Research and Implementation of Blockchain Consensus Mechanism Based on Improved DAG

Yuze Sun(✉) and Jian Wang

Beijing University of Technology, Beijing, China
714778255@qq.com

Abstract. Based on the development of digital currency, the blockchain technology has become increasingly popular, and the consensus mechanism is a critical aspect that requires further research. This paper proposes a new blockchain consensus mechanism, called UDAG, by combining the efficient chain consensus algorithm with the high concurrency of DAG structure. UDAG is based on the directed acyclic graph (DAG) structure and uses Pos-PBFT as the consensus algorithm. The proposed UDAG consensus mechanism improves transaction data in terms of timing, correlation, and immediacy compared to both the chain structure and the existing DAG structure. The experimental results demonstrate that UDAG is highly feasible, and it has the potential to significantly improve the performance of blockchain systems.

Keywords: Blockchain · DAG · PBFT · POS

1 Introduction

In recent years, blockchain technology has become a critical area of innovation, with the need to manage vast amounts of data in the modern era. However, the traditional blockchain chain structure has limitations in processing vast amounts of information, which could result in performance bottlenecks. As a result, the DAG (directed acyclic graph) [1] architecture has emerged as a preferred choice for enhancing blockchain performance.

DAG-style blockchains allow for concurrent transactions to be processed simultaneously, which is critical in improving the overall performance of the blockchain system. This is achieved through the use of asynchronous communication mechanisms, which enables a higher transaction throughput compared to the traditional chained blockchain structure. DagCoin [2] was the first to propose the theoretical concept of DAG-style blockchain, which was further developed by other projects such as IOTA [3] and ByteBall [4].

Despite the advantages of the DAG-style blockchain architecture, there is a major issue that arises from the asynchronous communication mechanisms used to process transactions - global chaos. To address this issue, the current focus of research in DAG-style blockchain solutions is on the master-chain delay selection approach. This approach

J. Yen et al. (Eds.): ICBIS 2023, AHCS 14, pp. 1442–1448, 2023.
https://doi.org/10.2991/978-94-6463-198-2_149

involves selecting historical transactions from the previous period that meet the requirements and linking them to form the master-chain. All subsequent transactions are then arranged according to the order established by the master-chain. However, this approach can result in conflicting master-chains for different users and inconsistent local historical transaction data.

To address the challenges posed by existing master-chain consensus algorithms in DAG-style blockchains, this paper proposes a novel consensus mechanism called User Chain Directed Acyclic Graph (UDAG).

UDAG is a user-chain based DAG consensus mechanism that incorporates the Pos-PBFT consensus algorithm and an improved DAG structure. The PoS-PBFT consensus algorithm [5] is used to set roles for PBFT [6] and to instantly construct the main chain subgraph of the DAG structure. This approach allows for control of the global state and transaction duration, while also shortening the consensus time.

2 Existing Research

Among the foreign DAG-style blockchains, IOTA is a well-known example that employs the Tangle model. This model creates a DAG structure by referencing two or more historical transactions and each block contains only a single transaction. The parallel verification feature of IOTA is a significant advantage over traditional blockchain architectures. However, IOTA still relies on a centralized coordinator to overcome issues related to the lack of global state and control over transaction duration. Additionally, security depends on the percentage of active users, further highlighting the limitations of the current approach.

Conflux [7] is a prominent DAG-style blockchain network in China that aims to improve the scalability of the Bitcoin system through a DAG structure. It employs a main-chain consensus mechanism and uses the GHOST protocol to select the main chain with the highest number of transaction subtrees, similar to other DAG-style blockchains. However, it still utilizes the PoW algorithm, which can be time-consuming and resource-intensive, limiting its practicality in certain scenarios. Moreover, the main-chain delay selection strategy employed by Conflux may result in inconsistencies in the selection of the main chain.

3 UDAG Consensus Mechanism Model Design

UDAG is divided into two main parts, the way the transaction blocks are organized and the consensus algorithm of the consensus group.

3.1 The Way UDAG'S Trading Blocks are Organized

3.1.1 The Way Blocks Are Organized

UDAG is designed to have a single chain for each user node, which means that only the user can create transactions associated with their node, as shown in Fig. 1. Each block in the UDAG is represented by a vertex and contains a single transaction. When

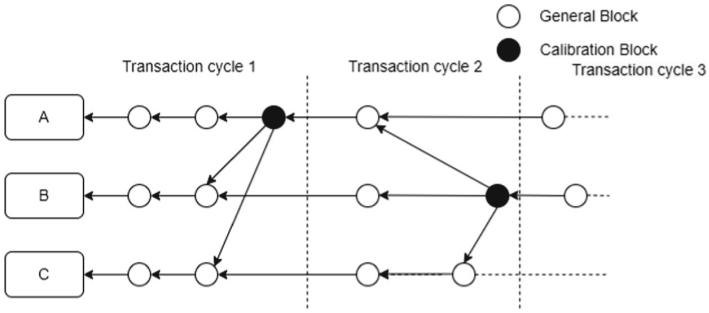

Fig. 1. Structure of UDAG

a transaction is generated, it can be directly packaged into a block and sent to the blockchain network. In order to maintain the traceability of the blockchain and quickly verify double-spending and conflicting transactions, a block must reference the latest block in the user node's history. The information contained in each block includes a timestamp, the hash of the transaction, a signature, a reference hash, and the transaction content.

3.1.2 UDAG'S Roles and Block Types

The UDAG consensus mechanism consists of two types of users: ordinary users and consensus group users. The consensus group is composed of ordinary users with the highest ranking based on their total assets, provided that their combined assets exceed half of the total assets in the system. The ranking is determined by the amount of assets held, and the highest ranked user is selected to be a consensus group user, while still retaining their status as a common user. Ordinary users are responsible for broadcasting their blocks to other nodes and storing the blocks they receive. Consensus group users are responsible for periodically reaching a consensus on a calibration block.

In UDAG, there are two types of blocks: normal blocks and calibration blocks. Normal blocks are generated by ordinary users and only contain the transaction data issued by the user and the referenced parent transactions. Calibration blocks are produced through periodic consensus of the consensus group and do not contain any valid transaction data, but serve as a reference for identifying the calibration blocks. The calibration block refers to all the leaf nodes in the local view where the consensus group nodes have reached a consensus, that is, the latest block for each user. The duration between two calibration blocks is referred to as a transaction cycle.

3.2 Consensus Algorithm for UDAG

UDAG adopts the PoS-PBFT consensus algorithm to generate calibration blocks periodically through consensus group users. The consensus group is composed of the m nodes with the highest equity, which are recalculated after each transaction cycle. The algorithm involves three roles: client, master, and slave nodes. The client sends a transaction request, which is sorted and numbered by the master node and broadcasted to the

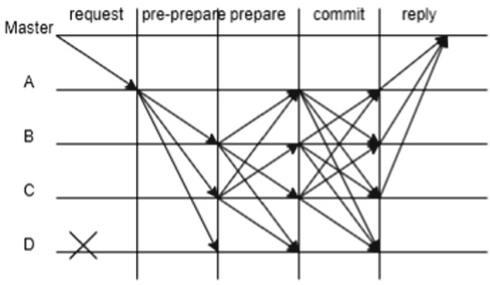

Fig. 2. PoS-PBFT consensus flow chart

slave nodes for verification and response. The node with the highest equity becomes the master node, which also acts as the client to initiate the request, and the remaining nodes in the consensus group act as slave nodes. Figure 2 shows the consensus process.

The PoS-PBFT consensus algorithm used in this mechanism has four phases: request, pre-prepare, prepare, commit, and reply. During the request phase, the client's Master node sends a request message m to the primary node A with specific format <REQUEST,o,t,master>. Here, o represents the requested operation, t is the timestamp of the request message, and master indicates that it is a request message from the client. In the pre-prepare phase, the primary node A assigns a number to the message m and broadcasts the message to the backup nodes B, C, and D with message format <<PRE-PREPARE,v,n,d>,m>, where v represents the view number, n represents the number assigned by the primary node to the request, and d represents the digest of the message m. The backup nodes validate the pre-prepare messages they have received and broadcast their validation results to other nodes in the prepare phase using message format <PREPARE,v,n,d,i>, where i is the node's own number. A node will only proceed to the next phase when it has collected more than $2f + 1$ consistent replies, including its own. After completing the corresponding operation, a node broadcasts its completion status to other nodes in the commit phase using message format <COMMIT,v,n,d,i>. Similarly, a node needs to collect more than $2f + 1$ consistent replies, including its own, to proceed to the next phase. Once a node has collected enough messages, it enters the reply phase and sends the completion message to the client using message format <REPLY,v,t,c,I,r>, where r is the node's execution result. The client only considers the request as completed when it has received responses from $f + 1$ different nodes.

4 Implementation of UDAG Consensus Mechanism Model

4.1 Overall Architecture

The overall architecture of the framework comprises three layers, namely, the foundation layer, the consensus layer, and the application layer.

The foundation layer is primarily responsible for implementing the underlying technologies in the blockchain network. This includes the parallel chain DAG structure, timestamp, distributed ledger, peer nodes, P2P network transmission, cryptographic algorithms, and other related technologies.

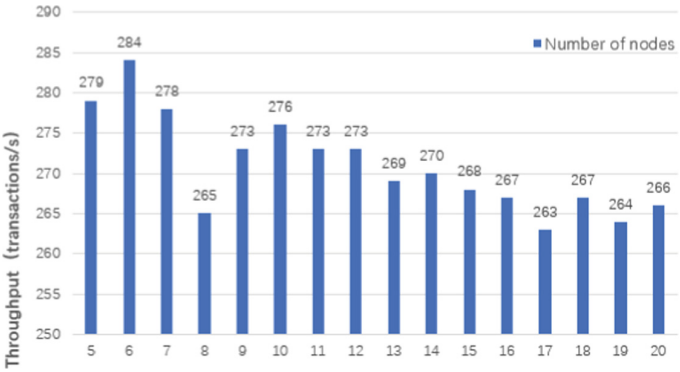

Fig. 3. Throughput of UDAG with different number of nodes

The consensus layer, on the other hand, is concerned with the confirmation of consensus group members and the PoS-PBFT consensus algorithm. The consensus group members are selected based on their total assets, with the top-ranked nodes forming the consensus group. The PoS-PBFT consensus algorithm is used to generate calibration blocks by consensus group members periodically. The consensus algorithm has three roles: client, master, and slave nodes.

Finally, the application layer serves as the technical application area of the framework, encompassing traceability, voting, trading, and other relevant areas. The framework can be used to develop various blockchain-based applications that leverage the underlying data structure and consensus algorithm.

4.2 Experimental Analysis

4.2.1 Experimental Environment

In the experimental environment, a bandwidth of 10 Mbit is used, and there are 20 nodes participating in the experiment. Each node is equipped with a single-core CPU, 2 GB of RAM, and the Windows 7 operating system. The calibration transaction generation period can be adjusted depending on the specific environment and requirements. For this experiment, the calibration transaction generation period is set to 2 min.

4.2.2 Analysis of Throughput

TPS indicates the number of transactions processed by the system per second and represents the carrying capacity of the system, the more transactions the system can handle per unit time, the higher the TPS of the system. Figure 3 shows the throughput test data of this consensus mechanism model with different number of nodes. As shown in the figure, the TPS is stable at about 265 as the number of user nodes increases.

4.2.3 Analysis of Data Consistency

UDAG adopts the Pos-PBFT consensus algorithm as its main-chain consensus mechanism, which allows for the calibration of transactions as a checkpoint, ensuring the

Table 1. Experimental test data

Model	Number of user nodes	Number of transactions issued	Number of transactions on the chain
UDAG	15	1600	1591
Fabric	15	1600	1555
UDAG	20	2000	1984
Fabric	20	2000	1936
UDAG	20	3000	2979
Fabric	20	3000	2918

consistency of the local ledger records of all user nodes within a transaction cycle subgraph. To compare the performance of UDAG with the Fabric model, experiments were conducted where the same number of outgoing transactions were set for both models, and the number of transactions verified and uploaded to the chain was recorded for different numbers of user nodes and outgoing transactions. All transactions used in the experiments were legitimate. The results are shown in Table 1.

4.2.4 Security Analysis

UDAG uses the Pos-PBFT consensus algorithm to generate calibration transactions, which can ensure a consistent transaction cycle subgraph and make the local ledger records of all user nodes consistent, as long as the consensus group is honest. Moreover, the consensus group members have more assets than other users, making them the most affected in case of an attack, thereby discouraging them from attacking the system.

The Pos-PBFT consensus algorithm is crucial in generating calibrated transactions and determining the transaction subgraph for the transaction cycle. Once the calibrated transactions are confirmed, the global transaction order becomes consistent and irreversible.

To prevent double spend attacks, UDAG requires users to refer to their previous transaction when generating a new transaction. This prevents two transactions referencing the same historical transaction and thus eliminates the possibility of block forking in the chain structure.

5 Conclusion

The paper proposes a blockchain consensus mechanism that utilizes a variant DAG structure. The asynchronous nature of the DAG structure allows for a high throughput, and the transaction organization increases the consistency of the system and strengthens the connection between transactions. This consensus mechanism can accommodate most legitimate transactions and does not require verification by subsequent transactions, which resolves the issue of uncontrollable transaction length that many DAG models encounter.

Future research could explore reducing network communication volume and node storage pressure, optimizing the matching process for various application scenarios, and finding a balance between security, centralization, and scalability.

References

1. Benčić F M, Žarko I P. Distributed ledger technology: Blockchain compared to directed acyclic graph[C]//2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2018: 1569–1570.
2. Lerner S D. DagCoin: A cryptocurrency without blocks[EB/ OL]. (2015–09–11)[2019–05–15]. https://bitslog.com/2015/09/11/dagcoin.
3. Popov S. The tangle[EB/OL]. [2019–05–15]. https: / /iota.org /IOTA—Whitepaper.Pdf.
4. Churyumov A. Byteball: A decentralized system for storage and transfer of value[EB/OL]. [2019–05–15]. https://byteball.org/Byteball.pdf.
5. Bentov I, Lee C, Mizrahi A, et al. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y[J]. ACM SIGMETRICS Performance Evaluation Review, 2014, 42(3): 34-37.
6. Castro M, Liskov B. Practical Byzantine fault tolerance[C]// Symposium on Operating Systems Design & Implementation. ACM, 1999:173–186.
7. Li C, Li P, Xu W, et al. Scaling nakamoto consensus to thousands of transactions per secon[EB]. arXiv: 1805. 03870, 2018.