SCALABILITY ASSESSMENT AND PERFORMANCE OPTIMIZATION OF HYPERLEDGER FABRIC

Cho Cho Htet

Department of Computer Engineering and Information Technology, Mandalay Technological University, Mandalay, Myanmar
karyancc2012@gmail.com

Aye Myat Thu

Department of Computer Engineering and Information Technology, Mandalay Technological University, Mandalay, Myanmar
ayemyatt2@gmail.com

Abstract

The feature of Blockchain as distributed ledger that are shared among nodes within a computer network, renowned for its pivotal role in cryptocurrency systems by ensuring a secure and decentralized the role of transaction record which is ensured for maintenance of security and decentralization in cryptocurrency systems. The Linux Foundation host the open-source framework of private blockchain, the Hyperledger Fabric (HLF). Smart contracts are utilized for transaction management and a modular architecture of blockchain framework, providing a foundation for the development of blockchain-based applications through plug-and-play components. In the realm of distributed systems, scalability emerges as a crucial design goal for developers. The most appropriate blockchain platform for the operations of the business industry, which need for the seamless addition of more users and resources without perceptible performance loss. An assessment of scalability is required as a large number of nodes involvement in the implementation of blockchain frameworks. In this paper, the impact of system configurations such as, transaction volume, node types is focused in the transition of V2.2.4 with the various significant issues with the architecture. The throughput, latency, processor, and memory usages are mainly analyzed based on the different number of transactions. According to the performance results of the proposed system, the scalability of the possible number of transactions and the different peer nodes can be supported in the implementation of blockchain-based system for HLF blockchain.

***Keywords*: Blockchain, Hyperledger Fabric; plug-and-play.**

1. Introduction

Modern transactional networks are essentially evolved versions of age-old networks that have been integral to business record-keeping throughout history. In these networks, business members engage in transactions, maintaining individual records of their interactions. The security of these transactions is paramount, ensuring that businesses selling items can establish a clear chain of ownership [13]. Over time, technology has transformed this process from stone tablets and paper folders to the current era of hard drives and cloud platforms. The exchange of information and processes has evolved to create a robust system of record that spans a business network, meeting the distinct requirements of visibility and trust. [6]

A blockchain network is a distributed ledger that records all transactions occurring within its domain. This ledger's distinct feature lies in its decentralization. It is replicated across numerous network participants, fostering collaborative maintenance [12]. This decentralized and collaborative nature emulates the real-world exchange of goods and services within businesses. Smart contracts play a pivotal role in this network, facilitating controlled access to the ledger. These contracts serve as a powerful mechanism, encapsulating information and simplifying its dissemination across the network. Smart contracts can be designed to automate specific transactional aspects, ensuring consistent information updates and enabling a many of ledger functions, from implementation to querying. [4]

Hyperledger Fabric stands as a distributed ledger platform, underpinned by a modular architecture that offers a robust foundation for various solutions. It ensures the levels of confidentiality, resiliency, flexibility, and scalability. It supports pluggable implementations of diverse components, seamlessly adapting to the intricate and dynamic nature of the economic ecosystem. [8]

This literature is focused on the scalability of the possible number of transactions and the different peer nodes in the Hyperledger Fabric blockchain. The Go language is used for the smart contract function of the chaincode. The LevelDB is used as a state database in Fabric blockchain. The paper is prepared as follows: Section II designates the related work for this proposed system, Section III describes the background theory of the system. The evaluation procedure is shown in Section IV. Section V confers the result and discussion of the proposed system.

2. Related Works

T. T. A. Dinh and et al. [1] concentrated on private blockchains. They used BLOCKBENCH to calculate three major blockchain systems: Ethereum, Parity, and Hyperledger Fabric. In terms of data write and read throughput, Hyperledger Fabric outperforms Ethereum and Parity. Hyperledger Fabric uses less disk space and has a lower latency than Ethereum and Parity. Hyperledger (v0.6) has a higher data write, read and disk use throughput than Hyperledger (v0.5) and (v1.0). In comparison to Hyperledger (v1.0), the nodes in Hyperledger (v1.0) conduct three extra phases to complete a transaction (v0.6).

P. Yuan and et al. [7] created a model employing Generalized Stochastic Petri Nets (GSPN). It was crafted to assess the performance of a Hyperledger-based system. This model dissects transactions into phases, utilizing a simulation-based method to compute system delay and throughput with precise arrival rates. The focus extended to the influence of different ordering service settings on system performance. Introducing a mathematical configuration gathering approach, they aimed to identify optimal ordering service configuration parameters for enhancing system performance. Validation of the proposed model and methodologies was conducted through experiments on a real-time system. System issues manifested during the transition from committing to endorsing phases, especially noticeable with an increase in transactions per block. The study enhanced in on optimizing performance during the endorsement phase for overall system enhancement in the future.

H. Yusuf and I. Surjandari [8] compared the performance of a blockchain network, featuring Kafka and Raft as implementations for the ordering service node in Hyperledger Fabric. The blockchain network was structured with three distinct channels, each housing two organizations and two peer nodes. Raft boasted five ordering service nodes, while Kafka featured three, with each dedicated to a specific channel. Employing a straightforward chaincode for the process, the chaincode timeout was set at 100 seconds to specifically explore the capabilities of Raft and Kafka. The performance assessment, conducted using Hyperledger caliper. It is a tool designed for measuring blockchain network transaction capabilities. Raft outperforms Kafka in terms of transaction completion. Simulation results indicated that Raft excels over Kafka in terms of maximum, minimum, and average throughput, showcasing its faster transaction processing capabilities.

A. Faroug and et al. [2] introduced a blockchain-based application for safeguarding and sharing medical data and vaccination records. This initiative harnessed the capabilities of both Hyperledger Fabric and Ethereum platforms. The assurance of record integrity and security was validated through the use of Ganache in a secure and predictable environment. The performance evaluation utilized Hyperledger Caliper and Explorer, showcasing the efficiency and transparency enhancements in record tracking. The implemented approach not only provides a secure avenue for updating, sharing, and validating records but also emerges as a valuable asset amid the ongoing challenges posed by the COVID-19 virus.

K.S.S. Wai and et al. [5] explored key configuration factors like batch timeout and block size. The analysis revealed their significant role in shaping an optimized blockchain-based storage system. Investigating into transaction throughput, average latency, and block height, the study examined the influence of adjusting batch timeout and block sizes across varying numbers of concurrent users within an organization. Organizations with a higher user count would benefit from considering larger values for batch timeout and block size. These research findings serve as a valuable guide for implementing an optimal blockchain-based storage system on HLF.

Q. Nasir and et al. [4] analyzed the comprehensive performance into the complexities of two iterations of Hyperledger Fabric, v0.6 and v1.0. The assessment covered a spectrum of parameters, including execution time, latency, and throughput, with a dynamic workload ranging up to 10,000 transactions on each platform. They focused on scalability and adjusting the number of nodes to 20 in each platform. The evaluation metrics revealed the superiority of Hyperledger Fabric v1.0 over v0.6 in terms of scalability, throughput, execution time, and latency. Despite this progress, it's worth noting that Hyperledger Fabric v1.0 hasn't quite reached the performance levels of current traditional database systems, particularly under high workload scenarios.

The orderer section's block timeout and block size parameters were examined by C.C. Htet and et al. [3] in order to identify the best settings that satisfy the user's system needs. By using the block timeout and block size, they looked at how many block heights there were in the blockchain. In the Hyperledger Fabric (HLF) blockchain, it is advised to choose a preferred maximum block size. They also discussed how a tiny block size can cause system congestion and an overly short block timeout.

3. Background Theory

Each project within the Hyperledger initiative is progressing through distinct stages of development. The anticipation of continual growth as more members actively engage in contributing to the advancement of blockchain technology. The overarching goal of Hyperledger is to construct innovative blockchain platforms that are tailor-made for specific industry use cases. The substantial contributions from the community have driven Hyperledger towards evolving into not just a blockchain platform but a protocol for conducting business transactions. This evolution extends further into becoming a comprehensive specification, serving as a valuable reference for constructing diverse blockchain platforms. This stands in contrast to earlier blockchain solutions that were confined to addressing specific industries or requirements. [7] [15]

A published reference architecture model by the Hyperledger project is provided in this section. A blockchain developer can utilize this design to create a blockchain that complies with the Hyperledger architecture's requirements. A whitepaper from Hyperledger outlines a reference architecture model that can be used as a blueprint for creating private distributed system. A business blockchain can be constructed by the several components that make up the reference architecture. [2] [11]

The five uppermost components offer diverse services. Firstly, the identity which offers membership services such as authorization, identification, and authentication. Next is the policy component, which offers services related to policy. Blockchain and transactions, which include Distributed Ledger, P2P Protocol, Consensus Manager, and Ledger Storage services. The consensus manager makes sure that updates to the ledger may only be made by agreement between blockchain network users. The last layer is the smart contracts layer, which uses Secure Container technology to host smart contracts and offers chaincode services in Hyperledger. Chaincode can be thought of as smart contracts' Hyperledger complement. From the standpoint of components, Hyperledger is made up of the following elements: [8]

- Consensus: These services are in charge of assisting the blockchain network's participants in reaching agreements with one another. It is necessary to be verified and agreed in order for the order and state of transactions in the blockchain network.
- Smart contracts: It is used for carrying out business logic in accordance with user specifications. The logic outline in the smart contracts stored on the blockchain is used to process transactions.
- Communication: Message exchange and transmission between blockchain network nodes are handled by this component.
- Security and crypto: This is the process of giving different cryptographic algorithms or modules the capacity to offer non-repudiation, privacy, and confidentiality services.
- Data store: This service allows the ledger's current state to be stored in various data stores. This implies that any database backend, like couchdb or goleveldb, can be used with data stores since they are also pluggable.
- Policy services: These services offer the management of various policies necessary for the blockchain network. Both consensus and endorsement policies are under this category.
- APIs and SDKs: Clients and applications can communicate with the blockchain through this component. Block queries, chaincode deployment and execution, and blockchain event monitoring are all made possible by SDKs. [5]

4. Evaluations

Caliper is a framework for blockchain benchmarking. It may be applied to any blockchain to assess its performance. Caliper supports the following blockchains: Ethereum, Fabric, Sawtooth, Burrow, and Iroha. A service called Caliper creates a workload for a particular system under test (SUT) and keeps track of how it responds over time [10]. Lastly, a report is produced by Caliper using the SUT responses that were observed. [5]

The primary content consists of the network configuration, benchmark configuration, and workload module. A benchmark's workload modules are its brains. Therefore, Caliper is a generic benchmarking system. After Caliper schedules transactions (TXs) for a particular round, the workload module for that round is responsible for creating and submitting the TX content. There may be a distinct task module linked with each round. The benchmark configuration defines the number of transaction load. The Hyperledger Fabric network certificate is configured in the Caliper network configuration. [9]

The framework supports a variety of performance indicators. Success rate, transaction read rate, throughput, latency, and the usage of CPU, memory, and I/O hardware resources used are a few of these [14]. The analysis of throughput, latency, and hardware resource consumption—including CPU and memory—takes place in this article. The evaluation of throughput is calculated by dividing the total number of successful and unsuccessful transactions by the time difference between the beginning and end of the network. The transaction per second unit (TPS) is used to express the throughput. The maximum latency is calculated by taking the maximum latency of all successful transactions. The second unit (seconds) is a polite measure of latency. In the Hyperledger blockchain network, the total allotment of peer nodes, orderer nodes, and installed chaincodes determines the hardware resource consumption, including CPU and memory. [7] [16]

5. Result and Discussion

In this paper, the different number of transactions and peer nodes are analyzed in the HLF blockchain. The transaction arrival rates are changed depend on the organization. The transactions are replicated in every node in blockchain network. The number of transaction and peer nodes affect the presentation of blockchain-based system. The scalability is one of the main factor in the distributed system. Healthcare Dataset (HD) is used to analyse the performance on the HLF blockchain. The patient id is defined as a ‘key’. caseNumber, treatmentDate, statWeight, stratum, age, sex, race, diagnosis, bodyPart, disposition, location and product are defined as the ‘values’ for PHR in the measurement of the performance.

Caliper is used to measure the performance of HLF framework such as, throughput, latency, processor and memory. The parameters for the different number of transactions are presented in Table 1. The number of transactions (10, 100, 1000 and 10000) are analyzed with the 2 peer nodes. Transaction throughput, latency, processor and memory usage are measured in HLF blockchain.

Parameters	**Values**
Number of transactions	10, 100, 1000 and 10000
Number of peers	2

Table 1. The parameters for different number of transactions.

In Fig. 1, the throughput of the different transactions are shown for the invoke function. If the number of transactions are higher from 10 to 1000, the amount of throughput will be increased. But, if the number of transactions are 10000, the amount of throughput will be marginally reduced. Therefore, the transactions 1000 is the highest throughput in 2 peer nodes according the performance results.

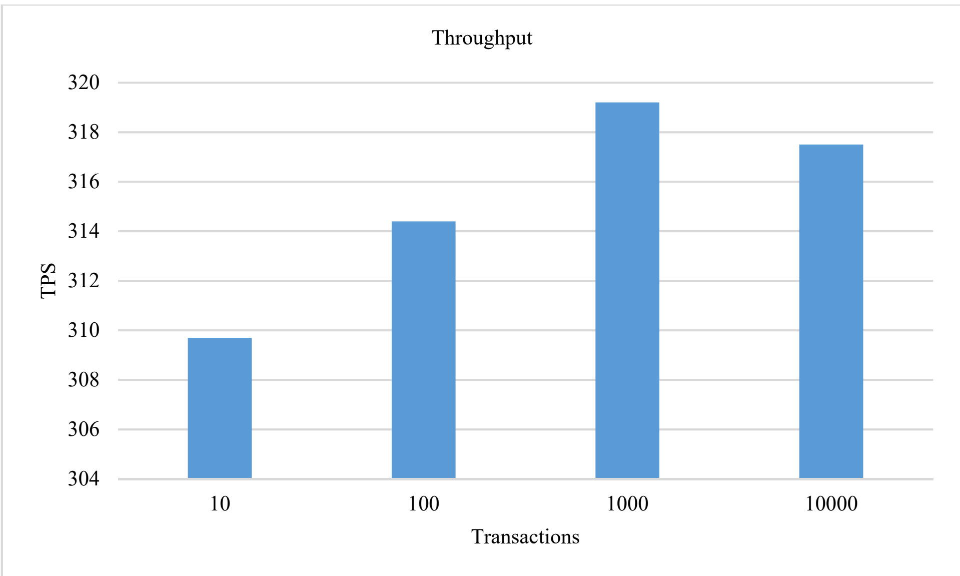

Fig. 1. Throughput for different number of transactions

According to the experimental results, the maximum latency of the different number of transactions are shown in Fig. 2 for the invoke function. If the number of transactions are greater, the maximum latency for execution will be increased. Therefore, the latency is affected by the arrival rate of the number of transactions.

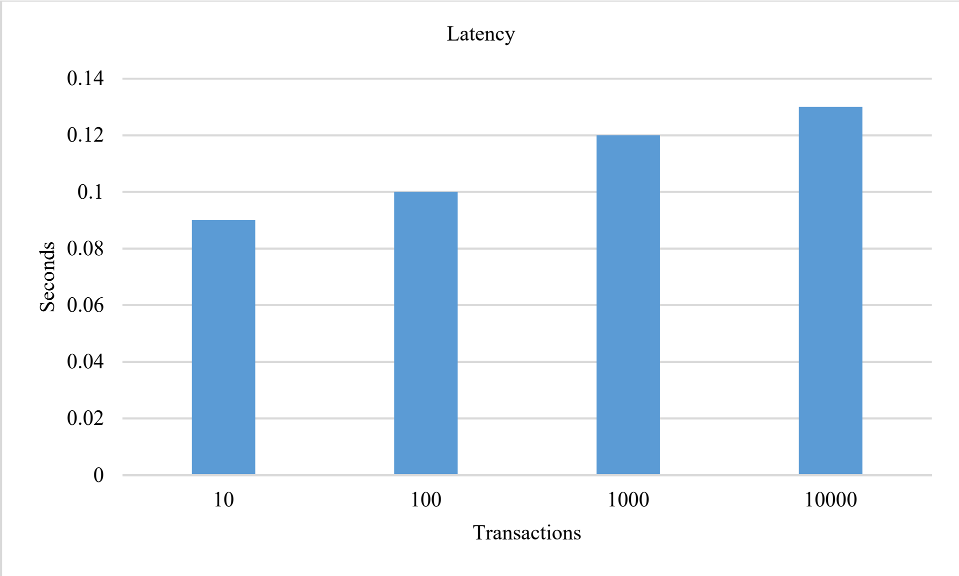

Fig. 2. Latency for different number of transactions

The usage of processor for the different number of transactions are shown in Fig. 3 for the invoke function. If the number of transactions are larger, the percentage of processor usage for execution will also be improved. Therefore, the processor usage is also affected by the arrival rate of the number of transactions.

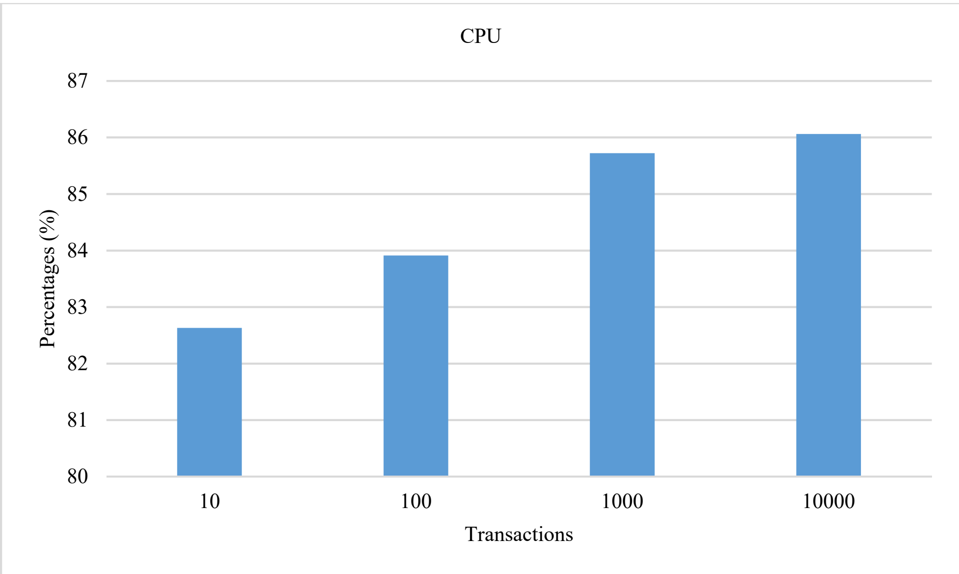

Fig. 3. Processor usage for different number of transactions

The usage of memory for the different number of transactions are shown in Fig. 4 for the invoke function. The memory usage is obviously increased among the number of transactions (10, 100 and 1000). But, the memory usage is slightly increased between the number of transactions (1000 and 10000). Generally, the percentage of memory usage for execution will also be increased if the number of transactions are higher.

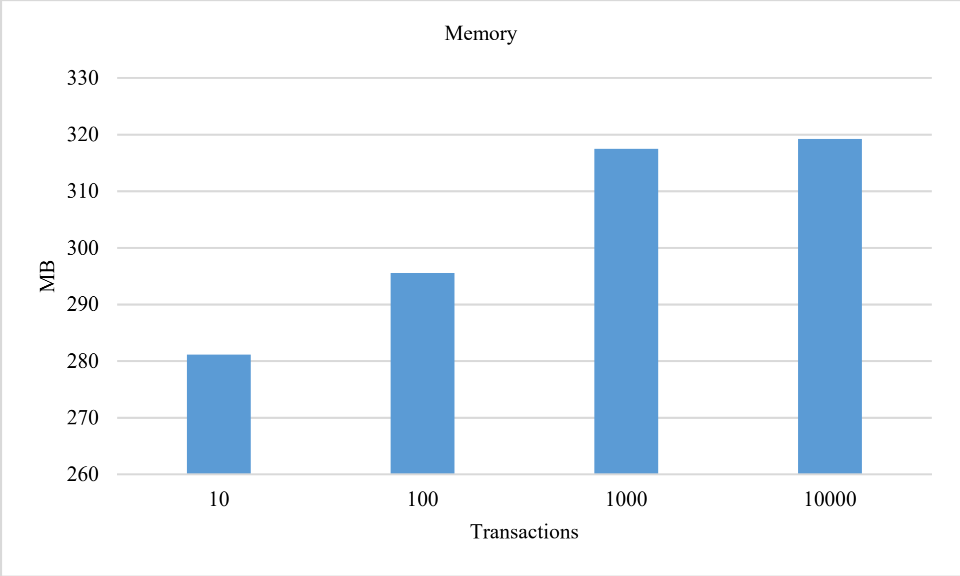

Fig. 4. Memory usage for different number of transactions

The parameters for the different number of peer are offered in Table 2. The number of transactions (10000) is analyzed with the different number of peer nodes (2, 4 and 8) to measure the performance of the HLF blockchain likes throughput, latency, processor and memory usage.

Parameters	**Values**
Number of transactions	10000
Number of peers	2, 4 8

Table 2. The parameters for different number of peers.

Fig. 5 demonstrates the throughput of each implementation for executing the invoke function with the number of transactions (10000). The throughput increases if the number of nodes in the network raises. The number of the peer nodes in the network affects the performance of HLF blockchain system. The throughput is dramatically increased between the numbers of the peer nodes (2 and 4). The responses for transaction need to wait from the more number of peers before creation as a block in blockchain. Therefore, the transaction throughput is not more increased between the numbers of the peer nodes (4 and 8).

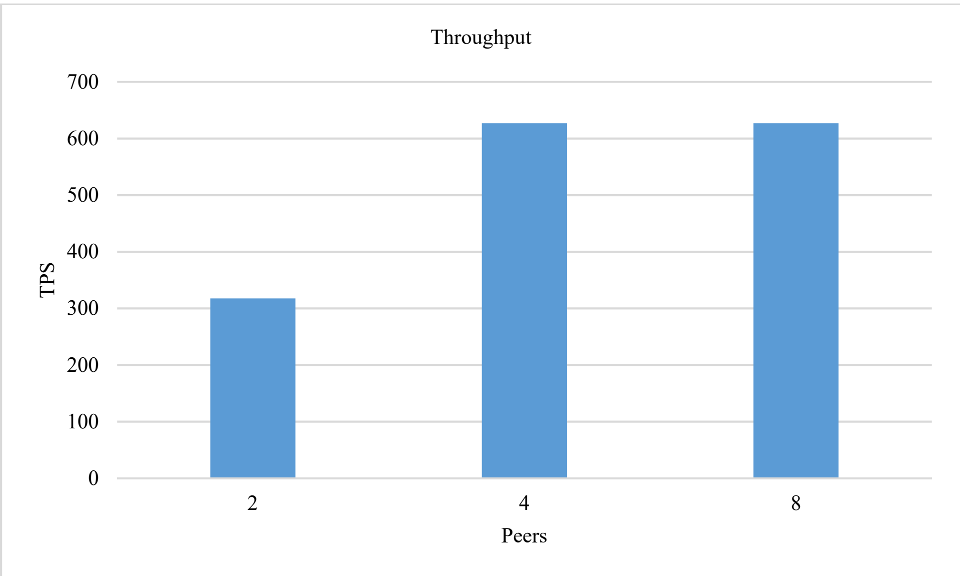

Fig. 5. Throughput for different number of peers

The maximum latency is demonstrated for each implementation of executing the invoke function with the number of transactions (10000) in Fig. 6. The latency rises if the number of peer nodes in the network increases. The number of the peer nodes in the network also affects the performance of HLF blockchain system. The latency is slightly increased between the numbers of the peer nodes (2 and 4). The responses for transaction need to wait from the more number of peers before creation as a block in blockchain. Therefore, the latency dramatically increases between the numbers of the peer nodes (4 and 8).

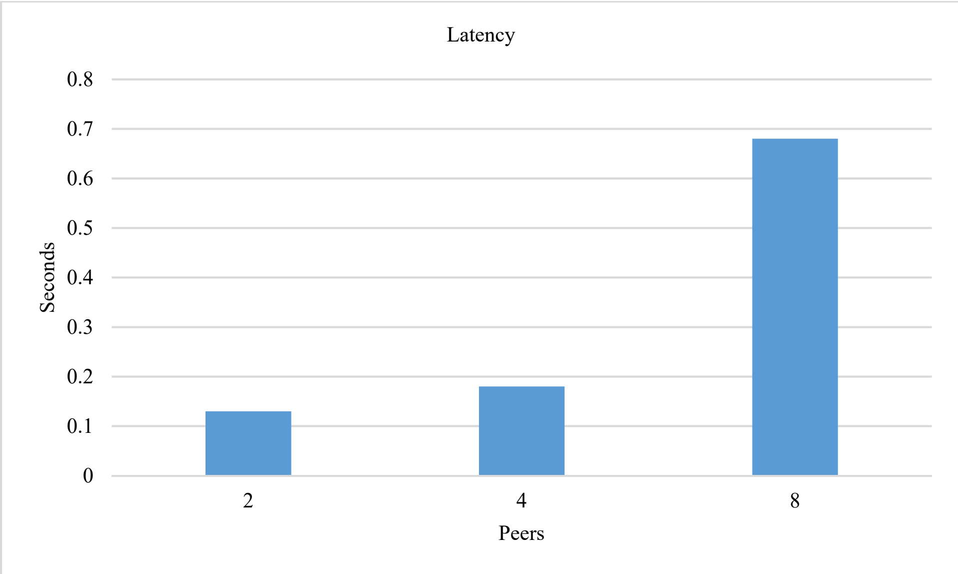

Fig. 6. Latency for different number of peers

According to the performance results, the usage of processor for the different number of peer nodes are shown in Fig. 7. The number of transactions (10000) are used for each implementation of executing the invoke function with. If the number of peer nodes are higher, the percentage of processor usage for execution will also be improved. The latency is dramatically increased between the numbers of the peer nodes (2 and 4). The latency slightly rises between the numbers of the peer nodes (4 and 8). Therefore, the processor usage is also affected by the number of peer nodes in the network.

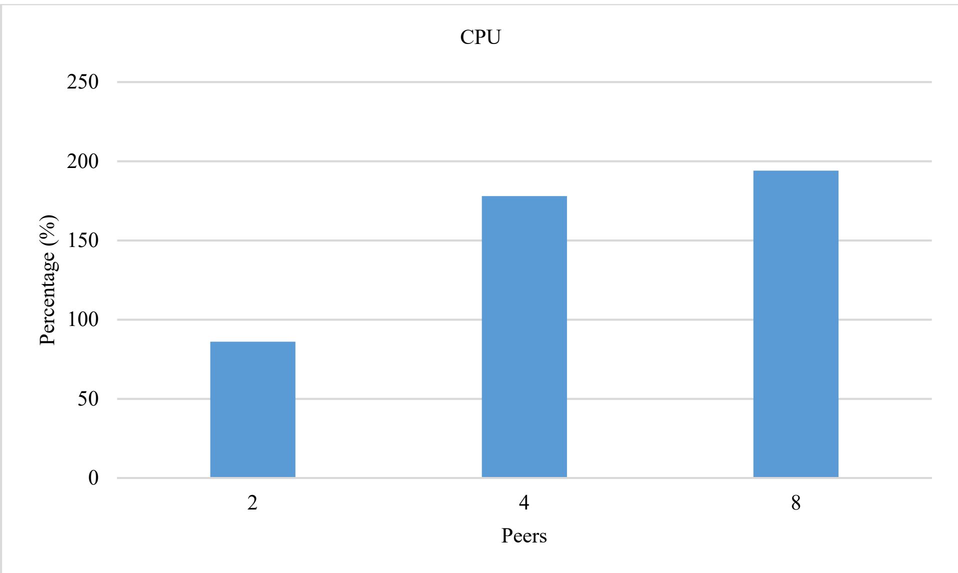

Fig. 7. Processor usage for different number of peers

The usage of memory for the different number of peer node (2, 4 and 8) for the invoke function are shown in Fig. 8. The number of transactions (10000) are used for each implementation of executing the invoke function. If the number of peer nodes are higher, the memory usage will also be increased. The memory usage is slightly increased between the numbers of the peer nodes (2 and 4). The memory usage dramatically increases between the numbers of the peer nodes (4 and 8). The transactions are replicated in every peer nodes of the HLF blockchain network. Therefore, the memory usage is also affected by the number of peer nodes in the network.

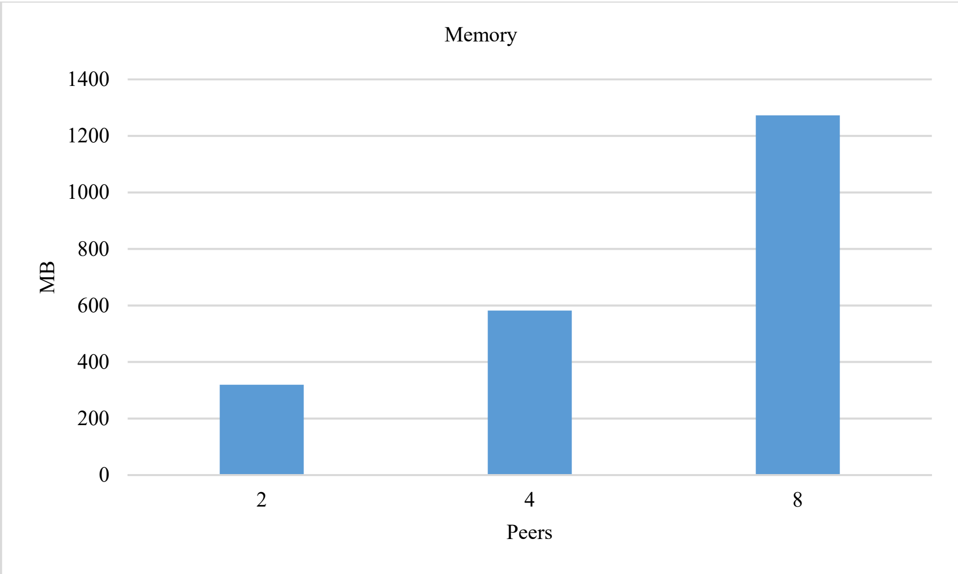

Fig. 8. Memory usage for different number of peers

6. Conclusion

This paper observes the performance of HLF blockchain by the different number of the transactions and peer nodes. The proposed system undertakes an analysis of manipulating the different nodes count in the fabric platform. It measures the throughput, latency, CPU usage, and memory usage in the invoke function, maintaining a constant number of transactions in the number of transactions (10, 100, 1000 and 10000) in the different number of peers in the network. According to the results, the amount of throughput will be increased if the number of transactions are higher up to 1000. The transactions 1000 is the highest throughput in 2 peer nodes. If the number of transactions are greater, the maximum latency, processor and memory usage for execution will also be increased. The responses for transaction need to wait from the more number of peers before creation as a block in blockchain. The transactions are replicated in every peer nodes of the HLF blockchain network. As a different number of peer nodes point of view, the maximum latency, processor and memory usage for execution will be increased if the number of peer nodes raise. Therefore, the scalability of the system has become one of the most important design objectives in the implementation of blockchain-based system. This analysis will provide valuable guidelines for designing an efficient and effective HLF blockchain system. The results outlined in this paper's analysis provides as a valuable resource consumption of network for the business industry, serving in the selection of the most suitable blockchain platform for their operations.

Conflicts of interest: All authors have no conflicts of interest in this paper.

References

[1] Dinh, T.T.A., and et al, 2018. “Untangling blockchain: A data processing view of blockchain systems”. IEEE transactions on knowledge and data engineering, 30(7), pp.1366-1385.
[2] Faroug, A. and Demirci, M., 2021, December. “Blockchain-Based Solutions for Effective and Secure Management of Electronic Health Records”. In 2021 International Conference on Information Security and Cryptology (ISCTURKEY) (pp. 132-137). IEEE.
[3] Htet, C.C. and Thu, A. M., 2023, September. “Performance Metrics with Block Height on Hyperledger Fabric Blockchain”. In 2023 International Conference (SARC) (pp. 41- 47).
[4] Nasir, Q., and et al, 2018. Performance analysis of hyperledger fabric platforms. Security and Communication Networks, 2018.6
[5] Wai, K.S.S. and Thein, N.N.M., 2023, February. Performance Analysis on Block Size Valuation of Hyperledger Fabric Blockchain. In 2023 IEEE Conference on Computer Applications (ICCA) (pp. 50-55). IEEE.
[6] Wickboldt, C., 2019. “Benchmarking a blockchain-based certification storage system” (No. 2019/5). Diskussionsbeiträge.

[7] Yuan, P., and et al, 2020. “Performance modeling and analysis of a Hyperledger-based system using GSPN”. Computer Communications, 153, pp.117-124.
[8] Yusuf, H. and Surjandari, I., 2020. “Comparison of performance between kafka and raft as ordering service nodes implementation in hyperledger fabric”. International Journal of Advanced Science and Technology, 29(7s), pp.3549-3554.
[9] Zeng, and et al., 2019. “A consortium blockchain paradigm on hyperledger-based peer-to-peer lending system”. China Communications, 16(8), pp.38-50.
[10] Zhang, and et al., 2020, September. “Performance Diagnosis and Optimization for Hyperledger Fabric”. In 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS) (pp. 210-211). IEEE.
[11] Blockchains. 2023. A Guide on Quorum Blockchain and their Use Cases. [online] Available at: <https://101blockchains.com/quorum-blockchain-use-cases/> [Accessed 4 June 2023].
[12] En.wikipedia.org. 2023. Hyperledger - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Hyperledger> [Accessed 4 June 2023].
[13] Medium. 2023. Hyperledger Fabric in practice. Main components and running them locally. [online] Available at: <https:// medium.com/coinmonks/ hyperledger-fabric-in-practice-part-1-main-components-and-running-them-locally-aa4b805465fa> [Accessed 30 May 2023].
[14] Programmersought.com. 2023. “Building Hyperledger Caliper on Fabric 2.2.0 Performance Test” - Programmer Sought. [online] Available at: <https:// www.programmersought.com/article/29668617484/>[Accessed 30 May 2023].
[15] Scalablockchain.com. 2023. Corda Platform for Blockchain | Scala Blockchain. [online] Available at: <https:// www.scalablockchain.com /corda.html > [Accessed 4 June 2023].

Authors Profile

Cho Cho Htet, is a PhD candidate at Mandalay Technological University, Mandalay, Myanmar. She completed M.E in Computer Engineering and Information Technology in 2015 from Mandalay Technological University. She is working as an Assistant Lecture in the Department of Computer Engineering and Information Technology at Mandalay Technological University. Information Security, Hyperledger Fabric and Blockchain Technology are some of her research interests.

Aye Myat Thu, received Ph.D (IT) at Mandalay Technological University, Mandalay, Myanmar, in 2014, she is working as a Professor in Department of Computer Engineering and Information Technology at Mandalay Technological University. Her research interest areas are Information Security, and Cryptographic.