2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Performance Analysis of Consensus Algorithm in Private Blockchain

Yue Hao[1], Yi Li[1], Xinghua Dong[2], Li Fang[1], and Ping Chen[1]

[1] Beijing University of Posts and Telecommunications
Beijing, China
Email: liyi@bupt.edu.cn

[2] Industrial and Commercial Bank of China
Beijing, China

Abstract— **Blockchain technology has the characteristics of decentralization, security and non-modifiable. The consensus algorithm is the core of blockchain framework, which solves the problem of mutual trust among nodes in the distributed network. Selecting the appropriate consensus algorithm or not directly affects the performance of blockchain. However, existing private blockchain platform lacks theory and data support for the performance analysis of the consensus algorithm. This paper proposes a method to evaluate the performance of consensus algorithm in private blockchain platforms of Ethereum and Hyperledger Fabric. Through quantitative analysis of latency and throughput, we obtain the performance evaluation results of consensus algorithms with different numbers of transactions. The results show that the consensus mechanism induces performance bottleneck. Besides, Practical Byzantine Fault Tolerance (PBFT) consistently outperforms Proof-of-Work (PoW) in terms of latency and throughput under varying workloads. This paper contributes to providing not only the quantitative data support for the further research of consensus algorithm, but also the guidance to the blockchain practitioners in the selection of consensus mechanism.**

I. Introduction

Blockchain technology is the underlying core technology of cryptocurrency system represented by bitcoin, and appeared for the first time in the bitcoin paper [1] published by Nakamoto. The outstanding advantages of blockchain technology are decentralization and non-modifiability. Through key technologies of peer-to-peer network, asymmetric encryption, distributed data storage, consensus mechanism and smart contract, blockchain achieves that the decentralized peer-to-peer transactions can be completed in a distributed system without mutual trust among nodes, which has solved the problems of high cost, low efficiency and unsafe storage of data in the traditional centralized system [2]. In addition, blockchain can ensure the security, trustworthiness and non-modifiability of the automatic processing of knowledge at the underlying data level, which can promote the further development of knowledge automation.

Due to the limitations of the characteristics in bitcoin, it is not suitable for some commercial application scenarios. Thus a particular type of authorized blockchain is introduced, namely private blockchain. This type of blockchain is based on a relatively closed environment where node identities are known. Through distributed consensus algorithm and smart contract, private blockchain is able to execute complex applications more efficiently than common blockchain based on Proof-of-Work, and achieve the Byzantine fault tolerance. Ethereum's private chain and Hyperledger Fabric’s alliance chain are the typical representatives of the current private blockchain.

In a distributed system, consensus mechanism can solve the problem of mutual trust among nodes in the principle of decentralization, and make the blockchain reach a balanced state in the network nodes. In the technology architecture of blockchain, the consensus algorithm, as the core mechanism of the blockchain, has a direct impact on the overall performance of the blockchain. This paper focuses on the performance analysis of consensus algorithm in private blockchain platforms. This research is of great importance for researchers to further optimize the consensus algorithm, as well as the blockchain developers to understand the limitations of the platforms and make a reasonable choice.

Previous works on performance of blockchain consensus protocol mainly focuses on the research and analysis of Proof-of-Work in the public blockchain. A quantitative framework has been proposed in [3] to analyze the security and performance implications of various consensus and network parameters of Proof-of-Work blockchain. Another study in [4] use time to consensus to measure the performance of blockchain.

In the performance research of private blockchain, a general framework has been proposed in [5], where the evaluation results under varying number of nodes are assessed through metrics of latency and throughput. Additionally, a more recent study in [6] focuses on varying number of transactions, which is considered to complement the findings in [5]. However, consensus protocols are not taken into account in the analysis presented in [6] where evaluation experiment is completed in the network environment of a single node. In this paper, the analysis is focused on the the consensus algorithm in private blockchain, including the performance analysis of consensus algorithms and the impact of varying number of transactions on the performance, which is expected to complement and extend the findings in [5] and [6] .

The contributions of this paper are summarized as follows. First, a method for evaluating the performance of consensus algorithms in private blockchains is proposed. This evaluation targets on the consensus algorithms in Ethereum and Hyperledger and the results are assessed through quantitative analysis of latency and throughput. Secondly, the performance of consensus algorithm is evaluated under different number of transactions which is up to 10000. Third, the results and implications of performance evaluation are analyzed and discussed. In brief, this paper provides data support and theoretical basis for the research on consensus algorithms in private blockchains, which is of great significance for further research and optimization of consensus algorithm. Besides, it is also very important for the blockchain practitioners to comprehend the characteristics of the private blockchain platform and choose the appropriate consensus algorithm.

This paper is organized as follows. Section II gives a brief overview of the target blockchain platforms and the consensus algorithms of this performance evaluation. In Section III, the method for evaluating consensus algorithms in the private blockchains is presented. Then, Section IV discusses the results, implications and limitations of this experiment. Finally, Section V concludes this paper.

II. Private Blockchain & Consensus Algorithm

This section briefly introduces the target blockchain platforms and the corresponding consensus algorithms, including Proof-of-Work variants in Ethereum and PBFT in Hyperledger Fabric. Two consensus algorithms above are the representative of current common consensus algorithms in the private blockchains.

A. Ethereum and PoW

Ethereum [7] is an open-source, public, blockchain-based distributed computing platform featuring smart contract functionality. It provides a decentralized turing-complete virtual machine, the Ethereum Virtual Machine (EVM), which can execute scripts using an international network of public nodes. Ethereum also provides a cryptocurrency token called "ether", which can be transferred between accounts and used to compensate participant nodes for computations performed.

PoW（Proof-of-Work）is the consensus algorithm which is used in bitcoin at first. Its core mechanism is the competition among the nodes to calculate a hash value of the block header which will be equal or smaller than a target value. One node would broadcast the block to other nodes once it reaches the target value and all other nodes must mutually confirm the correctness of the hash value [8]. If the block is validated, other miners would append this new block to their own blockchain. The consensus mechanism used in Ethereum is a variant of PoW, which has the same principles as PoW.

B. Hyperledger Fabric and PBFT

Hyperledger Fabric [9] is an implementation of permissioned blockchain technology that is intended as a foundation for developing blockchain applications or solutions with modular architecture. Hyperledger Fabric allows components, including consensus and membership services, to be plug-and-play. Specially, chaincodes [5] are deployed as Docker images interacting with Hyperledger's backend via pre-defined interfaces.

The consensus algorithm adapted in Hyperledger is PBFT which could handle up to 1/3 malicious byzantine replicas [8].The process of PBFT algorithm is divided into three phrase: pre-prepare, prepare and commit [10]. Fig. 1 shows the algorithm flow. The pre-prepare and prepare phases are used to totally order requests sent in the same view even

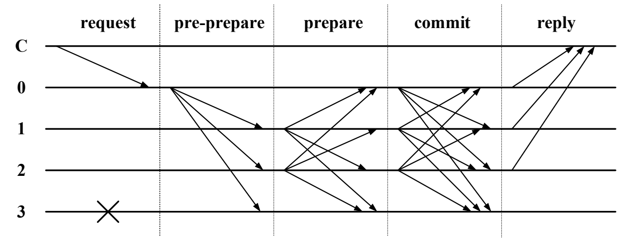

Figure 1. PBFT algorithm flow

when the primary, which proposes the ordering of requests, is faulty. The prepare and commit phases are used to ensure that requests that commit are totally ordered across views. In each phase, a node would enter next phase if it has received votes from over 2/3 of all nodes. The PBFT algorithm provides the fault tolerance of (n-1)/3 under the premise of ensuring activity and safety.

III. Methodology

This section mainly introduces the method used to evaluate the performance of consensus algorithms in private blockchains. Firstly, the abstraction layers of blockchain are introduced. Then we propose the evaluation framework and introduce infrastructure setup of the experiment, which is followed by the implementation and workload of the evaluation. Finally, the evaluation metrics are illustrated.

A. Blockchain Layers

According to the framework in [5], we divided blockchain into four abstraction layers: consensus layer, data model layer, execution layer and application layer. The consensus layer includes consensus protocol, through which nodes reach consensus and new blocks are created. The data layer includes the data structures, block contents, and transaction operations of blockchain. Besides, the execution layer includes details of the runtime environment which support the blockchain operations), which mainly contains the deployment of EVM and its smart contract in Ethereum, and the operating environment for chaincode in Hyperledger. And, the application layer includes a variety of blockchain applications such as DAO and Parity. This paper focuses on the consensus layer and evaluates the performance of consensus algorithms in the private blockchains. The evaluation architecture and experimental methods are introduced below.

B. Evaluation Architecture and Infrastructure Setup

Fig. 2 shows the architecture of the evaluation. In the performance evaluation framework of consensus algorithm constructed in this paper, there are four main modules: workload configuration module, consensus smart contract module, data collector module and the target blockchain platforms. The first three modules are integrated in the Yahoo Cloud System Benchmark (YCSB) testing tools. Specifically, the consensus contract module is mainly used to evaluate the performance of the consensus algorithm. The workload configuration module is used to set different workloads. And the data collection module collects the transaction data. Besides, client and driver serve as a connection between different modules, together to construct the entire evaluation architecture. In this evaluation, the two target blockchain platforms are Ethereum and Hyperledger Fabric.

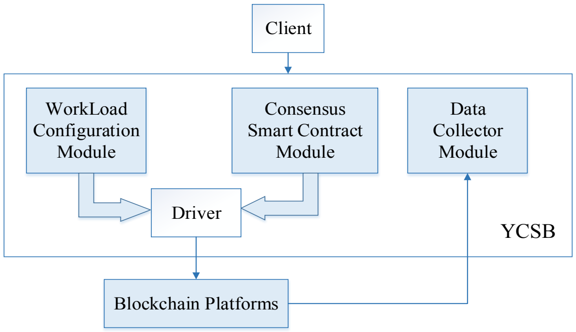

Figure 2. Evaluation architecture.

The experimental infrastructure is four servers, each with 8GB of RAM, 128G SSD hard drive and running Ubuntu16.04, and four clients to send transactions.

1) Ethereum: the method to build private blockchain of Ethereum is to, firstly download and install the latest version of Ethereum (geth1.7.3 [11]) on four servers in the same LAN, and deploy four nodes separately, then specify the same networkid and genesis.json for all nodes to ensure that nodes can be properly connected with each other, finally complete the connection between the main node and child node through method of the file configuration [12].

2) Hyperledger Fabric: the method to build private blockchain of Hyperledger Fabric is also to, firstly download and install the Hyperledger Fabric 1.0 [13] on the server, then configure the fabric environment, and finally deploy the fabric network [14] (which can be built in a stand-alone or multi-machine network) and test the chaincode.

C. Implementation and Workloads

The experimental process of this paper is mainly divided into two parts. The first part is the performance test of the consensus algorithm in private blockchains. The second part analyzes the impact of different transaction quantities on the performance of the consensus algorithm by configuring the workload.

1) Consensus Smart Contract: Part A has introduced the abstraction layers of blockchain. In order to assess the impact of the consensus layer separately, we constructed a consensus smart contract—DoConsensus Smart Contract. This DoConsensus contract minimizes the logical function, for the purpose of highlighting the impact of the consensus layer, which receives a transaction just as input and returns the transaction immediately. Therefore, the operations involved in the execution layer and data layer are minimized, and thus the overall performance of the blockchain will depend mainly on the consensus layer. Here a python script named exp.py is constructed to start the experiment. Smart contracts have been written and deployed in Ethereum and Hyperledger Fabric before the experiment, and we need to specify the values for configuration variables, then run the python script. The code snippet of the python script is shown in Fig. 3. In order to evaluate the performance of the consensus algorithm comprehensively, two set of comparative experiments are conducted in this section. With the same number of transactions (10000), the same number of servers (4) and the same number of clients (4), clients send the transaction requests to the blockchain. The consensus mechanism is

```
# python exps.py [-doconsensus]
if __name__=='__main__':
  error_msg = 'python exps.py [-doconsensus] \n'\
  ...
      + '    -doconsensus:  doconsensus benchmark\n'
  opts = ['-doconsensus']
  ...
# doconsensus option
if  len(sys.argv)>1 and sys.argv[1]=='-doconsensus':
  time.sleep(10)
  cp_cmd = 'cp env_doconsensus.sh env.sh'
  os.system(cp_cmd)
  change_config(1)
...
```

Figure 3. A code snippet from exp.py, written in python

closed by configuration in one set of experiments, while th consensus contract is deployed in the other set of experiments. Each experimental result is averaged in ten independent runs, and horizontal and vertical evaluation analysis are conducted accordingly.

2) Evaluation Workload: In addition to the performance evaluation of the consensus algorithm itself, there is another important content in this paper to explore the effect of different transaction volumes on the performance of consensus algorithm, which is included in the workload configuration module. The experiment conducted is that, with the same number of servers (4) and the same number of clients (4), clients send N transaction requests to the target blockchain platform, with the number of transaction requests set to 1, 10, 100, 1000 and 10000. Similarly, each experimental result is averaged in ten independent runs to assess the impact of the transaction quantities on the performance of consensus algorithms.

In addition, the client and the blockchain platform complete the communication through the HTTP request. In Ethereum, queries are done via the JSON-RPC APIs, which can return details and history of transaction. For Hyperledger Fabric, queries are implemented via chaincode that can access the ledger' s state.

D. Evaluation Metrics

Through running workloads of different configurations, we aggregate the statistics obtained from data collector integrated in YCSB into two performance evaluation metrics to evaluate the consensus algorithm:

Latency *Latency* indicates the reaction time of each transaction, that is, the difference between the time (t1) when the transaction is confirmed by the blockchain and the unix time (t2) when the transaction is deployed. The final metric we use to evaluate was the average latency of all the trasactions in ten independent runs for each group of experiments.

Throughput *Throughput* refers to the number of successful transactions per second. This indicator can directly reflect the transaction processing capabilities of blockchain platform, which indirectly reflects the impact of consensus algorithm on blockchain performance.

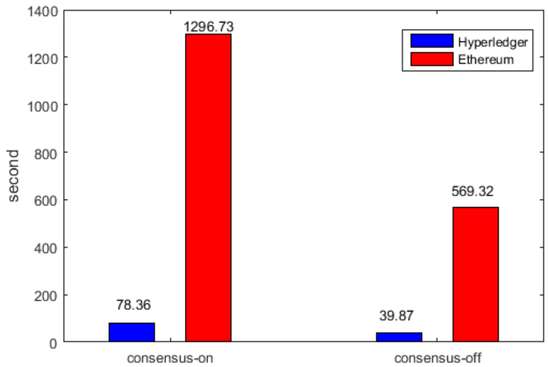

Figure 4. Average latency of Ethereum and Hyperledger with different consensus settings that is on and off

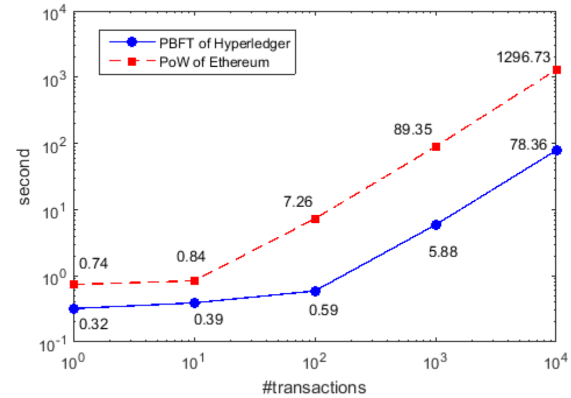

Figure 6. Average Latency of Ethereum and Hyperledger with varying number of transactions

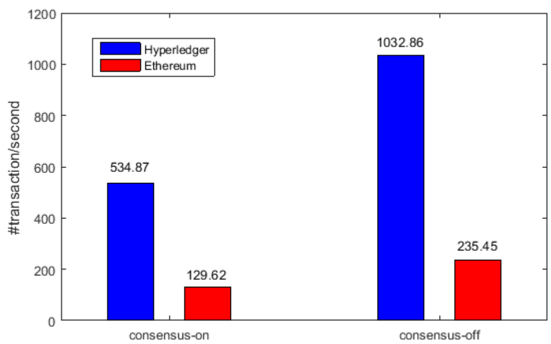

Figure 5. Average throughput of Ethereum and Hyperledger with different consensus settings that is on and off

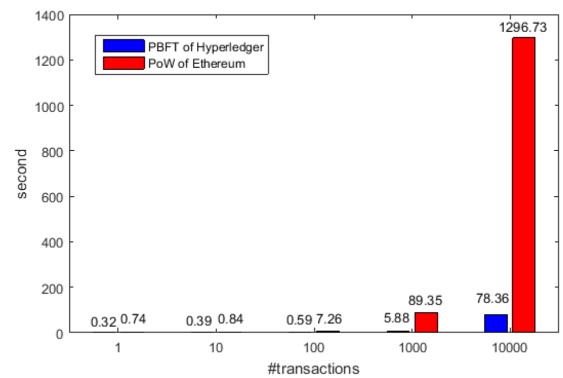

Figure 7. Comparison of average latency between Ethereum and Hyperledger

IV. Results and Discussion

A. *Performance Evaluation*

In this part, we compare the average latency and the average throughput to evaluate the performance of the blockchain under two experimental conditions where the consensus mechanism is deployed and closed.

Comparison of Average Latency In Fig. 4, we explore the average latency performance of Ethereum and Hyperledger Fabric under conditions of the consensus mechanism is deployed and closed, with the same number of transactions, servers and clients. First, in each blockchain platform, the deployment of consensus mechanism has shown the characteristic of the increasing latency compared to the closed situation. Among them, the average latency of Ethereum increases by 1.28 times, which is up to 1296.73s in the case of consensus-on, and Hyperledger Fabric's increases by 0.97 times. Besides, when the consensus mechanism is deployed, the average latency of Ethereum is 16.55 times that of Hyperledger Fabric. Different consensus algorithms make the average latency of Hyperledger much lower than that of Ethereum.

Comparison of Average Throughput In Fig. 5, we discuss the average throughput performance of Ethereum and Hyperledger Fabric on the case of the consensus mechanism is deployed and closed under the same experimental conditions as above. The deployment of consensus mechanisms has resulted in a decrease in average throughput on both platforms, compared to the off situation. In addition, under normal deployment, the average throughput of Hyperledger Fabric is 534.87 tx / s, which is about 4.13 times that of Ethereum that is up to 129.62 tx / s.

B. *Varying Workloads*

In this part, we still use the above two evaluation metrics: average latency and average throughput, to evaluate the performance of consensus algorithms in private blockchain platforms under different numbers of transactions.

Comparison of Average Latency Fig. 6 shows the log plot of the average latency of consensus algorithms in Ethereum and Hyperledger Fabric with different numbers of transactions. In the case of a small number of transactions, the latency of both algorithms is relatively small and PoW is about twice as that of PBFT. After 100 transactions, the average latency of PoW increased rapidly compared with PBFT. When dealing 10,000 transactions, the average latency of PoW reaches 1296.73s, which is 16.55 times of PBFT that is 78.36s. Fig. 7 shows a histogram of the mean latency between PoW and PBFT under different numbers of transactions. Obviously, as the number of transactions increases, the average latency for both algorithms has increased significantly, and the differences between them have also grown significantly. Especially for Ethereum, the average latency of PoW reaches 14.51 times when the number of trancastions increases from 1,000 to 10,000. On the whole, PoW outperforms PBFT in average latency.

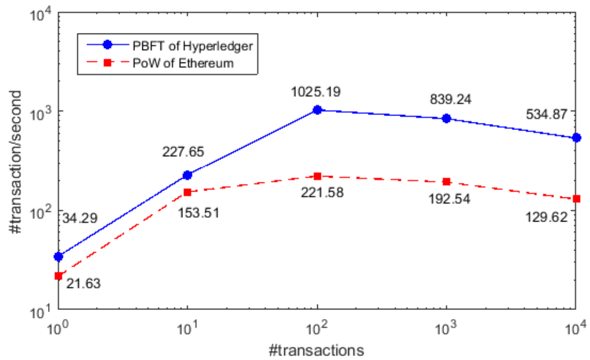

Figure 8. Average Throughput of Ethereum and Hyperledger with varying number of transactions

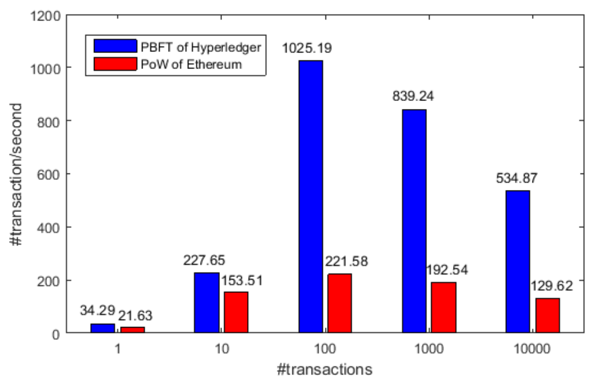

Figure 9. Comparison of average throughput between Ethereum and Hyperledger

Comparison of Average Throughput Fig. 8 shows the log graph of the average throughput of consensus algorithms in Ethereum and Hyperledger Fabric under different numbers of transactions. The average throughput of both algorithms increases as the number of transactions increases from 1 to 100. Under condition of 100 transactions the throughput of both algorithms was maximized, of which PBFT reaches 1025.19 tx / s, about 4.6 times of PoW. After 100 transactions, with the increase in the number of transactions, the average throughput of two algorithms decreases. Fig. 9 shows a histogram of the average throughput for PoW and PBFT with different numbers of transactions. The average throughput of PBFT is always higher than that of PoW, and the variation of Hyperledger Fabric's grows larger as the number of transactions increases.

C. Implications

The results of this paper show that for consensus algorithms, the performance of PBFT algorithm in Hyperledger Fabric is superior to that in Ethereum, both in terms of latency and throughput. Besides, through data analysis, the existence of the consensus mechanism will bring increasing delay and decreasing throughput to some extent, that is, performance bottleneck. This discovery is established both in Ethereum and Hyperledger Fabric. In essence, the consensus mechanism in the blockchain is to exchange performance for security and fault tolerance.

In addition, the analysis of this paper also shows that PBFT of Hyperledger Fabric always outperforms PoW of Ethereum in terms of both latency and throughput under different transaction volumes. With the increase in the number of transactions, PoW's latency sharply increases, while the throughput of PBFT changes significantly. Since the experimental condition is established to involve the minimum operations in the executive layer and the data layer, the overall performance of the blockchain is decided mainly by the consensus mechanism. That is to say, the performance of two blockchain platforms in the results directly represents the performance of the two consensus algorithms under different transaction volumes. These results complement [6] and solve the experimental limitations in [6].

The research results show that the consensus algorithm directly affects the performance of private blockchains. The selection of consensus algorithms, or the selection of different private blockchain platforms, will directly determine the throughput and latency of blockchain in the subsequent development process. In addition, the number of transactions is also very crucial to the selection of blockchain platforms, and to a certain extent, it will affect the subsequent performance indicators.

D. Limitations

This research of this paper focuses on the performance analysis of the consensus algorithms in private blockchains. Due to the design of the blockchain itself, there is a certain coupling between the executive layer, the data layer and the consensus layer, which can not be independent and extended completely. Therefore, the core of this paper is to design a consensus smart contract which involves the minimum operations of the execution layer and the data layer, in order to maximize the impact of the consensus layer and then analyze the performance of the blockchain. With the continuous development of blockchain, the decoupling between abstraction layers may perform better. We will further explore the impact of a completely decoupled consensus layer on the overall performance of the blockchain in future work.

In addition, the experiment in this paper evaluates the performance of the consensus algorithm under cases of four servers. From the research results in [5], Hyperledger Fabric will stop working when it exceeds 16 nodes. Therefore, under cases of different number of nodes, how the number of transactions affects the performance of consensus algorithms in two blockchain platforms needs further study.

V. Conclusion

This paper assesses the performance of consensus algorithms in the private blockchain platforms of Ethereum and Hyperledger Fabric. The results show that Hyperledger's PBFT algorithm outperforms Ethereum's PoW algorithm in terms of average throughput and average delay. In addition, both in Ethereum and Hyperledger Fabric, the existence of consensus mechanism causes performance bottlenecks in both platforms, due to the nature of exchanging security for performance. Another study result indicates that the performance difference of the two consensus algorithms increases as the number of transactions increases. But PBFT of Hyperledger Fabric consistently has higher throughput and lower latency. The study of this paper is important in two aspects. Theoretically, it provides theory and data support for the research of consensus algorithm in private blockchains, and is of great significance for further research and algorithm optimization of consensus mechanism in the future. In the application aspect, the research results can help blockchain practitioners understand the limitations of blockchain platforms deeply and provide guidance for the selection of consensus mechanism. In future work, we will evaluate the performance of a better decoupled consensus layer in the private blockchain. Another direction that deserves further study is the effect of the number of transactions on the performance of the consensus algorithm under case of different numbers of nodes. In addition, we are also interested in evaluating more blockchain consensus algorithms, such as PoS and DPoS, and explore their performance to impove our evaluation methodology.

REFERENCES

[1] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system[J]. Consulted, 2008.

[2] Y. Yuan, F. Y. Wang, Blockchain: The State of the Art and Future Trends[J]. Acta Automatica Sinica, 2016.

[3] A. Gervais, G. O. Karame, V. Glykantzis, et al, On the Security and Performance of Proof of Work Blockchains[C]// ACM Sigsac Conference on Computer and Communications Security. ACM, 2016:3-16.

[4] E. Kokoriskogias, P. Jovanovic, N. Gailly, et al, Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing[J]. Applied Mathematical Modelling, 2016, 37(8):5723-5742.

[5] T. T. A. Dinh, J. Wang, G. Chen, et al, BLOCKBENCH: A Framework for Analyzing Private Blockchains[J]. 2017.

[6] S. Pongnumkul, C. Siripanpornchana, S. Thajchayapong, et al, Performance Analysis of Private Blockchain Platforms in Varying Workloads[C]// International Conference on Computer Communication and Networks. 2017:1-6.

[7] "Ethereum," [Online]. Available: https://en.wikipedia.org/wiki/Ethereum/

[8] Z. Zheng, S. Xie, H. Dai, et al, An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends[C]// IEEE International Congress on Big Data. IEEE, 2017.

[9] "Hyperledger blockchain technologies for business," 2017. [Online]. Available: https://www.hyperledger.org/

[10] M. Castro, Practical byzantine fault tolerance and proactive recovery[C]// Symposium on Operating Systems Design and Implementation. USENIX Association, 1999:173-186.

[11] "Download Geth-Weir(v1.7.3), " 2017. [Online]. Available: https://geth.ethereum.org/downloads/

[12] "Ethereum Frontier Guide," [Online]. Available: https://ethereum.gitbooks.io/frontier-guide/content/connecting.html

[13] "hyperledger/fabric at v1.0," 2017. [Online]. Available: https://github.com/hyperledger/fabric

[14] "Building Your First Network," [Online]. Available: http://hyperledger-fabric.readthedocs.io/en/release/build_network.html