IIoT)[1][2]. Numerous IT corporations are acting as Cloud Service Providers(CSP) to provide cloud storage services, such as Hadoop Distributed File System (HDFS), Google File System (GFS), and Windows Azure. In cloud storage, server availability is significant for data utilization. However, the failure probability of a single server in many cloud services is typically in the range of [0.01%, 10%][3]. This makes it insecure to store data in a single cloud server especially for industrial data. Thus researchers have tried to find solutions of multi-cloud storage in data storing, such as Filecoin and Storj[4].

The key to multi-cloud storage is to find a storage allocation strategy managing data placement in multiple clouds that fulfill storage requests. Third-party servers/brokers are usually used to manage the data storage process[5][6]. However, this kind of system is vulnerable to malicious corruption since centralized third-party can be manipulated.

With features of distributed architecture, global consensus, and smart contract, blockchain has the advantages of being self-organizing, open, auditable, and safe to constitute a trustworthy environment[7][8]. In blockchain-based systems, all nodes in the system maintain the ledger that keeps data. Information stored in the blockchain cannot be modified or deleted without the permission of the majority of nodes[9]. Therefore, in the light of these advantages, researchers have utilized blockchain-based system to enhance the reliability of cloud storage[10].

Existing works have employed blockchain-based multi-cloud storage solutions to handle large-scale IIoT data storage[11][12][13]. These works either use an off-chain solution to select a data storage strategy or use a trusted middleware to find a suited storage allocation strategy. Nodes in the system have consensus on a data allocation strategy selected by some preset computing service providers. However, the process of finding an optimal and fair data allocation strategy can not be ensured since the preset service providers can be tampered with, making the above models inappropriate for a totally untrusted environment, such as a data storage system consisting of heterogeneous and various storage service demanders and providers[14]. Moreover, blockchain-based systems suffer from heavy communication costs and massive resource consumption to achieve consensus[15][16][17], making it hard to be applied to resource-constrained large scale IIoT data storage. Therefore, for the task of managing multi-cloud storage through blockchain for IIoT, the following challenges need to be addressed:

- Designing a mechanism to select a balanced and fair storage allocation strategy for multi-cloud storage.
- Designing a scalable and secure blockchain system that can support consensus on the selected storage allocation strategy with limited resource consumption.

Regarding these challenges, we propose a Scalable Two-layer blockchain System for distributed Multi-cloud storage(STSM) for IIoT. In STSM, a novel consensus mechanism integrating the storage allocation problem is introduced to achieve fair data allocation strategies and high QoS of data

This work is supported by the Joint Funds of the National Natural Science Foundation of China (No.U2001204), the National Natural Science Foundation of China(No. 62022061, No. 92167206) and the Open Research Project of Zhejiang Lab (No.2021KF0AB02). (Corresponding authors: Tie Qiu, Wenyuan Liu)

Tianyi Xu, Tie Qiu and Dengcheng Hu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, and also with Tianjin Key Laboratory of Advanced Networking(TANK Lab), Tianjin 300350, China (e-mail: tianyi.xu@tju.edu.cn; qiutie@ieee.org; hudengcheng@tju.edu.cn).

Chaoxu Mu is with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: cxmu@tju.edu.cn).

Zhiguo Wan is with Zhejiang Lab, Hangzhou 311121, Zhejiang, China (e-mail: wanzhiguo@zhejianglab.com).

Wenyuan Liu is with the School of Information Science and Engineering, the Key Laboratory of Software Engineering of Hebei Province, YanShan University, Qinhuangdao 066004, China (e-mail: wyliu@ysu.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3179733, IEEE Transactions on Industrial Informatics

storage. Cloud service providers are divided into groups to form the first layer to conduct inner consensus. Then inner consensus results are validated in the second layer to prevent malicious attacks and increase scalability. We summarize the contribution below.

- We model the data placement problem as a multi-objective optimization problem that guarantees the integrity of data to be stored and enhances the QoS of the system. Moreover, we introduce an evolutionary algorithm solution to it.
- We propose a novel consensus algorithm incorporating the data placement problem into it. In this consensus process, all the participants compete for leadership by searching for a balanced storage allocation strategy. We incorporate dynamic time intervals in leader selection to achieve high QoS for storage allocation. By that, a fair and balanced data storage strategy for multi-cloud paradigms can be achieved.
- We propose a two-layer blockchain system for distributed storage. Asynchronous consensus group is incorporated into the system to enhance scalability. Blocks are generated and validated locally in the first layer in order to achieve high throughput. Then they will be globally validated in the second layer to ensure liveness and safety. By this design, the proposed system can achieve both scalability and security at the same time.

The rest of this paper is organized as follows. Related works are introduced in section II. We propose our architecture in Section III. In Section IV, the storage allocation for multi-cloud workflow management and the proposed consensus algorithm are built. The asynchronous two-layer blockchain architecture mechanism for efficiency enhancement is defined too. Section V discusses simulations and observations. Section VI is the conclusion.

II. Related Work

Liu et al.[5] considered machine failure as a key factor to influence data availability. They classified machine failure into correlated and non-correlated failure and designed an optimization problem regarding data availability, consistency maintenance cost, storage cost. Then nonlinear optimization is used to solve it. Yu et al.[18] considered the problem over three dimensions: selecting the erasure code, placement of encoded chunks, and optimizing scheduling policy, and solved it via the computation of a sequence of convex approximations with provable convergence. Though those researches can achieve balance on efficiency and cost, also provide data availability neglecting the bottleneck of hardware limitations, a trusted third party or platform is needed to organize the process, making them vulnerable to Single Point of Failure.

To eliminate the limitation of a trusted third party, blockchain-based architectures are proposed. Danish et al. introduced a neural network-based middleware designed for intelligent selection of storage technology for IIoT applications. A blockchain-based data placement protocol is proposed to ensure data integrity, traceability, auditability, and decision verifiability[4]. Paper [11] identified the problem of cloud storage as a multi-object optimization problem and utilized a genetic algorithm to solve it. Mishra et al. proposed a blockchain-based multi-cloud storage system to store EHR data[19]. This model outsourced EHR data to multiple clouds according to the opinions of doctors and utilized blockchain to ensure integrity and correctness. In the above models, finding a suitable data storage allocation strategy relies on certain preset roles. Due to that centralized process, those schemes suffer from single-point-of-failure and can not guarantee the reliability of storage allocation.

Moreover, to enhance the scalability of blockchain, several methods to improve the architecture of blockchains are proposed. Authors in [20] introduced a layered architecture combining public chains and consortium chains. Business federations such as cloud federation platforms construct consortium chains to provide services. Services are generated in consortium chains and flow to customers on public chains. Public chains functions as interfaces for service providers and service requirers. Though this approach can guarantee the safety of service, the process of generating services is not transparent. Fair and balanced service cannot be guaranteed. Authors in [21] proposed a partition-tolerant blockchain to target the IoT devices with limited power and limited network connectivity. In order to solve the problem of the low storage capability of IoT devices, the authors integrated a support blockchain in their architecture to store the IoT data.

Paper [22] proposed a multi-layer PBFT-based consensus mechanism to reduce communication complexity for PBFT-based consensus mechanism. Unfortunately, PBFT-based consensus is not suitable for permissionless public blockchains, making it hard to be deployed in dynamic and complex environments. In Ripple[23], instead of reaching a consensus with the majority of the system, each node only reaches a consensus with a subset of nodes. Authors in [24] introduced a layered sharding blockchain system. They developed a layered sharding consensus based on the collaboration among several shards to increase the scalability of blockchain-based systems. The sharding protocol can improve the throughput, but degrade the fault tolerance of the system. Xu et al. proposed a novel consensus protocol named PoC[25]. They optimized the target of PoW by collaboration credit, in that way nodes with higher credit could find the hash for the target easily. However, reducing the hardness of finding hash for PoC can lead to forking. Also, finding hash would waste too much computational power.

Considering the limitation of the works above, this paper focuses on:

1) ensuring fair and balanced allocation strategy in a multi-cloud environment in IIoT.
2) achieving consistency across multiple CSPs.
3) improving the efficiency and scalability of the system.

III. Proposed Blockchain Architecture

In this section, we introduce the architecture of the proposed system, which jointly considers blockchain technologies and cloud storage. The proposed design is shown in Fig.1. The system is mainly maintained by CSPs and Fog Nodes. CSP can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3179733, IEEE Transactions on Industrial Informatics

be divided into admin modules and storage modules. An admin module functions to manage one or several storage modules. The role of each entity in the proposed architecture is given below:

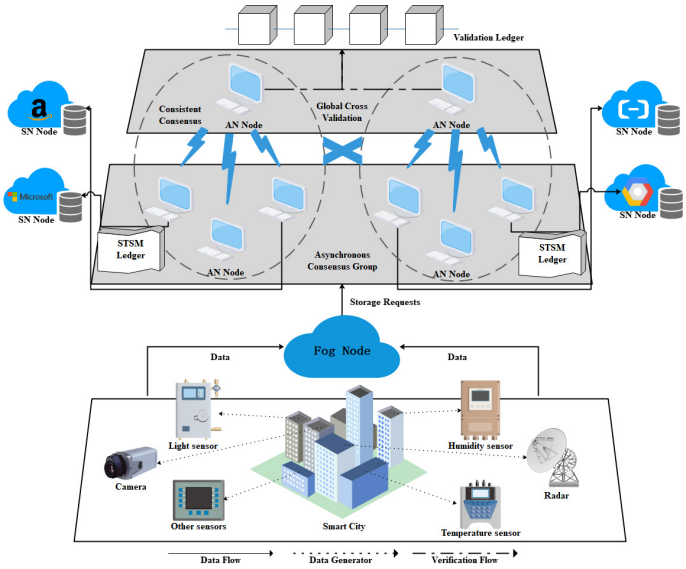


Fig. 1. Structure of STSM

1) *Fog Node*(FN): Fog nodes operate near terminal devices, to perform real-time computations such as gathering, filtering, and aggregating raw data generated by terminal devices in IIoT. Their responsibilities include: collecting and preprocessing the data generated by terminal devices and form storage requests, submitting the storage request to the blockchain system, and sending the data to cloud servers for storage according to the storage allocation strategy made by ANs.

2) *Admin Node*(AN): The admin module of each CSP plays the part of AN in our model. After fog nodes publish storage requests to the blockchain, ANs compute data placement strategy, a.k.a storage allocation. The strategy will be written on the blockchain to keep consistency. AN acts as full nodes of the blockchain system. ANs are separated into groups named Asynchronous Consensus Group(ACG) to increase scalability, which will be described in Section IV. ANs have three responsibilities: first, they update the parameters of the storage module of CSP to the blockchain. Second, they compete to solve the problem of storage allocation. Third, they conduct consensus to keep the consistency of storage allocation results on the blockchain.

3) *Storage Node*(SN): The storage modules of each CSP are represented as SN in the system. SNs act as light nodes in the blockchain system, i.e. they can extract and publish the information on the blockchain, but they don't have to keep the ledger. The responsibility of SN is to extract data placement decisions from blockchain and store data from fog nodes according to the storage allocation strategy.

The system is divided into two layers, the first layer contains paralleled asynchronous consensus groups to generate and validate blocks locally. Ledgers maintained inside the asynchronous consensus groups are called STSM Ledgers. The second layer consists of leaders of each asynchronous consensus group to organize cross-validation on blocks that are generated in the first layer. Validation information is stored in the second layer. The ledger in the second layer is called Validation Ledger.

Both STSM ledger and Validation Ledger are organized as blockchains. Participants can publish and extract information from both ledgers. In this way, transparency, traceability, auditability, and accountability are ensured.

IV. Proposed Consensus Algorithm and Blockchain Structure

A. Overview of the Workflow

A blockchain network is utilized to store the parameters of CSPs, the storage requests from fog nodes, and compute storage allocation strategies. In this way, the blockchain network functions as a storage and decision-making layer to provide traceability and management to CSPs and fog nodes. Storage requests and information of service providers are published to blockchains by smart contracts, as illustrated in other works[4][20]. Details are given below:

CSP Registration:

First, each CSP registers to the blockchain by getting a valid blockchain address and ID, namely,

$$CSP(AN, SN) \rightarrow Blockchain :$$

$$identity = \{AN_{ID}||AN_{address}||SN_{ID}||SN_{address}\}.$$

Blockchain based protocol:

1)All the ANs publish their SN's service parameter information on the blockchain network at time t. These service parameters include data availability, data transfer rate, and data storage price. The information updated at time $t+1$ will replace that at the time t. Therefore, ANs are able to use the up-to-date service parameters information to solve the problem of data placement.

$$AN \rightarrow Blockchain : input(SN) =$$

$$\{ID||data\ availability||transfer\ rate||price\}.$$

2)The fog nodes publish their storage requests between t and $t+1$, including data volume, data hash, timestamp etc, namely:

$$FN \rightarrow Blockchain : input(SR) =$$

$$\{ID||data\ volume||data\ hash||timestamp\}.$$

SR represents the storage request submitted by fog nodes.

3)To select the appropriate storage nodes to store data for service requirements published by fog nodes, ANs will use the up-to-date parameters of SN and storage requests together to compete for a storage allocation solution and start the consensus. We will formulate the storage allocation problem and the consensus process in the following subsections.

4) Once the competition and consensus are done, the storage allocation strategy is recorded and maintained in the blockchain. Then SNs will extract the storage allocation strategy from the blockchain and store data according to the strategy, namely,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3179733, IEEE Transactions on Industrial Informatics

$$AN \rightarrow Blockchain : input = \{ID_{SR}||A||timestamp\}$$
$$SN, FN \rightarrow Blockchain : retrieve = \{ID_{SR}||A||timestamp\}.$$
$$SN \rightarrow FN : retrieve = \{data\}.$$

A stands for storage allocation strategy.

B. Storage Allocation Problem

In distributed multi-cloud storage for large-scale IIoT, data redundancy is usually used to ensure data integrity. Data is commonly split into numerous blocks to be maintained. Considering the possibility of storage server failure, redundancy of raw data is conducted so that even some servers crash, other cloud servers can provide enough data blocks to restore the raw data. To conduct redundancy, data is commonly processed with erasure-coding to ensure data integrity[27].

Erasure coding is now a widely adopted redundancy technique, as an alternative to replication, for achieving low-cost durability guarantees in large-scale storage systems. Among many erasure coding constructions, Reed-Solomon (RS) code, a.k.a. RSCode, is the most popular erasure code deployed in production[28]. In our model, RSCode is adopted. In RSCode, raw data is split into m blocks in erasure-coding, and then $n-m$ encoded blocks are created, where n is the total number of blocks to maintain that data. (m, n) refers to the parameter of RSCode. The possibility that a data block can be retrieved depends on the possibility that the server functions well. To simplify the model, in this paper we assume that those n data blocks are stored in different SN, i.e. each SN only stores one block. Using any m blocks from those n ones, the original data can be restored.

Storage requests usually focus on storing data efficiently and ensuring a high data retrieve rate at a low cost. Finding storage allocation is often characterized as a multi-objective optimization problem(MOP). Inspired by [29], We use NSGA-II to find the optimal data storage allocation considering several variables of the cloud storage provider, including data availability, storage cost, and storage efficiency. Detailed steps are as follows.

1) Storage Allocation Coding: In the multi-cloud environment, suppose there are k SNs within set S_{SN} and $k > n$, and each SN can only store one data block of n, therefore a one-dimensional 0-1 array can be used to represent the data placement strategy A, as shown in Eq.1. If SN_i is chosen to store one data block, y_i should be set as 1.

$$A = \{y_1, y_2, ..., y_k\}, \sum_{i=1}^{k} y_i = n \tag{1}$$

2) MOP Definition a *Solving Process:* The MOP can be defined with expectat of maximum data availability da, maximum storage efficie se, and minimum storage cost sc, as shown in Eq.2.

$$MOP \begin{cases} \max\ da(A) \\ \max\ se(A) \\ \min\ sc(A) \end{cases} \tag{2}$$

The parameters related to MOP in SN can be expressed as a triple $SN = \{a_i, e_i, p_i\}$. a_i represents the data availability of SN_i, which means the probability that data can be successfully retrieved from SN. e_i is the data transfer rate and p_i is the storage price for each data unit. Regarding the feature of RSCode, to restore the original data, at least m data blocks should be retrieved. Thus the data availability of a strategy A can be represented as the possibility that at least m data blocks are available under that strategy. $da(A)$ is defined as follow:

$$da\,(A) = \sum_{k=m}^{n} \sum_{j=1}^{|S^A|} \left[\prod_{i \in S_j^A} a_i \prod_{i \in A \setminus S_j^A} (1 - a_i) \right] \tag{3}$$

S^A represents all the combinations of storage allocation strategies, by any of which the original data could be restored. $|S^A|$ is the number of combinations. S_j^A represents the jth combination in S^A. $A \setminus S_j^A$ represents the rest set that A excludes the nodes in S_j^A.

$se(A)$ depends on how long it takes to store the data blocks, as shown in Eq.4. $|f|$ represents the size of the data block.

$$se\,(A) = \max \left\{ y_i \times \frac{|f|}{e_i} | i = 1, 2, ...k \right\} \tag{4}$$

$sc(A)$ stands for the cost of all data blocks as Eq.5.

$$sc\,(A) = \sum_{i=1}^{k} (y_i \times (|f|) \times p_i) \tag{5}$$

By solving Eq.2 with NSGA-II, we can get a Pareto front of the optimal candidate data allocation strategies. The specific steps are as follows:

1) A parent population po_p is randomly generated with our coding scheme as Eq.1. According to po_p, an offspring po_o is produced according to *fast non-dominated sorting* and genetic operations with crossover and mutation.

2) After getting po_o, the *elite* selection strategy is carried out by $po_f = po_o + po_p$ using MOP proposed. To substitute the parent population, the best multiple individuals are chosen from po_f. This process produces new offspring po_o through crossover and mutation operations.

3) The population will continue to iterate according to operations 1) and 2). After the iteration is complete, we can obtain the final offspring as po_f.

The set of all results in po_f is called the Pareto front. We can further find the optimal solution from the Pareto front with entropy.

3) Optimal Storage Allocation Obtaining: We use *entropy* to find the optimal solution from the candidate data allocation strategies, i.e. the Pareto front achieved in the previous step. In the Pareto-optimal set, we define the QoS value of each strategy and select the strategy with the highest QoS. The main steps are as follows:

1) After getting the last population po_f from the result, we first normalize all fitness value and combine all value into a $N \times 3 (N = len(po_f))$ matrix as Eq.6. Each element can be noted as M_{ij}. j stands for the j_{th} candidate storage allocation strategy. The first element in a row stands for data

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3179733, IEEE Transactions on Industrial Informatics

availability of that strategy. Similarly, the second element stands for storage efficiency and the third element stands for storage cost.

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ \vdots & \vdots & \vdots \\ M_{N1} & M_{N2} & M_{N3} \end{bmatrix} \tag{6}$$

2) Then the proportion pr of the jth index of the ith element is calculated with Eq.7.

$$pr = \frac{M_{ij}}{\sum_{i=1}^{N} M_{ij}} \tag{7}$$

3) We calculate the entropy value of each index as Eq.8.

$$e = -k \sum_{i=1}^{N} p_{ij} ln(p_{ij}), k = \frac{1}{ln(N)} \tag{8}$$

4) Then the divergence of entropy d, the weight of each index w can be calculated according to Eq.9 and Eq.10.

$$d = 1 - e \tag{9}$$

$$w = \frac{d}{\sum_{j=1}^{3} d_j} \tag{10}$$

5) QoS value q_i can be calculated with weight w, as shown in Eq.11. The optimal solution is the candidate storage allocation strategy with maximum QoS, meaning that it can achieve balanced performance in data reliability, safety, and integrity.

$$q_i = \sum_{j=1}^{3} w_j pr_{ij} \tag{11}$$

C. Proof of Storage Allocation

The core of consensus is to select a node to generate blocks and make others agree with it. In our system, the goal of consensus is to select an AN to generate a block containing a data storage allocation strategy and make others verify it. Regarding that, we design the consensus protocol by integrating the optimization of storage allocation problems into a BFT-type consensus. We call it *Proof of Storage Allocation*, a.k.a. PoSA.

In PoSA, we use the process of finding the optimal storage allocation strategy as a way to find a leader to generate a new block. In normal proof-based consensus protocols such as PoW or PoS, the first proof-solved node is selected to generate blocks. In this way, all nodes compete to find the proof as fast as possible, which will lead to a massive waste of computational energy, let alone the proof itself is not meaningful to any practical service. Compared to them, in PoSA, finding an optimized storage allocation strategy is regarded as proof. The node that finds a strategy achieving the highest QoS will be chosen to generate a new block. Under this consideration. Two problems need to be solved:

a. Because of hardware difference and network delay, it takes unequal time for different nodes to find a solution for the proof, how can we find one with the highest QoS regardless of the computational time difference by different nodes and information transferring time?

b. Second, how can all nodes reach a consensus on the one with the highest QoS?

Focusing on the two problems raised above, we design the workflow of PoSA as shown in Fig.2. PoSA is divided into the phases of preparation, verification & vote, and blocking-out.

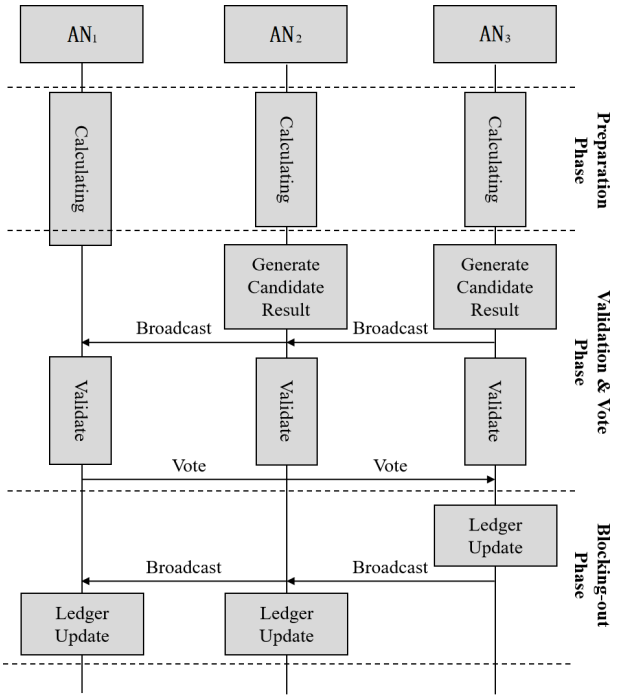


Fig. 2. Work chart of PoSA

Preparing Phase: PoSA is view-based. At a certain timestamp t, the set of storage requests published by fog nodes is defined as R' and is represented as a message $\langle R', t \rangle$. Then it will be encapsulated as $M_{re} = \langle Request, \langle R', t \rangle, d, v \rangle$ and broadcasted to S_{AN}, where d is the digest of M_{re}, v is a view count assigned to the M_{re}, S_{AN} is the set of ANs that participate in consensus.

After receiving M_{re}, each AN tries to find the optimal storage allocation strategy re_i based on the CSPs' parameters, and encapsulate the re_i as $M_{vo} = \langle Req - Vote, re_i, q_i, d, v \rangle$, where q_i denotes the QoS value corresponding to re_i.

Considering the first problem raised above, the solution of PoSA is to set a termination time T, we suggest a scheme to guarantee terminability. T can be obtained by Eq.12.

$$T_f = (1 + \alpha) T_{base} \tag{12}$$

where α denotes the offset rate, which is intended to adjust the consensus time to get enough storage allocation strategies. If more candidates are needed, α should be set larger and vice versa. Suppose the computational time spent on finding the storage allocation strategy for the first AN is T_f, and given

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3179733, IEEE Transactions on Industrial Informatics

the historical average time of storage allocation T_a. T_{base} can be calculated by Eq.13.

$$T_{base} = \begin{cases} T_a, \ if \ T_f < T_a \\ T_f, \ otherwise \end{cases} \tag{13}$$

Then each AN verifies M_{vo} generated by others during T, and stores them in a pool $\mathbb{V}$ called verification pool.

Validation & Vote Phase:

Considering the second problem raised above, we design the validation and vote phase as follows.

a) AN_i obtains the QoS value q from all M_{vo} in $\mathbb{V}$, and ranks all q to obtain the re_i corresponding to the maximum q as the optimal storage allocation strategy re_{best}.

b) AN_i encapsulates the M_{vo} and AN_{best}(the node which propose re_{best}) into $M_{co} = \langle Req - Commit, M_{re}, AN_{best}, AN_i, d, v\rangle$ and broadcasts it to $S_{AN} \setminus AN_i$, indicating that the AN_i supports AN_{best} as the consensus leader in this round.

c) Each AN_i verifies the M_{co} got. If AN_i learns that it is supported by more than half of ANs, it sends a message to $S_{AN} \setminus AN_i$ to become the new leader L_{cur}.

Blocking-out Phase:

L_{cur} encapsulates the transactions based on re_{best} and the metadata of R'. Then L_{cur} generates a new block B_{new}, and synchronizes it with $S_{AN} \setminus L_{cur}$ according to PBFT. When the consensus round is completed, the block is updated to a local STSM ledger. Furthermore, L_{cur} must transmit the B_{new} to the verification layer for cross-validation. Algorithm.1 shows the PoSA consensus pseudo-code.

Algorithm 1 PoSA Consensus

Require: Request set R', Finish time T
Ensure: Blockchain ledger update

while T **do** parallel
  $\mathbb{V} \leftarrow \{\}$
  for Each AN_i **do** parallel
    Storage Allocation using NSGA-II
    Generate candidate message M_{vo}^i
    $Broadcast(M_{vo}^i)$
    $\mathbb{V} \leftarrow M_{vo}^i$
  end for
end while
for Each AN_i **do** parallel
  Verify all M_{vo} in $\mathbb{V}$ and select optimal result re_{best}
end for
if Result re_{best} gets more than half votes **then**
  L_{cur} = Generator of re_{best}
end if
L_{cur} broadcast new block B_{new}
for Each AN_i **do** parallel
  validate and update ledger
end for

D. Asynchronous Consensus Groups

Basic Concept: In PoSA, the preparation phase needs all ANs to compute for the same requests, which is time-consuming and will lead to a low-efficiency problem with the piling of requests. We introduce the idea of *asynchronous consensus groups* inspired by Wang[26], and design a two-layer architecture to increase the effectiveness of storage request processing.

The first layer consists of all ANs, which are divided into several groups considering their properties such as geographical location, service capability, etc. PoSA is carried out within each group to handle storage requests. However, malicious nodes may launch attacks by controlling majority nodes inside one consensus group. To avoid that, we propose a global validation mechanism.

Global Validation Mechanism: After local consensus, ACG generates block B_{g_i}, where g_i represents the ACG number. After updating the local ledger, ACG broadcasts B_{g_i} to the second layer. The second layer is maintained by historical and current honest leaders from all ACGs. The nodes in the second layer are called AN^2.

After B_{g_i} is submitted to the second layer by L_{cur}, L_{cur} chooses unverified blocks in the second layer and checks whether the selected strategy is the best among all the candidates. Then L_{cur} would start a consensus to maintain the result into the validation layer. We use PBFT as the consensus protocol of the validation ledger.

As can be seen that this method is an asynchronous verification, so the removal of created blocks in the local STSM ledger is impossible. Thus the verification result would have an influence on leader selection in latter epochs and would trigger an incentive (penalty or reward) on the related ACG. There are many incentive methods for blockchain-based systems that can work for our model. We don't discuss them in detail since they are out of the scope of this paper.

E. System Analysis

Liveness and Safety Analysis: The blocks are validated locally and globally in and across layers. The consensus is PBFT-based both inside a local asynchronous consensus group and across groups. The safety and liveness are retained across groups with modified view change protocol.

The liveness of the system depends on PoSA and PBFT. In PoSA, when a storage request is published to the blockchain, ANs in the ACG will start a competition for candidate storage allocation strategies to design a strategy for its own benefit, and all the candidates are validated by the group in a PBFT-based consensus manner. After the candidate with the highest QoS is chosen, the leader is selected and the block is generated and verified. By this, the liveness of PoSA is guaranteed. The modified view change protocol replaces faulty group leader. Thus, liveness is guaranteed across groups. The consensus in subgroups is the precondition to invoke protocol in verification layers. Thus, safety is guaranteed across layers. This consistency with PBFT in safety and liveness within and across groups leads to consistency for the whole network.

However, there are some limitations to this model. Because the consensus is two-stage and asynchronous, when an AN is tampered with in an epoch, the global validation can not modify the corrupted strategy selected inside the AN, resulting in the corrupted storage strategy would infect the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3179733, IEEE Transactions on Industrial Informatics

storage process in one epoch. However, after global validation, tampered AN would be punished, and the following epoch would be reliable again.

Security Analysis: Common threats for blockchain-driven systems include data tampering, Sybil attacks. Also, PoSA is similar to crowdsourcing in some ways, therefore a kind of attack named "free-riding" should be considered.

For data tampering, the data model of blockchain is cryptographically embedded. At the same time, peer-to-peer messages are cryptographically secured and the transaction authentication stage is also tamper-proof.

A malicious node can launch a Sybil attack by forging multiple identities and pretending to be multiple nodes in the consensus. PoSA is secure from Sybil attacks. The analysis is as follows:

a) All nodes participating in PoSA shall have a certain computing power to generate an effective data storage allocation strategy candidate. In this way, it would be hard for malicious nodes to forge multiple identities at the same time since that will lead to its computer power dispersion.

b) Even if malicious nodes can forge identities to tamper voting phase, the result of STSM ledgers would be verified in the validation layer. In this way, even an ACG has been tampered with, malicious nodes could be detected in global validation and get punished.

A similar security threat to crowdsourcing is that since each AN is required to contribute its own result of the storage allocation problem, there may be lazy workers generating the results randomly. They are called free-riders and the action is called "free-riding". When no effective results are generated, there are chances that lazy workers are voted as the leader. However, it's reasonable to assume that every AN is selfish, so ANs tend to generate storage allocation candidates that match their own benefit, such as allocating them to store data requiring low availability. It is easy for those candidates to overwhelm the random ones in validating process even if they are selfish. Considering this incentive, AN would rather generate selfish candidates rather than random ones. Moreover, even when all the ANs generate selfish candidates, the one with the highest QoS value is relatively fair to others. In addition, T_{base} can be set dynamically to ensure enough effective candidates are generated and that will decrease the willingness of submitting random proposals. Based on this analysis, this system is secure to "free-riding".

Scalability Analysis: As the number of ANs increases, the communication overhead in a large-scale distributed network increases. After introducing the concept of ACG, S_{AN} is divided, the consensus is carried out within ACG. It does not need the participation of other groups and reduces the network communication overhead. Theoretically, the ideal communication pressure will be reduced to $\frac{1}{Count(ACG)}$, where $Count(ACG)$ is the number of ACG. As the number of nodes increases, the number of ACGs can be adjusted continuously. In this way, the system scalability is ensured and enhanced.

V. Experiments

A. Simulation Setup

The simulation experiment is conducted on a desktop PC with an Intel Core i7-7700 at 3.6 GHz and 8 GB RAM. The properties of CSP, which are data availability, unit transfer rate, are adopted from cases in *Cloud Harmony*[1]. The price of each CSP is referred from its official website(Microsoft, Amazon, Alibaba, etc.). And the computing power is divided according to the proportion. The bandwidth of each CSP is set as 1Mb/s. To simulate storage requests in IIoT, we generated storage requests ranging from 0.5GB to 2GB containing heterogeneous data(audio, video, picture, etc.) randomly in a certain time interval. For RSCode, (5,7) is chosen as the code parameter based on previous research. We develop a blockchain system based on Ethereum to implement STSM. Table.I shows the parameters related to the simulation experiments.

TABLE I
Parameters realted to the simulation experiments

Parameter description	Value
Number of CSPs	10, 13, 16, 19, 22
High computing power(%)	30%
Middle computing power(%)	40%
Low computing power(%)	30%
Population Size	100
Crossover and Rate	Uniform with 0.2
Mutation and Rate	Bit-flip with 0.05
Iterations Number	200

B. Evaluation Indicators

We verify PoSA from three perspectives: resource utilization, the efficiency of consensus, and allocation results. The resource utilization takes the CPU utilization during consensus execution as a reference. Time consumption is an observation of consensus efficiency. The allocation result can be verified from the QoS value.

In addition, the most common metric for blockchain efficiency is the speed of processing transactions. This paper uses Transaction Throughputs (TPS) to imply the performance of consensus. In our simulation, TPS is derivatively defined as the maximum number of storage requests that can be processed per second in order to further demonstrate the efficiency of the STSM.

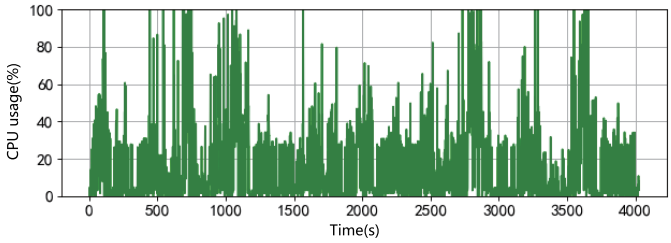
C. Experimental Results and Analysis

Fig.3 shows the observation of CPU usage in a certain time range. It can be seen that the peak value of CPU usage running PoW exceeds 100% multiple times and during the execution of consensus, the average CPU utilization is about 40%. However, the peak value of CPU usage running PoSA is only about 90% and the average CPU utilization is about 30%.

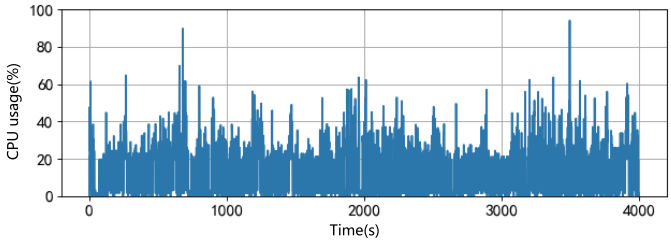
The calculation process refers to the phase of solving the problem of data storage strategy using NSGA-II, and

[1]https://cloudharmony.com

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3179733, IEEE Transactions on Industrial Informatics

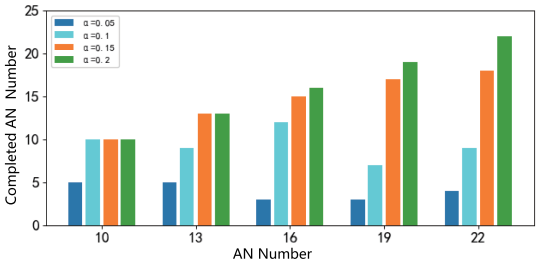


(a) PoW

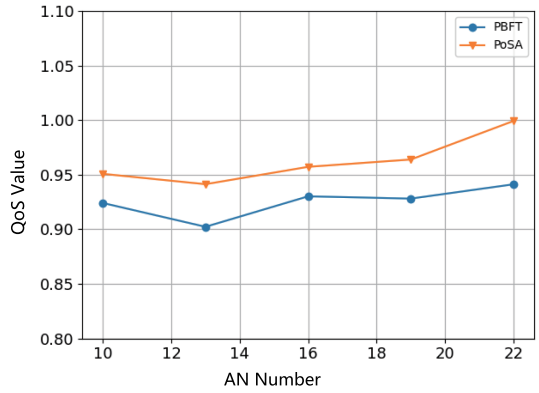


(b) PoSA

Fig. 3. CPU utilization of PoSA and PoW



(a) Completed AN number



(b) QoS value

Fig. 4. The results of completed AN numbers and QoS when α varies

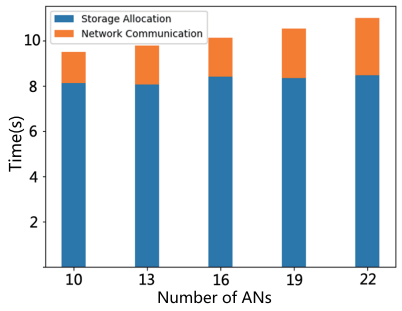


Fig. 5. Time for storage allocation and network communication

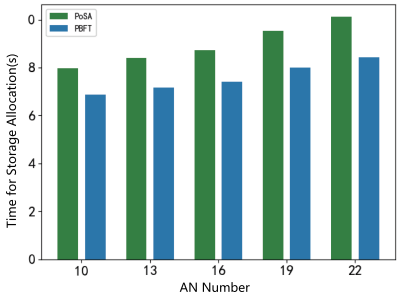
transmitting the results to the entire network. According to Eq.12, α is an important value that has a great influence on time-consuming of the calculation process. Thus we make an observation on different values of α. In our simulation, the number of candidate allocation strategies is not altered in different epochs. We have tested in previous work that the time of each epoch that the same number of candidates are generated differ from each other for less than 20%, so we set alpha to range from 0.05 to 0.2. We modify α from 0.05-0.2 and the value interval is set as 0.05. We evaluate α from two viewpoints shown in Fig.4: the number of nodes that finish the calculation process and the QoS value.

As can be seen from Fig.4, with the increase of α, the number of nodes that are able to finish the calculation process increases, and the same goes with the corresponding QoS value, until α reach 0.15. After that, even the number of nodes that can finish the calculation process keeps rising, the QoS value remains almost the same, which indicates that the system can achieve its best performance when α is set as 0.15.

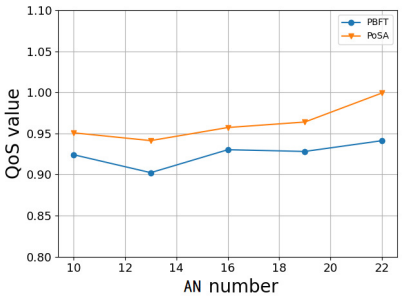
In PoSA, the time used for consensus contains two parts: the time for generating storage allocation strategy candidates, i.e. time for storage allocation, and the time for validating the candidates to select the leader, i.e. time for network communication. We make an observation on the two parts in the condition of the number of AN varies. The result is shown in Fig.5. As can be seen, the time for storage allocation changes little when the number of AN increases, however, the time for validating candidates to select the leader increases. The reason is that the time for storage allocation mainly depends on the number of required candidates in one epoch and that merely changes in our experiment. The process of selecting leaders costs lot of network communications due to the usage of PBFT. It can be seen that the performance of PoSA depends on the size of ACG and the bottleneck is the communication cost of PBFT consensus.

In many blockchain-based models, PBFT is usually chosen as the method of leader selection in consensus mechanism to achieve eventual consistency[20]. We compare PoSA with the method that uses PBFT to find the leader and make the whole network have a consensus on the storage allocation strategy generated by that leader. We compare the time for storage allocation and QoS value when AN varies. Fig.6 displays the experimental results. The time cost of PoSA is larger than the PBFT-based leader selection paradigm. That is because all participants calculate for storage allocation policy candidates in PoSA and only the leader selected calculates that in the PBFT-based leader selection paradigm. However, for QoS value, as the number of An increases, the QoS value of PoSA becomes higher than the baseline. This is because PoSA introduces storage allocation into the consensus and uses the result as an effective indicator to select a leader. In this way,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3179733, IEEE Transactions on Industrial Informatics



(a) Time for storage allocation



(b) Results of QoS value gained

Fig. 6. Comparison of PoSA and PBFT-based leader selection paradigm

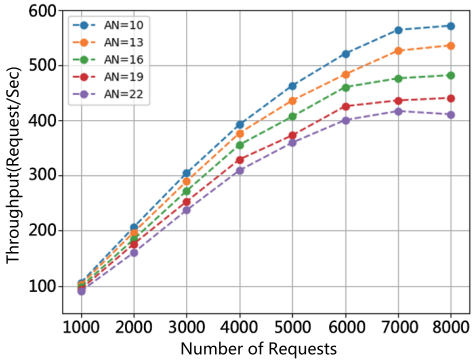


Fig. 7. TPS for different number of ANs with different number of requests in one block

a balanced and fair storage allocation strategy that has higher QoS value can be generated and the node that generates the corresponding strategy can be selected as the leader.

TPS is observed under the premise of different AN numbers and different block sizes. The maximum capacity to handle storage requests under the preset simulation environment is estimated. We start the number of storage requests from 1000 and increase it by 1000 each time in our simulation. The experimental results are shown in Fig.7. As we can see, since leader selection is PBFT-based in ACG, and the communication complexity is $O(n^2)$, thus as the number of nodes inside ACG grows, the leader selection will take additional time and results in a decrease in TPS. Besides, when the storage request number gets bigger (e.g. more than 7000 per epoch in this simulation), nodes in ACG will take more time in finding optimal storage allocation, and the system comes to a bottleneck in throughput due to communication complexity.

Through experiments, it can be seen that the PoSA consensus algorithm proposed in this paper has certain advantages in resource utilization compared with PoW and can obtain a higher QoS value compared with PBFT. With the improvement of the hardware, it can have better storage request processing capacity.

VI. Conclusion

In this paper, we propose a scalable two-layer blockchain system for distributed multi-cloud storage in IIoT. First, we build a mathematical model of MOP for data storage allocation and solve it with NSGA-II. Then we introduce a novel consensus method that integrates the MOP problem into leader selection to guarantee that fair and balanced storage allocation strategies to be selected. We build a two-layer blockchain architecture in which the asynchronous consensus group is proposed to increase efficiency and ensure the reliability of the model. In addition, the design of asynchronous consensus group makes the model highly scalable. Through theoretical analysis, we demonstrate the proposed architecture is secure and efficient. Through simulations and experiments, it can be seen that the proposed architecture can greatly increase the QoS and scalability of blockchain-based multi-cloud storage for IIoT.

References

[1] M. Shen, J. Duan, L. Zhu, J. Zhang, X. Du and M. Guizani, "Blockchain-Based Incentives for Secure and Collaborative Data Sharing in Multiple Clouds," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1229-1241, 2020.

[2] Z. Ning, S. Sun, X. Wang, L. Guo, S. Guo, X. Hu, B. Hu and R. Kwok, "Blockchain-enabled Intelligent Transportation Systems: A Distributed Crowdsensing Framework," *IEEE Transactions on Mobile Computing*, doi:10.1109/TMC.2021.3079984, 2021.

[3] D. Poola, S. K. Garg, R. Buyya, Y. Yang and K. Ramamohanarao, "Robust Scheduling of Scientific Workflows with Deadline and Budget Constraints in Clouds," in *Proceedings of 2014 IEEE 28th international conference on advanced information networking and applications*, 2014, pp. 858-865.

[4] S. M. Danish, K. Zhang and H. Jacobsen, "BlockAIM: A Neural Network-Based Intelligent Middleware For Large-Scale IoT Data Placement Decisions," *IEEE Transactions on Mobile Computing*, doi: 10.1109/TMC.2021.3071576, 2021.

[5] J. Liu, H. Shen, H. Chi, H. Narman, Y. Yang, L. Cheng and W. Chung, "A Low-Cost Multi-Failure Resilient Replication Scheme for High-Data Availability in Cloud Storage," *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1436-1451, 2020.

[6] N. Chen, T. Qiu, Z. Lu, and D. O. Wu, "An adaptive robustness evolution algorithm with self-competition and Its 3D deployment for Internet of Things," *IEEE/ACM Transactions on Networking*, vol. 30, no. 1, pp. 368-381, 2022.

[7] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan and M. Rajarajan, "Blockchain at the Edge: Performance of Resource-Constrained IoT Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 174-183, 2021.

[8] Y. Wu, H. Dai, H. Wang, and K.R Choo "Blockchain-Based Privacy Preservation for 5G-Enabled Drone Communications," *IEEE Network*, vol. 35, no. 1, pp. 50-56, 2021.

[9] J. Wang, X. Feng, T. Xu, H. Ning and T. Qiu, "Blockchain-Based Model for Nondeterministic Crowdsensing Strategy With Vehicular Team Cooperation," *IEEE Internet Things Journal*, vol. 7, no. 9, pp. 8090-8098, 2020.

[10] H. Zhou, X. Ouyang, Z. Ren, J. Su, C. de Laat and Z. Zhao, "A Blockchain based Witness Model for Trustworthy Cloud Service Level Agreement Enforcement," in *Proceedings of IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 1567-1575.

[11] M. Xu, G. Feng, Y. Ren and X. Zhang, "On Cloud Storage Optimization of Blockchain With a Clustering-Based Genetic Algorithm," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8547-8558, 2020.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3179733, IEEE Transactions on Industrial Informatics

[12] J. Ricci, I. Baggili and F. Breitinger, "Blockchain-Based Distributed Cloud Storage Digital Forensics: Where's the Beef?," *IEEE Security & Privacy*, vol. 17, no. 1, pp. 34-42, 2019
[13] Y. Wu, H. Dai, and H. Wang "Convergence of Blockchain and Edge Computing for Secure and Scalable IIoT Critical Infrastructures in Industry 4.0," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2300-2317, 2021.
[14] T. Qiu, J. Liu, W. Si and D. O. Wu, "Robustness Optimization Scheme with Multi-population Co-evolution for Scale-free Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1028-1042, 2019.
[15] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," White Paper, 2008. [Online]. Available: http://www.bitcoin.org/ bitcoin.pdf
[16] P. Vasin, "Blackcoin's proof-of-stake protocol v2," White Paper, 2014. [Online]. Available: https://blackcoin.co/blackcoin-pos- protocol-v2-whitepaper.pdf
[17] M. Vukolić, J. Camenisch, and D. Kesdoğan, "The quest for scalable blockchain fabric: Proof-of- work vs. BFT replication," in *Proceedings of International workshop on open problems in network security*, 2015, pp. 112–125.
[18] Y. Xiang, T. Lan, V. Aggarwal and Y. R. Chen, "Joint Latency and Cost Optimization for Erasure-Coded Data Center Storage," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2443-2457, 2016.
[19] R. Mishra, D. Ramesh and D.R. Edla, "Deletable Blockchain based Secure EHR Storage Scheme in Multi-Cloud Environment," in *Proceedings of 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems*, pp. 1057-1064, 2020.
[20] B.C. Ghosh, T. Bhartia, S.K. Addya, and S. Chakraborty, "Leveraging Public-Private Blockchain Interoperability for Closed Consortium Interfacing," in *Proceedings of IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1-10 .
[21] K. Karlsson, W. Jiang, S. Wicker, D. Adams, E. Ma, R. van Renesse and H. Weatherspoon, "Vegvisir: A Partition-Tolerant Blockchain for the Internet-of-Things," in *Proceedings of 2018 IEEE 38th International Conference on Distributed Computing System*, 2018, pp. 1150-1158.
[22] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao and M. A. Imran, "A Scalable Multi-Layer PBFT Consensus for Blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146-1160, 2021.
[23] D. Schwartz, Y. Noah and B. Arthur, "The ripple protocol consensus algorithm," Ripple Labs Inc White Paper vol. 5, no. 8, pp. 151, 2014.
[24] Z. Hong, G. Song, L. Peng and W. Chen, "Pyramid: A layered sharding blockchain system," in *Proceedings of IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1-10.
[25] C. Xu, K. Wang, P. Li, S. Guo, and J. Luo, B. Ye and M. Guo, "Making Big Data Open in Edges: A Resource-Efficient Blockchain-Based Approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 4, pp. 870-882, 2019
[26] J. Wang and H. Wang. "Monoxide: Scale out blockchains with asynchronous consensus zones," in *Proceedings of 16th USENIX Symposium on Networked Systems Design and Implementation*, 2019, pp. 95-112.
[27] M. Su, L. Zhang, Y. Wu, K. Chen and K. Li, "Systematic Data Placement Optimization in Multi-Cloud Storage for Complex Requirements," *IEEE Transactions on Computers*, vol. 65, no. 6, pp. 1964-1977, 2016.
[28] Q. Yao, Y. Hu, L. Cheng, P. P. C. Lee, D. Feng, W. Wang, W. Chen, "StripeMerge: Efficient Wide-Stripe Generation for Large-Scale Erasure-Coded Storage," in *Proceedings of 2021 IEEE 41st International Conference on Distributed Computing Systems*, 2021, pp. 483-493.
[29] X. Xu, S. Fu, W. Li, F. Dai, H. Gao and V. Chang, "Multi-Objective Data Placement for Workflow Management in Cloud Infrastructure Using NSGA-II," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 605-615, 2020.

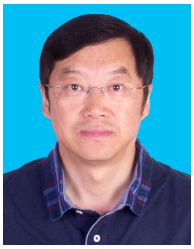
Tianyi Xu (M'22)is currently an engineer at the College of Intelligence and Computing, Tianjin University, China. His research interest includes Internet of Things and Blockchain and data mining. He has dozens of papers in international journals and conference proceedings, such as IEEE Internet of Things Journal, Future Generation Computer Systems, International Journal of Distributed Sensor Networks etc.

Dr. Tie Qiu (M'12-SM'16) is currently a Full Professor at School of Computer Science and Technology, Tianjin University, China. Prior to this position, he held assistant professor in 2008 and associate professor in 2013 at School of Software, Dalian University of Technology. He was a visiting professor at department of electrical and computer engineering of Iowa State University in U.S. (2014-2015). He serves as an associate editor of IEEE/ACM Transactions on Networking (ToN), IEEE Transactions on Network Science and Engineering (TNSE) and IEEE Transactions on Systems, Man, and Cybernetics: Systems, area editor of Ad Hoc Networks (Elsevier), associate editor of Computers and Electrical Engineering (Elsevier), Human-centric Computing and Information Sciences (Springer), a guest editor of Future Generation Computer Systems. He serves as General Chair, Program Chair, Workshop Chair, Publicity Chair, Publication Chair or TPC Member of a number of international conferences. He has authored/co-authored 10 books, over 200 scientific papers in international journals and conference proceedings, such as IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Industrial Informatics, IEEE Communications Surveys & Tutorials, IEEE Communications, INFOCOM, GLOBECOM etc. There are 18 papers listed as ESI highly cited papers. He has contributed to the development of 6 copyrighted software systems and invented 20 patents. He is a distinguished member of China Computer Federation (CCF) and a Senior Member of IEEE and ACM.

Dengcheng Hu received his master's degree in computer technology from Tianjin University in 2021. He is currently pursuing a Ph.D. in Computer Science and Technology at Tianjin University. His current research interests include blockchain technology, including consensus mechanisms, sharding technologies, and blockchain applications. During his Ph.D. studies, he participated in the development of applications such as blockchain-based voting systems.

Chaoxu Mu (M'15–SM'18) received the Ph.D. degree in control science and engineering from the School of Automation, Southeast University, Nanjing, China, in 2012. She was a visiting Ph.D. student with the Royal Melbourne Institute of Technology University, Melbourne, VIC, Australia, from 2010 to 2011. She was a Post-Doctoral Fellow with the Department of Electrical, Computer and Biomedical Engineering, The University of Rhode Island, Kingston, RI, USA, from 2014 to 2016. She is currently a Professor with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. Her current research interests include nonlinear system control and optimization, adaptive and learning systems.

Zhiguo Wan is a principal investigator in the Zhejiang Lab, Hangzhou, Zhejiang, China. His main research interests include security and privacy for cloud computing, Internet-of-Things and blockchain. He received his B.S. degree in computer science from Tsinghua University in 2002, and Ph.D. degree in information security from National University of Singapore in 2007. He was a postdoc in Katholieke University of Leuven, Belgium and an assistant professor in the School of Software, Tsinghua University, Beijing, China.



Wenyuan Liu received the B.S. and M.S. degrees in computer science at Northeast Heavy Machinery Institute, China, and the Ph.D. degree in computer science at the Harbin Institute of Technology in 2000. Since 1996, he has been with the School of Information Science and Engineering at the Yanshan University, Qinhuangdao City, China, where he is currently a professor. His research interests include wireless sensor networks and mobile networks.