Performance Analysis of Hyperledger Fabric on Multiple Infrastructure Setup

2023 International Conference on Digital Applications, Transformation & Economy (ICDATE) | 979-8-3503-1068-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDATE58146.2023.10248830

1st Nur Farahi Binti Idris
Information Security Lab
MIMOS Berhad
Wilayah Persekutuan Kuala Lumpur, Malaysia
farahi.idris@mimos.my

2nd Muhammad Azlan Bin Suhaimi
Information Security Lab
MIMOS Berhad
Wilayah Persekutuan Kuala Lumpur, Malaysia
azlan.suhaimi@mimos.my

3th Mohd Shakir Bin Zakaria
Information Security Lab
MIMOS Berhad
Wilayah Persekutuan Kuala Lumpur, Malaysia
shakir.zakaria@mimos.my

4rd Ahmad Zuhairi Bin Ismail
Information Security Lab
MIMOS Berhad
Wilayah Persekutuan Kuala Lumpur, Malaysia
ahmadz@mimos.my

***Abstract*—Blockchain technology has transformed traditional data storage and organization by introducing decentralized ledgers, ensuring data integrity through its resistance to tampering and alteration. Hyperledger Fabric, a popular industry platform, effectively manages high volumes of transactions on the blockchain. However, concerns about throughput and the factors influencing its value persist. This research paper conducts a performance analysis of Hyperledger Fabric on two distinct infrastructure setups: physical servers and virtual servers. The goal of this evaluation is to establish a baseline configuration derived from the performance results, which can be beneficial for future blockchain enhancement research. The experimental outcomes, based on 18,000 transactions, demonstrate that physical servers generally outperform virtual servers in terms of throughput, success rate, average latency, and resource consumption. These findings can guide practitioners in optimizing their blockchain platform adoption, tailored to their specific application requirements.**

Keywords—performance baseline, Hyperledger Fabric, Hyperledger Caliper, physical server, cloud server

I. Introduction

Blockchain, a decentralized and immutable database, simplifies asset tracking and transaction recording within corporate networks. As a distributed ledger, it allows trusted parties to exchange transactions in a peer-to-peer manner. The technology's numerous benefits have led to its adoption across various industries such as supply chain operations [1], property investments [2], commerce [3], and healthcare [4]. Multiple blockchain platforms, including Hyperledger Fabric [5], Ethereum [6], Corda [7], Omni [8], Ripple [9], and MultiChain [10], offer versatile frameworks for diverse purposes. Users are primarily concerned with data throughput, regardless of the platform.

Hyperledger Fabric, a popular platform for building blockchains, has gained attention for its low throughput while handling large transaction volumes. To understand this issue, performance evaluation tests are required using benchmarking tools like Hyperledger Caliper [11], Blockbench [12], and BCTMark [13]. Previous studies have evaluated Hyperledger Fabric versions, different blockchain platforms, and the number of peers. However, limited research has been conducted on the impact of varying infrastructure setups.

Existing research by Qassim et. Al [14] has compared Hyperledger Fabric versions 1.0 and 0.6, revealing that v1.0 consistently outperforms v0.6 in scalability, throughput, execution time, and latency. However, Julian Dreyer et. Al [15] has performed same research as Qassim, and found out that Hyperledger Fabric version 2.0 is better than the others. Another study [16] compared Hyperledger Fabric and Ethereum, showing that the former has better latency and throughput and consumes fewer hardware resources. However, Ethereum had a higher transaction success rate. Furthermore, that study [16] tested seven different SDK versions of Hyperledger Fabric (v1.0 to v1.4.4), observing improvements in the Transfer function success rate with each subsequent version.

To the best of our knowledge, no research has compared blockchain performance on different infrastructure setups, such as hardware servers and virtual servers. Identifying baseline server configurations is crucial for achieving optimal throughput and guiding practitioners in selecting the best infrastructure for their business purposes. This paper examines the performance of Hyperledger Fabric v2.2 on two distinct infrastructure setups with identical CPU and memory specifications, and conducts various test scenarios to establish baseline capacities.

The paper is structured as follows: Section I presents an introduction, literature review, and research objectives. Section II offers a high-level overview of Hyperledger Fabric, Hyperledger Caliper, and the system architecture flow. Section III discusses the characteristics of both server types and the blockchain platform access mechanism. Section IV analyzes the findings and their implications. Finally, Section V concludes the paper.

II. Overview of Hyperledger Fabric and Hyperledger Caliper

A. Overview of Hyperledger Fabric

Managed by IBM and the Linux Foundation, Hyperledger Fabric is an open-source private permissioned blockchain platform designed for enterprise applications. It comprises three main components: peers, clients, and ordering nodes. Figure 1 illustrates the general transaction cycle involving these components. The Hyperledger Fabric documentation [5] provides a brief explanation of each component's functions:

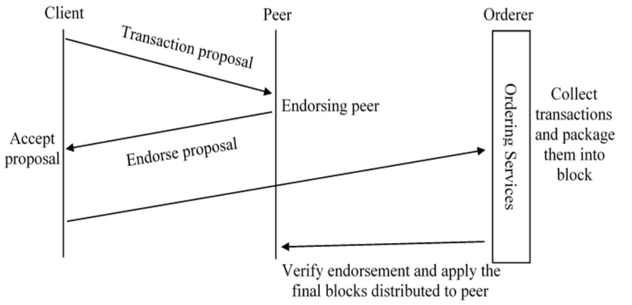

Fig 1. General transactions of three main components in Hyperledger Fabric

1. Client: Acts on behalf of end-users, requesting transaction proposals from peers, submitting endorsed transactions to the ordering service, and receiving transaction updates after completion.
2. Peer: Validates and executes transactions while maintaining a full ledger copy. Peers may serve as endorsing peers, confirming proposal validity, or committing peers, committing proposals to the blockchain. These roles can be performed independently or by the same peer entity within an organization.
3. Orderer: Packages endorsed transactions into blocks and sends them to committing peers for ledger addition.

Transaction regulation in Hyperledger Fabric involves chaincode, computer code that enables users to define functions for network interaction. A transaction, according to the official documentation, is an invoke or instantiate request submitted by peers for ordering and validation. To secure transactions, channels are created, granting access to transactions and data only to channel members. Figure 1 depicts the general transaction cycle from client to orderer.

B. *Overview of Hyperledger Caliper*

Launched in 2018, Hyperledger Caliper is a benchmarking tool for measuring blockchain performance, assessing response time, latency, or specialized tasks like block-writing time to persistent storage. The test results usually involve the effects of dependent variable changes, such as system throughput when the number of transactions varies. This facilitates the development of standard measurements, known as benchmarking, for comparing systems using the same baseline configuration obtained from evaluation tests.

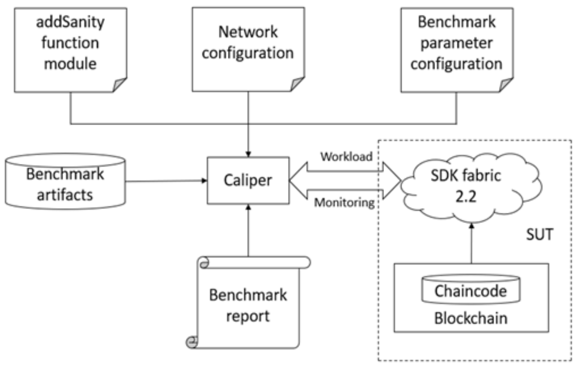

Fig 2. Flow of Hyperledger Caliper to the Blockchain

Hyperledger Caliper requires several inputs for benchmarking. Three configuration files are set up before testing: the benchmark configuration file, detailing benchmark execution parameters (e.g., the number of rounds, transaction rates, and transaction content-generating modules); the network configuration file, describing the System Under Test (SUT) topology (in this research, SDK Fabric version 2.2 and peer version 2.2 are installed, with MIMOS chaincode deployed using Go language and a single-peer, single-organization identity for invoking the addSanity test case); and the workload module, containing the smart contract functions, including the addSanity function in this case.

III. Methodology

This paper evaluates the performance of two servers with distinct specifications by manipulating and examining key variables, such as the number of workers, number of transactions, transaction loads, and transaction durations. The experiment is divided into three stages to assess the server's ability to handle a high volume of transactions.

A. *Phase 1: Blockchain platform and server setup*

The performance experiment was conducted using Ubuntu 18.04 LTS, with Hyperledger Fabric version 2.2 as the blockchain platform and Hyperledger Caliper as the benchmark tool. Two servers with different specifications were used a physical server with 24 CPU cores and 256GB memory, and a virtual server (cloud) with 12 CPU and 32GB memory. To ensure a comparative analysis, CPU and memory restrictions were standardized between the physical and virtual servers to 12 CPU and 28GB memory by adjusting the container in the docker-compose files. A custom chaincode was deployed in the network to carry out the performance test on both servers, using the identity specified in the network configuration to invoke the custom smart contract.

B. *Phase 2: Benchmark Framework Implementation*

A benchmark file is configured within Hyperledger Caliper to outline how the benchmark should be executed. In this paper, various parameters are adjusted, such as the number of workers and transaction load. Two test cases, as shown in Table 1, are employed to examine the impact of workers and transaction load on the system's throughput. Before running Caliper, a network configuration file is created, describing the System Under Test (SUT) topology, including network endpoint addresses, clients or identities present in the network, and the smart contract ID for communication with Caliper. Upon test completion, the application automatically generates an HTML report containing performance information, such as successful transactions, latency, and system throughput.

C. *Phase 3: Performance result collection*

In addition to Hyperledger Caliper results, data on CPU utilization during Caliper operation, memory usage, and the rate of data written to disk I/O in blocks per second were collected. The performance test was repeated three times to obtain an average result. This methodology was implemented to understand the server's behavior when the blockchain is operational.

IV. Findings

Two test cases are defined by varying key parameter specifications. This testing is executed on the physical hardware server. The resulting performance of the blockchain platform is then recorded and analyzed. Specific parameters used for performance evaluation are summarized in Table I.

TABLE I. TEST CASE TO EVALUATE THE SYSTEM'S PERFORMANCE

Parameter	No. of workers	No. of tx	Tx duration	Txload
Test case I: Impact no. of workers	20-100	18000	60	1
Test case II: Impact of tx load	30	18000	60	1-35

Test case I aims to discover the optimal number of workers capable of producing high throughput while handling 18,000 transactions in 60 seconds. Various numbers of workers are examined, ranging from 20 to 100, with a 10-worker interval.

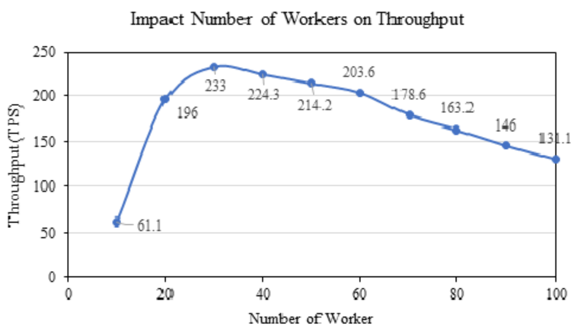

Fig 3. Impact number of workers on throughput in physical server

Figure 3 shows that throughput increases when the number of workers reaches 30. This implies that when submitting 18,000 transactions to the blockchain for 60 seconds, the optimal number of workers is 30, which can produce a high number of throughputs. Afterward, throughput decreases gradually. In this specific test environment, the system can handle up to 233 throughput per second. The observation indicates that the system can receive approximately 18,000 transactions to the blockchain without errors in 60 seconds by having 30 workers. Each worker in Hyperledger Caliper represents a simulated user or client that interacts with the blockchain network, executing transactions and generating workload. By having more workers, the clients will actively participate in the network, and continuously generate and process transactions. This will lead to high load on the network, which require additional computational resources such as CPU, memory and network bandwidth, which in turns leads to higher power consumption. As a result, the resources get shared among a larger number of workers, potentially causing performance degradation and lower throughput.

Figure 4 demonstrates that throughput increases with a positive non-linear pattern until it reaches 20 transaction loads, after which it gradually decreases. This result indicates that in test case II, when the number of transaction loads increases, the throughput also increases. However, the system experiences a decrease in throughput after implementing more than 20 transactions of loads, which may be due to the latency of the data submitted to the system. The pattern of latency resembles a linear graph. Meanwhile, Figure 5 represents the throughput comparison between the physical server and the private cloud.

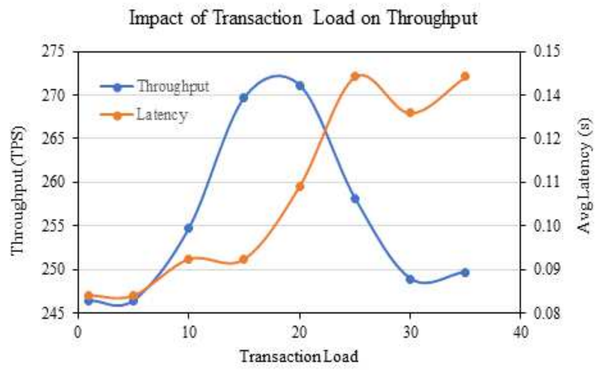

Fig 4. Impact of transactions load on throughput in physical server

To effectively compare the performance of a physical server and a private cloud, it is crucial to align the specifications of the physical server with the private cloud. A test was conducted to process 18,000 transactions with a 20 transactions load within 60 seconds. The results in Figure 5 show that the private cloud's performance increases gradually until the 20-second mark, and the physical server's performance increases until the 30-second mark, after which both stabilize. However, the private cloud's throughput remains stagnant, while the physical server can handle more than 20 TPS even after 40 seconds. This suggests that the private cloud may reach its maximum throughput capacity at 40 seconds and become saturated when processing a large volume of data, whereas the physical server can continue to process data efficiently beyond that point.

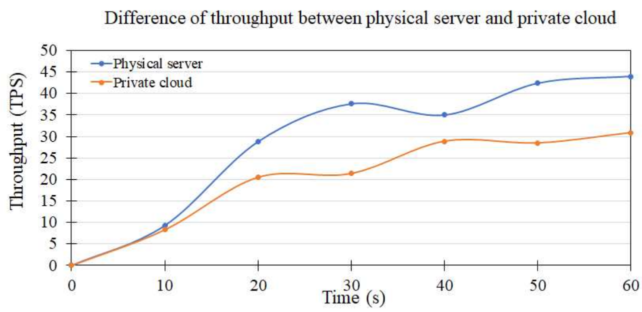

Fig 5. Difference throughput between physical server and cloud

Transactions can also impact CPU utilization. Each transaction requires processing power to execute the necessary operations, and the more transactions processed, the more CPU resources are typically required. During the warmup phase, only 1/12 of the CPU load was utilized, indicating low computational intensity. In contrast, CPU load fluctuated during the workload loop and addSanity function, suggesting that Hyperledger Fabric performance depends on transaction complexity and workload. As shown in Figure 5, the physical server can handle higher transactions compared to the private cloud, resulting in higher TPS. Since more transactions are processed by the physical server, the CPU utilization is higher compared to the private cloud, as shown in Figure 6.

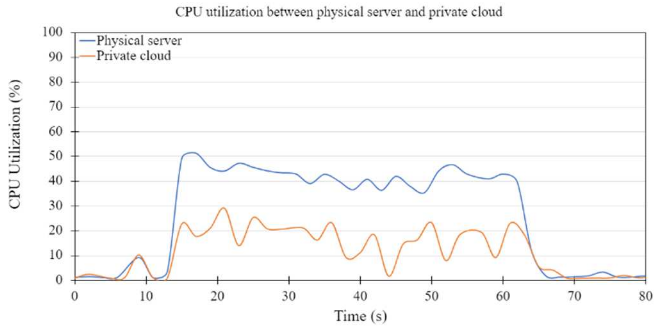

Fig 6. Percentage of CPU utilization between two different infrastructure setups

The physical server consumed half of the total CPU load (60%) and committed 44 TPS, while the private cloud utilized 30% CPU load and achieved 30.9 TPS. The private cloud's inability to fully utilize the available CPU may be due to other limitations, such as disk I/O speed and latency caused by virtualization, resulting in lower performance compared to the physical server. Another reason for high utilization could be the Peer nodes and CouchDB, which are responsible for high CPU load and disk I/O consumption. Peer nodes ensure the security, integrity, and availability of blockchain data through transaction validation, consensus participation, and data storage and retrieval. CouchDB serves as the state database, enabling efficient data management.

Figure 7 represents the I/O activities, where data written to servers in blocks per second are recorded. Initially, both plots remain the same because there is no process of writing in the block due to the workload initialization in Hyperledger Caliper. Examining the graph closely for the 60-second period from 8 seconds to 68 seconds reveals that data needing to be written into the block increases for both servers as they begin to receive a significant number of transactions. During this period, the physical server recorded 2,038 successful transactions added to the blockchain, while the private cloud recorded 1,599 successful transactions. Numerous read and write operations were performed on the physical server as it was able to process the data more quickly and efficiently. As a result, the two servers achieved different levels of throughput.

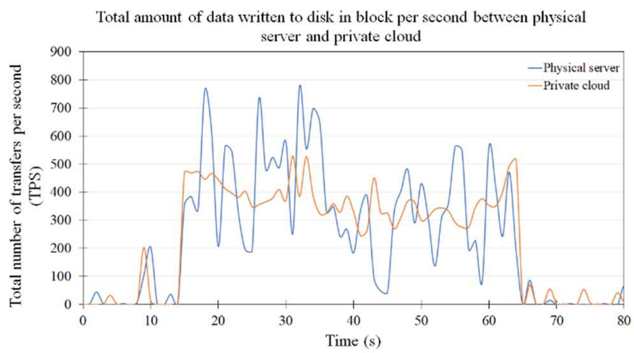

Fig 7. Total amount of data written to disk in block per second

Our private server employs full virtualization where each resources; e.g. CPU, memory and disk, are virtualized. Virtualization is a technology that enables virtual machines (VMs) to run in a multiple operating systems and applications on a single physical host machine. However, full virtualization comes with some overhead. The virtualization layer, also known as the hypervisor, requires additional processing power on the host machine to manage and coordinate the virtualized resources. Additionally, accessing virtualized resources introduces latency as the process within a virtual machine must pass through the virtualization layer to access the CPU, memory, or disk. This introduces some delays or overheads in the process, as the hypervisor needs to handle the request and translate it to the corresponding physical resources on the host machine.

The virtual CPU allocated to the virtual machine may not perform at the same level as the physical CPU on the host server, which is responsible for executing guest operating system instructions and managing virtual machine resources like memory and I/O devices. The lower total transfers per second observed in private cloud in Figure 7 suggests that the CPU is not efficiently managing I/O operations between guest operating systems and virtual machine resources, leading to slower response times and reduced system performance.

When processing a transaction, memory access operations like reading or writing data typically occur. Increased transaction processing requires more memory to store data and intermediate results. This can cause higher memory usage if the system does not efficiently manage memory allocation and deallocation. Figure 8 highlights the significant gap in memory consumption between the physical server and private cloud, with the former using a substantial amount of memory during the test. Virtual machines often use a portion of the host server's physical memory allocated by the hypervisor, resulting in slower memory allocation times and increased virtualization overhead.

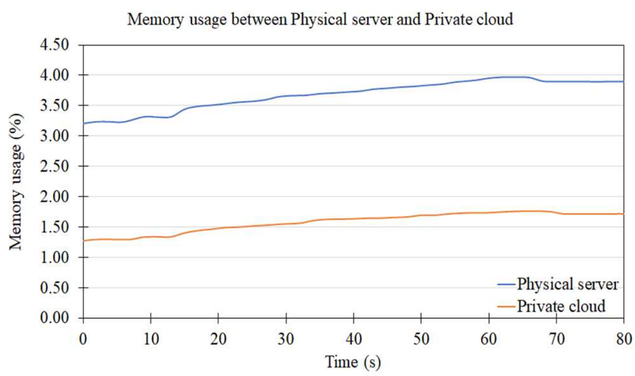

Fig 8. Percentage usage of memory between two different infrastructure setups

On the other hand, this Figure 8 indicates there is a significant gap between memory consumption of physical server and private cloud. Physical server utilized a significant amount of memory during the test compared to private cloud.

V. Conclusion

This paper presented a performance baseline and comparative performance test between physical server and private cloud on Hyperledger Fabric version 2.2. Assessments shows that physical server is running in a better performance with higher throughput compared to private cloud when the workload is varied up. As a result, when a high number of transactions are being processed in a private cloud, the virtual

CPU may become a bottleneck, causing the transaction processing to slow down. This can result in a lower number of transactions being processed per second, as compared to a physical server that has direct access to the physical CPU resources since there are additional overheads and complexities involved in managing I/O operations due to the need for virtualization software to translate between the virtual machine and the physical hardware. To sum up, the throughput of transaction can be improved by optimizing the configuration of container, as such in this research the CouchDB and peer nodes to ensure optimal performance of the CPU utilization. Other than that, the rate of data read and write in the CPU also can be improved by optimizing the software code to minimize the amount of time spent waiting for I/O operations to complete. While performing the test between physical server and private cloud, it can be concluded that Hyperledger Fabric performance is influenced by factors such as workload complexity, infrastructure choice, and the critical roles of peer nodes and CouchDB. The private cloud's underutilization of CPU resources suggests the presence of other limitations affecting its performance. The choice between deploying Hyperledger Fabric on physical or virtual servers involves a trade-off between performance and convenience. Physical servers offer dedicated hardware resources, resulting in potentially higher performance and stability. However, they require upfront investments, maintenance, and physical space. In contrast, virtual servers provide convenience through easy provisioning, scalability, and flexibility. They allow for rapid deployment and resource management but may introduce overhead and latency. The decision should consider specific network requirements, including performance, scalability, flexibility, and cost. Overall, the performance evaluation results across all the statistics, including successful transaction, transfer rate and throughput demonstrated that physical server has a better performance compared to private cloud by 30% different.

ACKNOWLEDGMENT

This work is financially supported by the MIMOS Berhad and MOSTI (Grant No: SRF PG3-P1 Blockchain System for Customs AEO and OGA).

REFERENCES

[1] S. E. Chang and Y. Chen, “When blockchain meets supply chain: A systematic literature review on current development and potential applications,” *IEEE Access*, vol. 8. Institute of Electrical and Electronics Engineers Inc., pp. 62478–62494, 2020.

[2] O. Konashevych, “Constraints and benefits of the blockchain use for real estate and property rights,” *J. Prop. Plan. Environ. Law*, vol. 12, no. 2, pp. 109–127, Jul. 2020.

[3] L. Zhang, Y. Xie, Y. Zheng, W. Xue, X. Zheng, and X. Xu, “The challenges and countermeasures of blockchain in finance and economics,” *Syst. Res. Behav. Sci.*, vol. 37, no. 4, pp. 691–698, Jul. 2020.

[4] A. Fusco, G. Dicuonzo, V. Dell’atti, and M. Tatullo, “Blockchain in healthcare: Insights on COVID-19,” *Int. J. Environ. Res. Public Health*, vol. 17, no. 19, pp. 1–12, Oct. 2020.

[5] “Hyperledger Fabric – Hyperledger Foundation.” https://www.hyperledger.org/use/fabric (accessed Oct. 28, 2022).

[6] “Ethereum.” https://ethereum.org/en/ (accessed Oct. 28, 2022).

[7] “Corda .” https://www.corda.net/ (accessed Oct. 28, 2022).

[8] “Omni Layer.” https://www.omnilayer.org/ (accessed Oct. 28, 2022).

[9] “Ripple.” https://ripple.com/ (accessed Oct. 28, 2022).

[10] “MultiChain .” https://www.multichain.com/ (accessed Oct. 28, 2022).

[11] H. Sukhwani, N. Wang, K. S. Trivedi, and A. Rindos, “Performance modeling of hyperledger fabric (permissioned blockchain network),” *NCA 2018 - 2018 IEEE 17th Int. Symp. Netw. Comput. Appl.*, pp. 1–8, 2018.

[12] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K. L. Tan, “BLOCKBENCH: A Framework for Analyzing Private Blockchains,” *Proc. ACM SIGMOD Int. Conf. Manag. Data*, vol. Part F127746, pp. 1085–1100, Mar. 2017.

[13] D. Saingre, T. Ledoux, and J. M. Menaud, “BCTMark: A Framework for Benchmarking Blockchain Technologies,” *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, vol. 2020-November, Nov. 2020.

[14] Q. Nasir, I. A.Qasse, M. Abu Talib, and A. Bou Nassif, “Performance Analysis of Hyperledger Fabric Platforms,” *Secur. Commun. Networks*, pp. 114–121, 2018.

[15] J. Dreyer, M. Fischer, and R. Tönjes, “Performance analysis of hyperledger fabric 2.0 blockchain platform,” *CCIoT 2020 - Proc. 2020 Cloud Contin. Serv. Smart IoT Syst. Part SenSys 2020*, pp. 32–38, 2020.

[16] M. Dabbagh, M. Kakavand, M. Tahir, and A. Amphawan, “Performance Analysis of Blockchain Platforms: Empirical Evaluation of Hyperledger Fabric and Ethereum,” *IEEE Int. Conf. Artif. Intell. Eng. Technol. IICAIET 2020*, 2020.