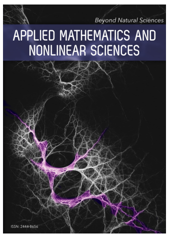

Applied Mathematics and Nonlinear Sciences

https://www.sciendo.com

The construction of blockchain platform for engineering museums based on vectorized image processing technology

Shuiting Du[1,†], Hao Yuan[1], Ruiqi Li[1], Jinxiu Zhang[1]

1. 1. Digital Division of State Grid Gansu Electric Power Company, Lanzhou, 730050, China

Submission Info

Communicated by Juan Luis García Guirao
Received May 22, 2022
Accepted October 19, 2022
Available online April 27, 2023

Abstract

The characteristics of blockchain such as tamper-evident and distributed can effectively solve the increasingly prominent security and privacy issues in engineering pavilions, however, the current throughput of mainstream blockchain platforms is far from meeting the demand for rapid chain-up of massive data in engineering pavilions, and the high redundancy storage mechanism adopted by traditional blockchain cannot be applied to engineering pavilion scenarios. To address the above problems, based on vectorized image processing technology, a hierarchical blockchain architecture is first proposed to store block data in multiple distributed cloud servers and edge servers in a hierarchical manner to cope with the growing data volume in the industrial Internet. Then based on the topology between edge servers, a blockchain network slicing algorithm is designed based on an improved complex network association partitioning algorithm, which improves the blockchain network throughput while shortening the slicing time. Finally, the impact of block broadcast time on throughput is demonstrated by formalizing the block broadcast process as a spanning tree, based on which a spanning tree-based intra-slice master node selection algorithm is proposed to further improve the throughput of the blockchain network. The analysis and experimental results show that the proposed scheme effectively reduces the partitioning time of the blockchain network and the broadcast time of the blocks within each partition. Compared with the classical association partitioning algorithm, this paper's partitioning algorithm can reduce the partitioning time by about 36% without sacrificing the quality of partitioning when partitioning a large-scale network.

Keywords: image restoration; vectorized image processing; blockchain; dynamic grouping; slicing
AMS 2020 codes: 68U05

†Corresponding author.
Email address: 18093197079@163.com

ISSN 2444-8656
https://doi.org/10.2478/amns.2023.1.00176

sciendo OPEN ACCESS © 2023 Shuiting Du, Hao Yuan, Ruiqi Li, and Jinxiu Zhang1, published by Sciendo.
This work is licensed under the Creative Commons Attribution alone 4.0 License.

1 Introduction

The engineering hall, an important application scenario for 5G, is driving industrial production efficiency and productivity to unprecedented levels [1]. Currently, the Industrial Internet is widely used in different commercial and industrial domains such as smart grid, e-commerce, energy control and efficient logistics [2]. However, the centralized architecture used in industrial Internet systems raises security and privacy concerns and may have a single point of failure to provide stable services [3]. These issues are becoming increasingly prominent as the number of devices connected to the industrial Internet continues to grow.

Blockchain is a distributed shared ledger, an irreversible public database that enables unrelated participants to reach consensus on the occurrence of a specific transaction or event without the need for centralized authorization. The characteristics of blockchain, such as tamper-proof, decentralized and traceable, can effectively solve the security and privacy issues in the industrial Internet. However, the combination of the two still faces many challenges, which are summarized in the following two aspects.

Insufficient throughput: The large amount of data collected and generated by industrial Internet devices needs to be stored securely, efficiently and in real time. However, the current throughput (Transactions Per Second, TPS) of mainstream blockchain platforms is far from meeting the demand for rapid on-chain storage of massive data in the industrial Internet [4-5].

High storage pressure: Blockchain applications in different scenarios are enhanced by its high redundancy storage mechanism (each node stores a complete copy of the ledger) which enhances the openness and transparency of data and ensures that the data is not tampered with. However, on the other hand, this will bring huge storage pressure to the blockchain, and the high redundancy storage mechanism adopted by traditional blockchain cannot be applied to the industrial Internet scenario.

The sharding technology, which was initially applied in the database field, is the most direct and effective means to improve the throughput of blockchain [6-7]. The application of sharding technology to blockchain is to split the original blockchain network into several small-scale blockchain networks, each of which consists of a part of the original network, called "sharding". Transactions in the entire network are distributed to different slices for parallel processing, thus increasing the throughput of the blockchain in an approximately linear manner [8-9]. In the literature [10], a public chain-based sharding scheme ELASTICO is proposed.In each consensus cycle of ELASTICO, participants are required to compute a Proof of Work (PoW) answer, which is used to configure the shards. Each slice uses the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm to verify transactions, and the consensus results are submitted to the final slice, which is responsible for generating the final decision on the consensus results of the other slice. Finally, the decision result will be returned to update the other slices. However, ELASTICO needs to reconfigure the slices after each consensus round, and any slice needs to store the block data of all other slices in the network, which causes a waste of computation and storage resources. To solve the problems of ELASTICO, a sharding protocol called OmniLedger was proposed in the literature [11]. It uses a distributed random number generation scheme and verifiable random functions to configure the sharding, which reduces the computational overhead of the sharding process. However, OmniLedger needs to broadcast to the whole network when processing cross-slice transactions and has the same fault tolerance as ELASTICO, which is only 1/4. Based on this, literature [12] proposes a slicing scheme RapidChain that improves the fault tolerance to 1/3. Meanwhile, to solve the problem that OmniLedger needs to broadcast to the whole network when processing cross-slice transactions. RapidChain designs an inter-slice routing protocol to quickly verify cross-slice transactions and reduce communication overhead. However, RapidChain is designed based on the assumption of network synchronization and its performance in asynchronous networks has not been verified. The literature [13] proposes Monoxide, a horizontally scalable slicing protocol, by designing a specific asynchronous consensus region so that the throughput can increase linearly with the number of consensus regions and does not degrade the decentralization of the system. In addition, Monoxide designs a PoW scheme for amplifying the arithmetic power so that the effective arithmetic power of each region remains the same as that of the whole network, thus guaranteeing the security of each slice.

While most of the existing slicing protocols are built on the basis of public chains, the slicing protocols of federated chains have been rarely explored. Since public chains allow any node to join and the block data is completely public, it is necessary to increase the cost of node mischief with a large number of complex calculations to improve the security of the network when slicing public chains. However, nodes in a federated chain are authenticated by a Certificate Authority (CA) and join the network, and they usually only fail to participate in the consensus process as expected due to downtime, network latency, etc. [14]. Thanks to the closed nature of the federated chain network, the literature [15] proposes a federated chain sharding protocol MDIoTSP that does not require complex computations to secure the network, which is able to shorten the block generation cycle while maintaining the same throughput as ELASTICO. However, its sharding configuration process only considers the geographical location of nodes and lacks a sharding reconfiguration process, which makes the network may not continue to work properly after long-term operation due to the failure of some nodes and reduces the robustness of the system.

Although the aforementioned literatures have addressed the performance bottleneck of blockchain to some extent, none of them is designed for industrial Internet scenarios, and they fail to consider the problem of insufficient blockchain capacity. Therefore, there is an urgent need to design a new blockchain architecture to cope with the huge amount of data from the industrial Internet.

To address the above needs, this paper proposes a Hierarchical Sharding Blockchain (HSChain) based on vectorized image processing technology. The key idea is to divide the blockchain network into multiple shards based on the topology between nodes, and to select the master node for each shard that minimizes the block broadcast time. Each slice runs the PBFT consensus algorithm on a set of disjoint transactions for verification. The successfully verified transaction blocks of each slice within a consensus cycle (epoch) are packaged by the master node into a compressed block, which contains a "pointer" to these transaction blocks. Each edge layer slice periodically offloads the transaction blocks to the cloud blockchain layer for storage, and only the smaller compressed blocks are stored locally.

2 Vectorized image processing techniques

2.1 Remote sensing image restoration processing

Following the imaging principle of remote sensing images and the general steps of image restoration processing, a mathematical model of remote sensing image degradation is established, which needs to reflect the causes of remote sensing image degradation and represent them in the form of functions to derive the degradation function of remote sensing images [16-18]. In this paper, the initial remote sensing image is denoted by $f(x,y)$, where x and y are the pixel values in the horizontal and vertical directions of the initial image, respectively. In addition, the degradation function of the remote sensing image is $h(x,y)$, and the degraded remote sensing image can be obtained by convolving the degradation function $h(x,y)$ with the initial remote sensing image [19]. An additive noise term is introduced along with the degraded remote sensing image response, which is denoted as $n(x,y)$. The main purpose is to simulate the noise interference generated during the generation or transmission of remote sensing images. The final result of the degradation of the remotely sensed image is derived, i.e. $g(x,y)$, the recovery function of the image can be derived by inverse analysis of the degradation function, but since the general degradation process and the noise distribution cannot be accurately described, an approximate estimate of the original image is derived, denoted as $F(x,y)$. Under this model, the relationship between the degraded image and the original image can be derived as.

$$g(x,y) = h[f(x,y)] + n(x,y) \tag{1}$$

In the process of remote sensing image recovery processing, assuming that the degradation function is linearly invariant, then after the derivation it can be obtained as

$$h(x,y) = \frac{[g(x,y) - n(x,y)]}{f(x,y)} \tag{2}$$

The degradation function under the mathematical model of remote sensing image degradation can thus be derived.

The preprocessing of remote sensing image recovery is divided into several steps, such as binary image conversion, grayscale equalization, image enhancement, noise removal, and atmospheric radiation correction, etc. The main purpose of the preprocessing is to remove the noise interference in the original remote sensing image, enhance the characteristics of the base image, and convert the original remote sensing image into a format that can be directly processed by computers [20-21].

The derived degradation function $h(x,y)$ can be derived by doing the deconvolution derivation process, and the process of remote sensing image recovery function operation is shown in Equation (3).

$$H(x,y) = h^{-1}[h(x,y)] \tag{3}$$

The FAST corner point detection method is used to detect the feature points in remote sensing images, and the detection results of the feature corner points are used as the starting diffusion coordinates of the image recovery function [22-23]. Assuming that the covariates corresponding to the fitted straight lines of 2 adjacent feature points under the recovery function can be expressed (a_1,b_1) as and (a_2,b_2), the coordinates of the feature corner points are

$$\begin{cases} \hat{x} = \dfrac{b_1 - b_2}{a_1 - a_2} \\ \hat{y} = \dfrac{a_1 b_2 - a_2 b_1}{a_1 - a_2} \end{cases} \tag{4}$$

Taking the solved coordinates of the feature corner points as the recovery starting point, the remote sensing image is iterated with the derived recovery function by doing diffusion convolution, and the recovery result of the closest approximation to the original image can be estimated by iterative operations. The iterative process can be expressed as follows.

$$F(x,y) = f_k(x,y)\left[h(-x,-y) * \frac{f(x,y) * h(x,y) + n(x,y)}{h(x,y) * f(x,y)} \right] \tag{5}$$

Where the parameter $f_k(x,y)$ represents the coordinates of the FAST feature corner points, and $F(x,y)$ is the remote sensing image recovery result.

2.2 Remote sensing image target contour vectorization processing

On the basis of the remote sensing image recovery results, the FAST corner point detection method is used to realize the vectorization processing of the target contour in the remote sensing image, and the specific processing flow is shown in Figure 1. According to the processing flow in the figure, firstly, the multi-scale area morphology segmentation processes the remote sensing recovered image and extracts the target area, then the corner point detection method is used to record the boundary points of the target area and extract the corner points on the boundary curve, and finally the corner points are adjusted by using the geometric constraints and connected in a fixed way, so that the vectorization results of the target contour of remote sensing image can be obtained [24-25].

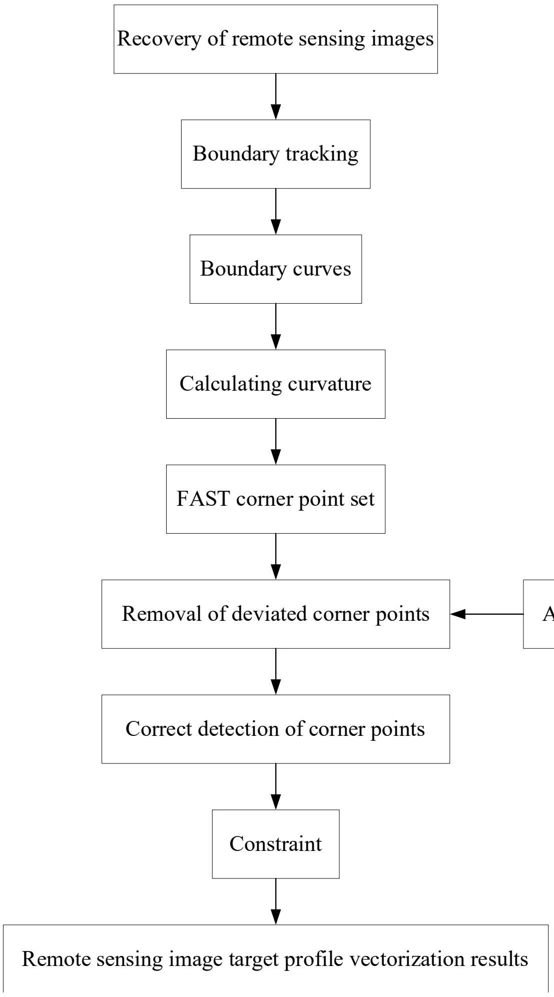
Recovery of remote sensing images
Boundary tracking
Boundary curves
Calculating curvature
FAST corner point set
Removal of deviated corner points
Adaptive support area
Correct detection of corner points
Constraint
Remote sensing image target profile vectorization results

Figure 1. Flow chart of target profile vectorization processing

In the process of target contour corner point detection in FAST, a pixel in a remotely sensed image can be identified as a target contour corner point if it has a large difference in gray value from a sufficient number of pixel points in its surrounding neighborhood [26-27]. Based on the above detection principle, multiple steps are taken to derive accurate focus detection results, respectively. In order to reduce the computational effort of target contour corner point detection and improve the efficiency of corner point detection, the target location of the remote sensing image is first determined and the target boundary is sampled and processed. The key points on the target boundary are selected to represent the boundary, however, the target contour is not a regular graph, so the curvature of the edge contour needs to be calculated to derive the corner point detection results [28]. The smoothing of the target boundary in the remote sensing image is achieved using the filter no, followed by the calculation of the curvature on the target contour curve using the following equation.

$$k(u,\sigma)=\frac{x_u(u,\sigma)y_{uu}(u,\sigma)-x_{uu}(u,\sigma)y_u(u,\sigma)}{\left(x_u(u,\sigma)^2-y_u(u,\sigma)^2\right)^{\frac{3}{2}}} \tag{6}$$

Where u and σ are the arc length and scale parameters of the target contour, respectively. The solution of other parameters can be expressed as

$$i_j(u,\sigma)=i(u)\otimes g_j(u,\sigma) \tag{7}$$

Where the values of i is x or y, the values of j is u or uu, and the corresponding $g_u(u,\sigma)$ and $g_{uu}(u,\sigma)$ are the first and second order derivatives of $g(u,\sigma)$ with respect to u, respectively. The operator $\otimes$ denotes the convolution operation. The curvature of the target contour can be solved by substituting the solution result of Eq. (7) into Eq. (6).

The curvature of the reference contour can extract all the corner points in the target contour, however, there are focal points generated by edge noise interference and rounded corner points in the detection results of corner points, so the erroneous corner points containing the above-mentioned cases need to be eliminated in the extraction process of the target contour focal points [29-30].

The transformation method of mathematical morphology is used to refine the processing of the target contour corresponding to the corner points and set the vectorization constraints. To ensure the efficiency of vectorization processing of the image, geometric right angle constraints are set. The initial vectorization result is obtained by connecting the adjacent corner points in a clockwise manner with the recovery feature corner points as the starting point, and connecting the neighboring corner points in a straight line segment in the first and last order. Then, the midpoint, slope and length of the frontal line segment between any two corner points are calculated. The deviation between the straight line segment and the main direction is calculated separately. Taking the number of edge points at a certain interval as a step, set the main direction of the contour corner point calculation as θ, adjust the slope of each straight line segment to θ or $\theta+90^{\circ}$, combine the frontal length and midpoint position coordinates of the initial straight line segment, adjust the position of the target contour corner point, and obtain the final vectorization result.

3 Experiments and results analysis

3.1 DPoI consensus algorithm design

DPoI calculates the importance score of a node using the node's time to find random numbers, as well as the node's activity, transaction volume and reputation value. The SHA256 hash function is introduced to calculate the time taken by nodes to find random numbers to enhance the importance of nodes with higher arithmetic power. The longer a node takes to find a random number, the worse its arithmetic power is, and therefore the less important it is. When a node finds Nonce, it immediately broadcasts it to the whole network. To reduce the difficulty of the hash calculation and the arithmetic power spent on finding random numbers, the last 4 digits of the timestamp in the last block are set to the random number (Nonce) required in the hash calculation, and the percentage of the time to find Nonce, *Ltime*, is used as a factor to evaluate the importance score. The *Ltime* of the top 80% of nodes is recorded as the ratio of the time spent by the node to the time spent by the last place; the *Ltime* of the bottom 20% of nodes is recorded as 1. The activity and transaction percentage of the previous round are introduced to fully evaluate the importance of the node in the current consensus round. After a node initiates a transaction, it needs to broadcast it to the whole network immediately. On the basis of nodes agreeing to the transaction, these nodes will broadcast this transaction to the whole network again. The transaction volume of a node involved in a consensus round is recorded as the transaction volume of that node, and the total transaction volume of the whole network in a consensus round is recorded as the total transaction volume. In each consensus round, the ratio of the node's participation in broadcasting to the total number of broadcasts in the system is taken as the activity, denoted as $aValue$, and the ratio of the node's participation in transaction volume to the total transaction volume in the system is taken as the transaction share, denoted as *iTrade*.

1) Voting reputation function

To improve the motivation of nodes to vote, a voting reputation function is established. Based on the number of nodes participating in each consensus round, the voting reputation value of the nodes is calculated, and the more nodes participate in voting, the smaller $vCredit$ is, and the increase in $Credit$ has a boosting effect, as an incentive for most nodes to participate in voting. In the Byzantine fault-tolerant algorithm (PBFT), the Byzantine nodes can tolerate at most one-third of the total number of nodes in the system to "betray", that is, if more than two-thirds of the nodes are normal, the whole system can work normally. Borrowing from the PBFT algorithm, this paper stipulates that as the number of votes decreases, the nodes will receive less revenue. The voting reputation is defined as follows.

$$vCredit = \begin{cases} \sum_{i=1}^{k} X_i - \sum_{i=1}^{n} Y_i, m \geq \frac{2}{3}n \\ \frac{n-m}{n}, m < \frac{2}{3}n \end{cases} \tag{8}$$

Where m is the number of nodes participating in voting in the current round; n is the number of summary points in the current round; k is the number of completed consensus rounds from the time the system starts operation; X_i is the number of times a node votes normally in the i th consensus round, and multiple votes can be conducted within a node group until the bookkeeping node is elected and the group rotation ends; Y_i is the number of times the node does not vote in the i th consensus round.

2) Bookkeeping reputation function

To reduce the probability of successful block creation by a malicious node, a bookkeeping reputation function is established to evaluate the bookkeeping reputation value in this round. The successful or unsuccessful bookkeeping behavior of the node in the i th round is used as an influencing factor to calculate the credibility value of the node in the $i+1$ th round. The bookkeeping reputation is defined as follows.

$$aCredit = \begin{cases} 1, \text{non-booking node in round i} \\ \dfrac{M}{T_{i-1,i}}, \text{booking node in round i} \end{cases} \tag{9}$$

Where each consensus round generates one block; the non-bookkeeping node in the first round is uniformly set to 1; if the bookkeeping node succeeds in bookkeeping, the value is 1, otherwise it is 0. denotes the block-out time (in min) in the first round.

Importance evaluation process: each node finds Nonce by hash calculation according to the currently set difficulty value, and when each node finds Nonce, it immediately broadcasts it to the whole network, and then finds a random number of nodes and calculates the importance score iValue of the node by combining the four values of Ltime, aValue, iTrade and Credit. iValue is calculated as: iValue

$$iValue = \mathrm{a}Value + iTrade + Credit - Ltime \tag{10}$$

It reduces the probability of a node with stronger arithmetic power to obtain bookkeeping rights, increases the weight of reputation value, transaction volume and activity, dynamically evaluates the importance of nodes, strengthens the influence of reputation value on the competition for bookkeeping rights, and helps to reduce the occurrence of mischievous behavior of miner nodes.

The nodes of the edge layer blockchain network are served by CA-authenticated edge servers, which usually only fail to participate in the consensus process as expected due to downtime and network delays, but the possibility of nodes being maliciously hijacked cannot be ruled out. Therefore, the reputation mechanism is introduced and the reputation value is used to describe the trustworthiness of the nodes. The reputation status of nodes can be classified into four categories, ST0, ST1, ST2, and ST3, depending on their reputation status. The reputation status of nodes changes with their behavior in the consensus process, and nodes that perform well in the consensus process are rewarded with reputation value, and the reward formula is

$$r_i^j = r_i^j + r_{\text{reward}} \tag{11}$$

Where is the reputation value of the first node in the first slice and is the rewarded reputation value. Nodes that make wrong decisions in the consensus process are penalized by the reputation value, and the penalty formula is

$$r_i^j = r_i^j - r_{\text{punishment}} \tag{12}$$

Where is the reputation value of the penalty, and the values of and can be adjusted according to the actual application scenario.

Based on the access mechanism of the federated chain and the description of nodes' trustworthiness status by the above reputation mechanism, malicious nodes will be isolated from the network. Therefore, when slicing the edge layer blockchain network, there is no need to ensure the security of the blockchain network after slicing through complex calculations. Based on the topology between edge servers, the edge layer blockchain network can be represented as a matrix in the form of.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \tag{13}$$

Where is the adjacency matrix of the edge-layer blockchain network, which is used to depict the connectivity between the nodes. It means that there are edges between the nodes and are connected; it means that there are no edges between the nodes and are connected and need to be forwarded by other nodes to communicate, and it is specified.

For an edge-layer blockchain network with n nodes, the improved FN algorithm consists of the following execution steps.

Step 1: The network is initialized into n slices, i.e., each node acts as a slice each. At this point Q = 0, and satisfies the following equation.

$$e_{ij} = \begin{cases} \frac{1}{2m}, n_i \sim n_j \\ 0, \text{other} \end{cases} \tag{14}$$

$$a_i = \frac{k_i}{2m} \tag{15}$$

Where is the degree of the node with edges connected to it.

Step 2: Iterate through the pairs of connected edges and determine whether the constraints are satisfied after merging the pairs.

$$card\left(s_i \cup s_j\right) \le 100 \tag{16}$$

The function is used to calculate the number of elements of a finite set. Equation (16) is used to control the size of the merged slice to ensure that each slice contains no more than 100 nodes and to reduce the number of computations. If the slice pair satisfies Equation (16), the merged

$$\Delta Q_{ij} = e_{ij} + e_{ji} - 2a_i a_j = 2\left(e_{ij} - a_i a_j\right) \tag{17}$$

Then, according to the principle of the greedy algorithm, the slice pair that can increase Q the most or decrease it the least is selected and merged from the slice pairs that satisfy (16). After each round of merging, update the corresponding, and sum the rows and columns of the corresponding, slices in the matrix, and then calculate.

Step 3: Repeat Step 2 to continuously merge the slices until the convergence condition is satisfied.

$$\forall s_k、 s_l \in S, \quad card\left(s_k \cup s_l\right) \ge 100 \tag{18}$$

The physical meaning of Equation (18) is that continuing to merge any two slices in the network will result in a merged slice size greater than 100. With the help of Equation (18), the improved FN algorithm in this paper can reduce the number of merging rounds and thus improve the time performance of the algorithm. To avoid low credibility nodes from affecting the consensus process of the network, the node authority is classified according to the credibility status of the nodes, as shown in Table 1.

Table 1. Classification of node permissions

Node Credibility Status	Ability to run for master node	Ability to participate in consensus
ST3	Yes	Yes
ST2	No	Yes
ST1	No	Yes
ST0	No	No

3.2 Performance Analysis

The two performance metrics are slice time and slice result Q-value, and the comparison scheme is the classical association partitioning algorithm FN algorithm, and the experimental results are averaged over multiple runs. First, we compare the time spent by both algorithms for the same network partitioning to verify the performance of the improved FN algorithm in reducing the partitioning time. Then the modularity Q values of the two algorithms are compared for the same network after binning to measure the binning quality of the improved FN algorithm. The datasets used for the experiments are five relational network graphs from Stanford University's large network dataset site and MarkNewman's personal dataset site, and the size information of the five networks is shown in Table 2.

Table 2. Basic information of the data set

Number	Network Name	Number of nodes	Number of edges
1	Dolphin social network	62	159
2	American college football	115	613
3	Neural network	297	2359
4	Email-Eu-core network	1005	25571
5	Political blogs	1490	16718

The target contours were vectorized on the basis of the results of remote sensing image restoration, and the occupied space of the vectorized image and the distribution number of corner points were counted. The statistical results are shown in Table 3.

Table 3. Comparison results of contour vectorization

Experiment serial number	Traditional remote sensing image processing methods		Methodology of this paper	
	Space occupied /kB	Number of corner points	Space occupied /kB	Number of corner points
1	246	65	131	153
2	258	79	105	169
3	272	83	124	154
4	204	52	130	182
5	285	91	118	175
6	247	66	122	179

Through the calculation, it can be found that the average occupied space of the image resulting from the traditional image processing method is 252kB and the average number of corner points is 73, while the average occupied space of the corresponding image using the design method is 122kB and the average number of corner points is 169. It can be concluded that the FAST corner point detection-based remote sensing image recovery and target contour vectorization can effectively compress 35.6% of the occupied space, and the processing accuracy of vectorization is higher compared with the traditional image processing methods, achieving the dual requirements of compressing data and ensuring accuracy.

In order to verify the performance of the minimum depth spanning tree algorithm in reducing the block broadcast time, the spanning tree depth of the master node is selected as the performance index. The comparison scheme is the master node selection strategy adopted by the consensus algorithm, and the experimental results are averaged over multiple runs. To ensure the fairness of the experiment, the PBFT algorithm can only select the master node from the alternative node set of the minimum depth spanning tree algorithm; then compare the impact of the number of slices on the performance of the minimum depth spanning tree algorithm; finally, test the impact of network connectivity on the performance of the minimum depth spanning tree algorithm. Finally, we test the effect of network connectivity on the performance of the minimum depth spanning tree algorithm.

The experimental data are generated from the Salama model, a stochastic network generation model, which has two important network characteristics, namely, the ratio of short edges to long edges, and the density of edges in the network. The larger the and determines the connectivity of the network, the better the connectivity of the network. The experiment generates networks of different sizes but with the same connectivity to simulate each piecewise blockchain network by controlling the network characteristics parameters of the Salama model. In the following, unless otherwise specified, the experimental networks are generated by the same ratio to ensure the same connectivity for different sizes of networks.

As can be seen from Figure 2, when the same network is divided into different slices, the spanning tree depth of the master node selected by the minimum depth spanning tree algorithm for each slice is inversely proportional to the number of slices. This is because the experiment assumes that each slice has the same size (contains the same number of nodes) for the control variable. So the more the number of slices, the smaller the size of each slice, the smaller the spanning tree depth of the master node selected by the minimum depth spanning tree algorithm. Based on this, HSChain can dynamically divide the edge layer blockchain network according to the business demand of the industrial Internet platform layer, and divide the network into more slices when the network load is large to prioritize the throughput demand; when the network load is small, divide the network into fewer slices to better secure the network.

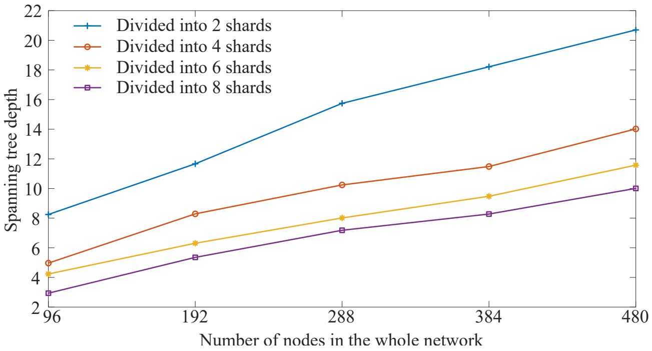

Figure 2. Comparison of spanning tree depth with different number of shards at the same network size

As can be seen from Figure 3, the larger the slice (i.e., the better the connectivity of the network), the smaller the spanning tree depth of the master node selected by the minimum depth spanning tree algorithm. This is because the spanning tree depth of a connected graph (a network in which any two nodes are connected by edges) has a theoretical upper and lower limit. A globally coupled network with n nodes requires only one hop to reach any position in the network, which has a constant spanning tree depth of 1 and the best connectivity, while a network with n n nodes and n-1 edges has a minimum possible spanning tree depth of (n-1)/2 and a maximum possible depth of n-1, which has the worst connectivity. Therefore, the better the connectivity is, the closer the minimum possible depth of the spanning tree is to 1.

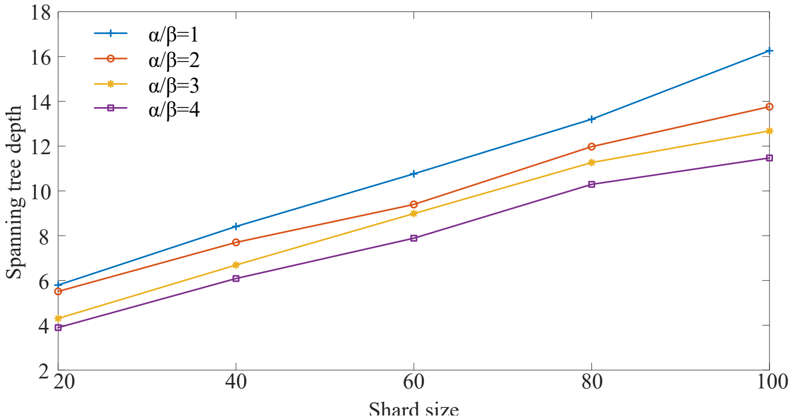

Figure 3. Comparison of spanning tree depth for different connectivity shards with the same size

4 Conclusion

This paper addresses the problems of insufficient throughput and high storage pressure faced by blockchain when applied to engineering pavilions. Firstly, based on vectorized image processing technology, a tiered storage architecture is proposed to maintain the massive data generated by the engineering pavilion in the edge blockchain layer and the cloud blockchain layer in a tiered manner to solve the problem of insufficient blockchain storage capacity. Then a blockchain network is analyzed from the perspective of complex networks, and a blockchain network partitioning algorithm is designed and improved based on the classical association partitioning algorithm, which shortens the partitioning time while improving the blockchain throughput. Finally, the impact of block broadcast time on throughput is demonstrated, and a master node selection algorithm that can minimize the block broadcast time within each slice is proposed to further improve the throughput of the edge layer blockchain. The analysis and experimental results show that the proposed scheme can reduce the broadcast time of blocks within each slice while reducing the slice time compared with the classical association division algorithm and the strategy of randomly selecting master nodes. The next work will improve the offloading mechanism of edge layer transaction blocks and design a more efficient consensus algorithm for the engineering hall scenario.

References

[1] Li, G., Zhao, Q., Zhang, D., et al. (2021). GT-Chain: A Fair Blockchain for Intelligent Industrial IoT Applications. IEEE Transactions on Network Science and Engineering, https://doi.org/10.1109/TNSE.2021.3099953

[2] Sen, S. and Song, L. (2021). An IIoT-Based Networked Industrial Control System Architecture to Secure Industrial Applications. IEEE Industrial Electronics and Applications Conference (IEACon). Penang, Malaysia: 280-285.

[3] Leng, J., Ye, S., Zhou, M., et al. (2021). Blockchain-Secured Smart Manufacturing in Industry 4.0: A Survey. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(1), 237-252.

[4] Wu,Y., Dai, H-N. and Wang, H. (2020). Convergence of Blockchain and Edge Computing for Secure and Scalable IIoT Critical Infrastructures in Industry 4.0. IEEE Internet of Things Journal, 8(4), 2300-2317.

[5] Zeng, P., Wang, X., Dong, L., et al. (2021). A Blockchain Scheme Based on DAG Structure Security Solution for IIoT. IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). Shenyang, China, 935-943.

[6] Javaid, U. and Sikdarm B. (2020). A Checkpoint Enabled Scalable Blockchain Architecture for Industrial Internet of Things. IEEE Transactions on Industrial Informatics, 17(11), 7679-7687.

[7] Qi, X., Zhang, Z., Jin, C., et al. (2021). A Reliable Storage Partition for Permissioned Blockchain. IEEE Transactions on Knowledge and Data Engineering, 33(1), 14-27.

[8] Hafid, A. and Samih M. (2020). Scaling Blockchains: A Comprehensive Survey. IEEE Access, 8, 125244-125262.

[9] Cai, X., Geng, S., Zhang, J., et al. (2021). A Sharding Scheme based Many-objective Optimization Algorithm for Enhancing Security in Blockchain-enabled Industrial Internet of Things. IEEE Transactions on Industrial Informatics, 17(11), 7650-7658.

[10] Luu, L., Narayanan, V., Zheng, C., et al. (2016). A Secure Sharding Protocol for Open Blockchains. The 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16). New York, USA: ACM, 17-30.

[11] Kokoris-Kogias, E., Jovanovic, P., Gasser, L., et al. (2018). OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding. IEEE Symposium on Security and Privacy (SP). San Francisco, USA: IEEE, 583-598.

[12] Zamani, M., Movahedi, M. and Raykova, M. (2018). RapidChain: Scaling Blockchain via Full Sharding. The 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18). New York, USA: ACM, 931-948.

[13] Wang, J. and Wangm H. (2019). Monoxide: Scale out blockchains with asynchronous consensus zones. 16th USENIX Symp. Netw. Syst. Design Implement. Boston, USA: USENIX, 95–112.

[14] Yeasmin, S. and Baig, A. (2020). Permissioned Blockchain-based Security for IIoT. 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). Vancouver, Canada: IEEE, 1-7.

[15] Tong, W., Dong, X., Shen, Y., et al. (2019). A Hierarchical Sharding Protocol for Multi-Domain IoT Blockchains. ICC 2019-2019 IEEE International Conference on Communications (ICC). Shanghai, China: IEEE, 1-6.

[16] Huang, C., Wang, Z., Chen, H., et al. (2020). RepChain: A Reputation-Based Secure, Fast, and High Incentive Blockchain System via Sharding. IEEE Internet of Things Journal, 8(6), 4291-4304.

[17] Girvan, M. and Newman, M. E. J. (2002). Community Structure in Social and Biological Networks. Proceedings of the National Academy of Sciences of the United States of America, 99(12), 7821-7826.

[18] Fang, C., Tao Y., et al. (2021). Research on Leakage Current Waveform Spectrum Characteristics of Artificial Pollution Porcelain Insulator. Frontiers in Energy Research. https://doi.org/10.3389/fenrg.2021.798048

[19] Newman, M. E. J. (2004). Fast Algorithm for Detecting Community Structure in Networks. Physical Review E Statistical Nonlinear & Soft Matter Physics, 69(6), 066133.

[20] Newman, M. E. J. and Girvan M. (2004). Finding and evaluating community structure in networks. Physical Review E Statistical Nonlinear & Soft Matter Physics, 69(2), 026113.

[21] Sukhwani, H., Martinez, J. M., Chang, X., et al. (2017). Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric). IEEE 36th Symposium on Reliable Distributed Systems (SRDS). Hong Kong, China: IEEE, 253-255.

[22] Yu, Y., Liu, S., Yeoh, P. L., et al. (2020). LayerChain: A Hierarchical Edge-Cloud Blockchain for Large-Scale Low-Delay IIoT Applications. IEEE Transactions on Industrial Informatics, 17(7), 5077-5086.

[23] Fang, C., Tao Y., et al. (2021). Mapping Relation of Leakage Currents of Polluted Insulators and Discharge Arc Area. Frontiers in Energy Research, https://doi.org/10.3389/fenrg.2021.777230

[24] Castro, M. and Liskov, B. (1999). Practical Byzantine Fault Tolerance. The third symposium on operating systems design and implementation (OSDI '99). New Orleans, USA: USENIX Association, 173-186.

[25] Salama, H. F. (1996). Multicast Routing for Real-time Communication on High-speed Networks. Raleigh, USA, North Carolina State University.

[26] Li, W., Feng, C., Zhang, L., et al. (2020). A Scalable Multi-layer PBFT Consensus for Blockchain. IEEE Transactions on Parallel and Distributed Systems, 32(5), 1146-1160.

[27] Wang, G., Shi, Z. J,. Nixon, M., et al. (2019). SMChain: a scalable blockchain protocol for secure metering systems in distributed industrial plants. International Conference on Internet-of-Things Design and Implementation. New York, USA: ACM, 249-254.

[28] Yang, J., Paudel, A., Gooi, H. B., et al. (2021). A Proof-of-Stake Public Blockchain-Based Pricing Scheme for Peer-to-Peer Energy Trading. Applied Energy, 298, 117154.

[29] Fang, W., Zhang, W., Pan, T., et al. (2018). Cyber Security in Blockchain: Threats and Countermeasures. Journal of Cyber Security, 3(2), 87-104.

[30] Cho, H. (2018). ASIC-Resistance of Multi-Hash Proof-of-Work Mechanisms for Blockchain Consensus Protocols. IEEE Access, 6, 66210-66222.