Optimization Scheme of PBFT Consensus Algorithm Based on Changan Blockchain

Ziqiang Zhou[1,2], Xianke Zhou[3], Jiajia Han[4], Peng Lu[3*], Ying Yao[4], Shuang Hu[3]

[1] *Zhejiang Huayun Clean Energy Co., Ltd. Hangzhou, China, 310008*
[2] *College of Electrical Engineering Zhejiang University，Hangzhou, China, 310027*
[3] *Institute of Computing Innovation, Zhejiang University, Hangzhou, China, 310008*
[4] *State Grid Zhejiang Electric Power Company Electric Power Research Institute, Hangzhou, China, 310014*

Abstract

Blockchain has the characteristics of tamper resistance, traceability, promoting data sharing, optimizing business processes, reducing operating costs, improving collaboration efficiency and building trusted application systems. Blockchain is applied to high elastic grid to meet the demand of mutual trust, reducing costs, and improving user involvement. A scheme of PBFT consensus algorithm is proposed in this paper to improve performance. Firstly, by analyzing the architecture and core processes of the existing alliance chain, the main factors affecting the performance are located. Secondly, an optimized Byzantine fault-tolerant consensus algorithm based on parallel execution and signature caching is proposed. Finally, the improved consensus scheme is tested and analyzed on Changan blockchain. Experiments show that the scheme significantly improves the performance of the alliance chain.

Keywords

consesus algorithm, Changan blockchain, PBFT, high elastic grid

1. Introduction

The traditional power grid is becoming smarter with the rapid development of light, wind, and heat. The intelligence, security and interactivity of smart grid make the intelligent control of "power, information and business flow" and the two-way communication between suppliers and users possible, so that the power grid can realize intelligent interactive operation.

Blockchain is decentralized, confidential and traceable, which meets the needs of distributed, interactive and data security of smart grid. The consensus algorithm guarantees data security and authenticity across nodes in a blockchain system without central nodes. Blockchain consensus algorithms [1,2,3] are evaluated by throughput, proof-of-work, decentralization, and algorithm security. We will introduce several major blockchain consensus algorithms, such as PoW, PoS, DPos, PBFT and RAFT.

PoW Consensus mechanism. PoW is a proof-of-work based consensus mechanism [4]. Higher arithmetic power increases a node's likelihood of becoming a bookkeeping node. The PoW mechanism is environmentally unfriendly; the more arithmetic power, the longer the mining time and the more bitcoins obtained. In a smart grid environment with few active nodes, a malicious node may have more than half of total computing power, which makes it possible to create a longer branch chain to complete an arithmetic attack. So PoW isn't suitable for smart grid blockchain.

ISCIPT2022@7th International Conference on Computer and Information Processing Technology, August 5-7, 2022, Shenyang, China

EMAIL: jx_zzq@sina.com (Ziqiang Zhou); xkzhou@zjuici.com (Xianke Zhou); han_jiajia@zj.sgcc.com.cn (Jiajia Han); lupeng@zjuici.com (Peng Lu); yao_ying@zj.sgcc.com.cn (Ying Yao); hs@zjuici.com (Shuang Hu)
ORCID: 0000-0001-5191-5321 (Ziqiang Zhou); 0000-0002-4106-7936 (Xianke Zhou); 0000-0002-9270-2258 (Jiajia Han); 0000-0002-9200-7896 (Peng Lu); 0000-0002-4343-2183 (Ying Yao); 0000-0003-0803-7257 (Shuang Hu)

PoS Consensus mechanism. PoS [5] is a consensus mechanism based on proof of stake; node equity is proportional to bookkeeping rights. In PoS, the higher the equity, the easier it is for the user to obtain bookkeeping rights, which leads to less active nodes and hinders token circulation in the blockchain network. So Pos is not suitable to be used in smart grid directly.

DPoS Consensus Mechanism. DPoS uses voting elections to achieve consensus [1,5]. Elected representatives generate consensus blocks. Voters can remove ineffective representatives. DPoS improves throughput and reduces latency, while there still some problems such as low voter enthusiasm and the inability to quickly deal with malicious nodes.

PBFT Consensus Mechanism. Practical Byzantine Fault Tolerance (PBFT) [3] guarantees activity and security in a network of m nodes with (m-1)/3 fault tolerance. The algorithm's performance decreases as the number of grid nodes increases due to its poor scalability and complex communication mechanism. Byzantine fault tolerance ensures (m-1)/3 security and activity [5]. Distributed computers agree through message exchange. HotStuff [6] solves view switching's high communication complexity. Tang Qiang's team at the New Jersey Institute of Technology [7] have proposed the first fully practical asynchronous Dumbo Byzantine Fault Tolerance (DumboBFT) algorithm. Chunmei Guo et al [8] improved the traditional distributed consensus algorithm, constructed the P-PBFT consensus algorithm, introduced the lift mechanism, and the nodes can be dynamically changed, proving that the algorithm has good throughput while reducing algorithm overhead; the Byzantine fault tolerance algorithm is designed to solve the Byzantine general problem.

RAFT Consensus Mechanism. Raft is a leader-based consensus algorithm [1] that accepts client requests and processes all nodes run by the leader's servers. To get around this, private blockchains use a different consensus algorithm that doesn't take into account the Byzantine fault.

Private, public, and alliance blockchains differ in node rights and interests. In public chain, all users can join the blockchain network, and PoW, PoS, etc. are typical consensus mechanisms; in private chain, all nodes in the system must be trusted, and Paxos, Raft are typical consensus mechanisms; in alliance chain, only nodes that pass identity authentication can join the blockchain network. Alliance chains can better meet enterprises' needs for real-world business scenarios than public chains. This paper proposes a modified blockchain-based smart grid consensus optimization scheme.

2. Analysis of the Performance Constraints of Smart Grid Consensus Mechanism

Through study of blockchains such as Changan blockchain (Chainmaker), Hyperledger Fabric, and Tendermint, the transaction execution process of the alliance chain can be summarized as shown in Figure 1.

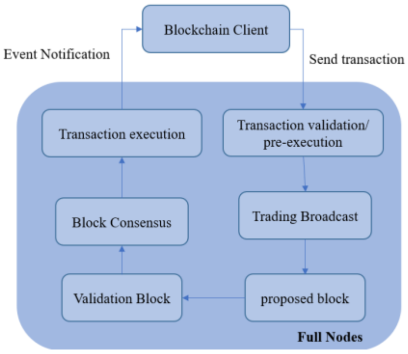

Figure 1: Typical alliance chain transaction execution process

Figure 1 shows a typical alliance chain's core process. After receiving a transaction through the RPC interface, the node verifies the client ID book, transaction signature, transaction nonce, and other information. Some alliance chains also perform contract pre-execution on the transaction and return the read/write set. After verification, the transaction is stored in a pool and broadcast over P2P. The outgoing node packs the transaction pool into blocks, adds signatures, and broadcasts them to other nodes. Other nodes verify the new block's legitimacy and agree on it using consensus algorithm. Alliance chain nodes execute new block transactions and update ledger data and contract status.

In a business scenario, the alliance chain's performance focus on transaction confirmation speed and throughput. TPS equals single block transaction capacity/block-out interval time, while TCS equals transaction confirmation speed. Therefore, performance constraint factors can be found through analyzing block transaction capacity and outgoing block interval time. Theoretically, increasing block transaction capacity and shortening block out interval can directly improve TPS. Bitcoin's block size is fixed at 1MB and block out interval is 10 minutes, while the corresponding block size of Ethernet is 10million gas and 14 seconds. In Changan blockchain, block size is 100 and interval is 1s~2s. Take Changan blockchain's certificate contract as an example: the node with Intel Xeon@2.4GHz 2-core, 8G Memory, Centos7.1, and 200G HardDisk is tested. In a blockchain network with 200G nodes, order nodes and peer nodes execute 5KB data storage transactions. The maximum number of transactions in a single block and the block interval are modified and tested. The TPS peaks when the block capacity = 1200 and the block interval = 0.5s, and then declines. Figure 2 shows the relationship between block capacity and TPS.

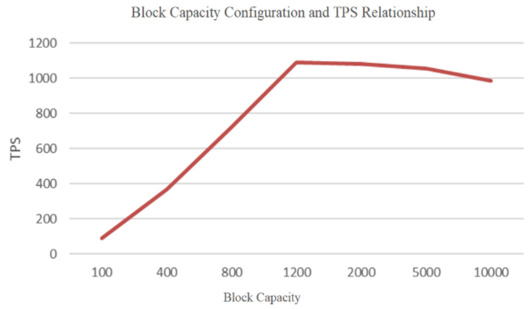

Figure 2: Block capacity allocation versus TPS

Combined with the typical alliance chain transaction execution process in Table 1, the above phenomenon reveals: the entire transaction process from sending, packing to execution is serial, and each step of the execution process has a fixed time overhead, such as signature verification, transaction broadcasting, consensus voting, etc. No matter how the block-out configuration is changed, the minimum processing time cannot be changed. Increasing block capacity affects transfer speed, propagation success rate, execution speed, and block out interval time. Increasing block capacity and shortening block out time within a certain range can improve TPS.

Table 1
Execution time of each phase at TPS peak

Process Phase	Time taken (%)
Transaction validation/pre-execution	29.6
Block Proposal	5.1
Block Consensus	57
Transaction execution	8.3

The consensus algorithm, which accounts for 57% of blockchain performance, is the dominant factor, according to the analysis. The consensus algorithm and process are improved to reduce serialized links in the core process. DAG-based consensus [9] allows concurrent transactions, and Consensus algorithms can be optimized according to its characteristics. For example, DAG consensus algorithm has the problem of transaction confirmation, while Hash Graph has the problem of transaction repetition, which affect block validity and is easily attacked by malicious nodes. This paper proposes an optimized consensus algorithm scheme to combine the above directions and highly resilient grid blockchain business scenarios.

3. Consensus algorithm optimization scheme

Caching, parallelism, and cryptographic algorithms have been studied to improve blockchain performance. The literature [10] proposes a performance optimization method for single-chain blockchain transactions based on caching technology and refines and analyzes the resource consumption and latency of each step of the inefficient block submission process. Using multi-threading and a modified IBFT voting algorithm, [11] improves PBFT transaction throughput to 1140 transactions/sec. Literature [12] proposed a sharing, aggregated signatures, and cryptographic draws public chain system. A FPGA-based hash acceleration optimization algorithm for the blockchain is proposed to address the low computational efficiency of the blockchain's hash function, which affects overall computational efficiency and even leads to security risks. By combining blockchain and FPGA acceleration card, the hash algorithm improves the hash function's computational performance. A lightweight hashing algorithm is chosen to perform multiple hashes and transform the hash algorithm structure to ensure data security.

Blockchain is required for grid participant identity management, power transaction aggregation, and supply usage record verification, allowing thousands of small-scale energy sources to participate in grid stabilization services through trusted market mechanisms. Fast transaction confirmation, many supported nodes, and reliable data are needed. RAFT algorithm cannot meet decentralization security requirements, while the performance of PBFT consensus decreases as the power grid becomes larger.

In the power grid scenario, in order to ensure the fairness of transactions, the degree of decentralization of power transaction matching links is required to be high. In the power supply and consumption link, it is required to store the power supply and consumption data, which cannot be tampered with and can be audited. We flexibly select the consensus algorithm's weight and selection to solve the performance problem. Figure 3 shows the flow chart of our consensus algorithm.

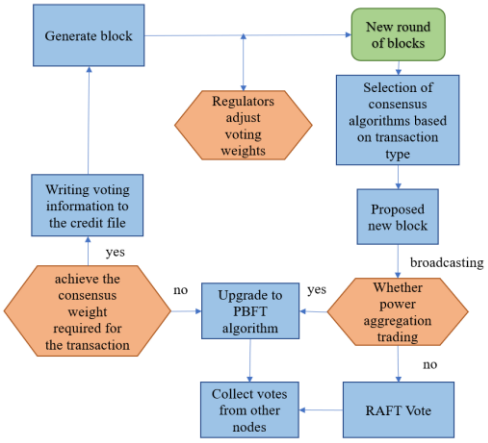
Generate block
New round of blocks
Regulators adjust voting weights
Selection of consensus algorithms based on transaction type
Writing voting information to the credit file
Proposed new block
yes
broadcasting
achieve the consensus weight required for the transaction
no
Upgrade to PBFT algorithm
yes
Whether power aggregation trading
no
Collect votes from other nodes
RAFT Vote

Figure 3: Consensus algorithm optimization flow chart

All blockchain nodes participating in the consensus process need to pass the authentication. In different scenarios such as power matchmaking transactions and power access licenses, the identity of each node is different, and the corresponding permission scenarios and voting weights are also different. This ensures data isolation and blockchain security. When processing transactions, it switches between RAFT and PBFT depending on the type of transaction. If it's a power matchmaking transaction, it uses PBFT voting; otherwise, it uses RAFT voting. After consensus, all nodes' votes are collected and judged to see if the transaction's consensus weight is reached. If so, the voting information is written to the grant file; otherwise, it returns to the PBFT algorithm for consensus.

In the whole process control, the grid platform manager can dynamically adjust the voting weight according to the credit score in the voting record, and give rewards and penalties in the audit. By controlling the weight of the vote can control whether the transaction is up to standard, for the substandard transaction naturally will not be executed. This can strengthen the regulator's ability to control the entire transaction information recording process.

4. Experimental results and analysis

This section verifies the effectiveness of the optimization strategies on system performance improvement through several comparison experiments. In the following, Section 4.1 details the specific configuration and metrics of this experiment, Sections 4.2 shows the comparative experiments on the performance impact of optimization strategies.

4.1. Experimental setup

This experiment uses 4 units to build a cluster of pressure testing clients and 4 blockchain nodes for configuration. This experiment uses LAN networking, and 4 blockchain nodes form a blockchain cluster and simulate network interconnection through a switch. The network topology diagram as shown in figure 4.

The specific node configuration of Changan blockchain for this experiment is as follows: encryption algorithm: national secret, block interval: 2000ms, block capacity: 5000, consensus algorithm: TBFT (optimized PBFT algorithm).

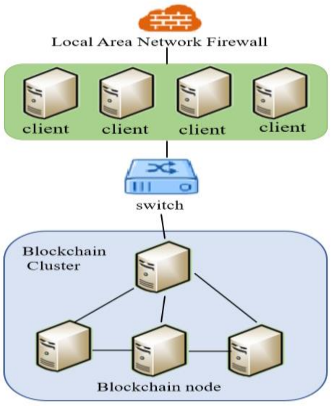

Figure 4: Experimental network topology

4.2. Consensus algorithm optimization experiments

In this section, the default TBFT algorithm and the optimized consensus algorithm of Changan blockchain will be used to execute 50,000 tariff matchmaking transactions and grid certificate deposit transactions respectively. Table 2 shows the TPS of the consensus algorithm optimization test. Table 3 shows consensus algorithm optimization test transaction confirmation time. It shows that ours scheme improved performance obviously.

Table 2

TPS of the consensus algorithm optimization test

Consensus algorithm	Avg TPS	Total TRANS	Execution time
TBFT	872	100000	1min57s
ours	1153	100000	1min36s

Table 3

Consensus algorithm optimization test transaction confirmation time

Consensus algorithm	AVG TRANS CONF time	Total TRANS	AVG block CAP
TBFT	4.6s	100000	4000
ours	3.9s	100000	4500

5. Conclusion

This paper addresses the characteristics of low operational efficiency and high computational overhead of the serial mechanism of the coalition chain, and improves the consensus algorithm performance by switching between multiple consensus algorithms in combination with the highly resilient power grid business scenario; Experiments based on Changan blockchain shows the system performance is significantly improved.

This work was supported by Science and Technology Foundation of State Grid Zhejiang Electric Power Co. Ltd “research and application of state key technology of blockchain supporting high elastic power grid balance (B311DS21000H)”.

6. References

[1] Bamakan, Seyed Mojtaba Hosseini, Amirhossein Motavali, and Alireza Babaei Bondarti. "A survey of blockchain consensus algorithms performance evaluation criteria." Expert Systems with Applications 154 (2020): 113385. doi: 10.1016/j.eswa.2020.113385.

[2] Tang Chunming, Chen Yuqing, Zhang Zidi. Improved consensus algorithm based on binomial swap forest and HotStuff, 2021. URL:http://kns.cnki.net/kcms/detail/51.1307 . TP. 20210804.1442.010.html

[3] Ziegler M H, Großmann M, Krieger U R. " "Integration of fog computing and blockchain technology using the plasma framework." IEEE International Conference on Blockchain and Cryptocurrency (ICBC). IEEE, 2019: 120-123. doi: 10.1109/ BLOC. 2019.8751308.

[4] Zou Jun, et al. "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services." IEEE Transactions on Services Computing 12.3 (2018): 429-445. doi: 10.1109/TSC.2018.2823705.

[5] Lamport, Leslie, Robert Shostak, and Marshall Pease. "The Byzantine generals problem." Concurrency: the works of leslie lamport. 2019. 203-226. doi: 10.1145/ 3335772. 3335936.

[6] Li Qinan， Xue, Zhihao， Zhang Xuejun. "Improved Fast-HotStuff Blockchian Consensus Algorithm. " Computer Engineering,2021,47(08):14-21. doi: 10.19678/j.issn.1000-3428.0060847.

[7] Guo Bingyong, et al. "Dumbo: Faster asynchronous bft protocols." Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. 2020. doi: 10.1145/3372297.3417262.

[8] Guo Chunmei, Zhu Baoping． "An Improved Blockchain Consensus Algorithm." Computer & Digital Engineering,2020,048(006):1290-12931349. doi: 10.3969/j.issn.1672-9722.2020.06.006.

[9] Kan Jia, Shangzhe Chen, and Xin Huang. "Improve blockchain performance using graph data structure and parallel mining." 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN). IEEE, 2018. doi:10.1109/ HOTICN. 2018.8606020.

[10] Gorenflo, Christian, et al. "FastFabric: Scaling hyperledger fabric to 20 000 transactions per second." International Journal of Network Management 30.5 (2020): e2099. doi:10.1002/nem.2099.

[11] Samy, Hossam, et al. "Enhancing the performance of the blockchain consensus algorithm using multithreading technology." Ain Shams Engineering Journal 12.3 (2021): 2709-2716. doi: 10.1016/j.asej.2021.01.019.

[12] Fu Jinhua, et al. "A study on the optimization of blockchain hashing algorithm based on PRCA." Security and Communication Networks 2020 (2020). doi: 10.1155/2020/8876317.