Performance Evaluation of Permissioned Blockchain Platforms

Ahmed Afif Monrat  
PhD Student  
Luleå University of Technology  
Skellefteå, Sweden  
Email: ahmed.monrat@ltu.se

Olov Schelén  
Professor  
Luleå University of Technology  
Skellefteå, Sweden  
Email: Olov.Schelen@ltu.se

Karl Andersson  
 Associate Professor  
Luleå University of Technology  
Skellefteå, Sweden  
Email: karl.andersson@ltu.se

Abstract—Blockchain is a technology for storing an immutable history of transactions in a decentralized platform by using cryptographic principles. Many industries have become interested in adopting blockchain within their IT systems. However, the accessibility, privacy, performance, and scalability aspects of different blockchain-based platforms are still legitimate concerns when designing an enterprise solution. Permissioned blockchain frameworks facilitate a way to immutably store confidential records. Numerous research studies have been carried out on the opportunities, challenges, application areas, and performance analysis of different public and permissioned blockchain-based platforms. However, the implication of blockchain in recent private enterprise solution requires detailed comparative analysis. This paper conducts a performance and scalability analysis of popular private blockchain platforms, including Ethereum (private deployment), Quorum, Corda, and Hyperledger Fabric. Each of these platforms is assessed by varying the workloads (no. of transactions and nodes) and determining the performance evaluation metrics such as throughput and network latency.

Index Terms—permissioned blockchain; performance evaluation, transaction, Ethereum, Hyperledger fabric, Corda, Quorum, Caliper

I. INTRODUCTION

Blockchain is a distributed ledger technology (DLT) that provides an immutable and trustable ledger of records. Since it was first introduced in 2008 [1], blockchain technology has surpassed Bitcoin cryptocurrency applications. Blockchain has become an emerging and leading technology that promises to revolutionize financial services [2], supply chains [3], healthcare [4], energy [5] and public services [6]. Blockchain supports a distributed ledger, where an identical copy of the ledger is replicated throughout a blockchain network. The technology typically allows transactions to be anonymous and secured among business partners, and automatically verifies and records data using cryptographic algorithms without the need for a central authority or intermediary. Several blockchain frameworks that provide adaptable and flexible platforms and support various applications have become available. While several blockchain projects are being piloted, there are some concerns about the technical challenges of a blockchain platform in terms of its throughput, latency and its ability to scale [7]. A blockchain network can be either permissionless (public) or permissioned (private). In a permissionless network or public network such as Bitcoin and Ethereum, anyone can join the network to initiate and validate transactions [8]. Due to a large number of nodes in a public network, a proof of work consensus approach is applied to order transactions and create blocks. In a public blockchain, the identity of the address/account owner is typically unknown.

In a permissioned blockchain, the identities of the owners are known and authenticated cryptographically [9]. This network can have built in extensive access control mechanisms to limit the access and issue transactions in the blockchain network.

Permissionless platforms are substantially concerning regarding performance, scalability and privacy due to their open access nature and resource expensive consensus process [10]. Conversely, the permissioned network is highly suitable for enterprise applications by ensuring the accessibility of authenticated participants while avoiding a complex consensus approach, which renders these platforms energy and resource efficient [11]. Therefore, it is not fair to compare public blockchain networks with private platforms while analyzing the performance issues.

This paper focuses on the performance and scalability evaluation of permissioned blockchain platforms. These following queries need to be answered:

a) How does each permissioned blockchain platform react to varying workloads regarding the number of transactions and nodes that are associated with the evaluation?

b) On which context does one platform facilitate better performance than the other platforms?

To answer these questions and to identify performance bottlenecks, this paper aims to present a quantitative analysis to investigate the performance of different permissioned blockchain based platforms by varying the network workloads. For this research, we have considered platforms such as Ethereum (private) [12], Corda [13], Quorum [14] and Hyperledger Fabric [15]. The network workload refers to a varying number of transactions, transaction rates and transaction types. Performance of the blockchain network that is being evaluated include the throughput (in transactions per second,tps), latency (in seconds) and scalability (i.e. the number of participants that the blockchain network can serve). The major contributions for this paper are listed as follows:
a) An empirical study of recent consensus protocols for permissioned blockchain platforms.
b) A quantitative performance analysis of different permissioned blockchain platforms by varying the network workloads, which reveals the limitations and bottlenecks of these systems.
c) Implementation of a proof of concept using a cloud computing service to facilitate the ideal environment for deploying and evaluating these platforms.

This paper is organized as follows: Section II provides an overview of the related works. Section III introduces the consensus protocols that are employed in the permissioned blockchain platforms. In section IV the deployment of the permissioned blockchain platforms in cloud computing services is presented. Section V presents the results and discussion on the the performance and scalability evaluation. Section VI concludes the paper.

II. RELATED WORKS

Dinh et al. proposed a Blockbench framework for benchmarking the performance of private blockchain platforms [16]. This research conducts a performance comparison among private blockchain such as Parity, Ethereum and Fabric (Version 0.6). To predict the latency of blockchain based systems, Yasaweerasinghelage et al. proposed a simulation framework and performance modeling [17]. Sukhwani et al. presents a performance modeling framework for a permissioned blockchain network that incorporates a PBFT consensus protocol [18]. Kocsis et al. proposed a performance evaluation model for blockchain technology, which was utilized to evaluate Hyperledger fabric v0.6 [19]. A comparison between proof of work and Byzantine fault tolerance based blockchains regarding scalability and performance is presented by Vuckolic et al. [20]. A performance evaluation for two private blockchain implementations; Ethereum and Hyperledger fabric, by varying the number of transactions was presented by Suporn et al [21]. Other works in this field are focused on evaluating the security, performance and scalability aspects of both public and private blockchains [22], [23], [24], [25], [26]. Based on this literature survey, table I presents a comparison among permissioned blockchain platforms. The design challenges in public platforms are quite different from those involved in building permissioned platforms, and therefore, a direct comparison of performance between the two types of platforms is not fair. To the best of our knowledge, we did not identify any papers that conduct a thorough performance and scalability evaluation among permissioned blockchain platforms, especially among Ethereum, Corda, Quorum, and Hyperledger. Therefore, this research aims to determine the discrepancies regarding performance, such as throughput and latency, among these permissioned blockchain platforms.

III. CONSENSUS PROCESS IN DIFFERENT PERMISSIONED BLOCKCHAIN PLATFORMS

In a blockchain network, while processing a block, the nodes are required to perform certain activities, such as participating in the authentication and verification of the transactions, block mining activities, communication over the network, and collaborative construction of trust within the blockchain system without the interference of a central authority. There is always the risk that individual nodes may behave maliciously or act against the principle goal or that the network communication crashes. To facilitate continuous service that ensures the availability, confidentiality, integrity, and accessibility, a secure mechanism is needed to make all the participant nodes reach a global agreement about which information should be appended to the blockchain [27]. This process is known as a consensus protocol, which stimulates the trust of all nodes in the blockchain system.

This section presents a brief discussion on some of the widely utilized consensus approaches in different permissioned blockchain platforms, such as PBFT, RAFT, Apache Kafka, proof of authority (PoA) and QuorumChain.

A. PBFT

Practical Byzantine Fault Tolerance (PBFT) is an improved version of the Byzantine fault tolerance (BFT) consensus algorithm, where members are partially trusted and the system can resist the class of failures derived from the Byzantine Generals’ Problem [28]. This consensus approach is mostly seen in the Hyperledger Fabric platform. PBFT facilitates the certainty of reaching a consensus even when the network consists of malicious nodes. However, the number of malicious nodes in the network should not exceed more than one-third of the total number of nodes, which can be considered as a significant constraint. The network becomes more secure as the number of nodes increases. To complete a transaction, the majority or at least 51% of the nodes in the network have to approve the transaction [29]. It is important to observe that the protocol does not require a 100% consensus ratio because a valid transaction can be rejected if one malicious node does not approve the transaction. Some nodes in a permissioned network might reject the transaction with malicious intent. The PBFT consensus approach can address this issue. However, the message count increases exponentially while adding nodes or replicas in a PBFT network.

B. RAFT

Permissioned Blockchain platforms such as Quorum and Hyperledger Fabric use the RAFT consensus algorithm. This consensus approach ensures that all the nodes in a cluster agree upon the same series of state transition by offering a generic way to distribute a state machine across a cluster of computing systems [30]. Nodes in RAFT can have separate identities, such as candidate, follower or leader. The nodes nominate the leader via a process known as election. The leader is responsible for sending all messages to the
participating nodes in the network. Initially, when the leader receives a message, it propagates the message to other nodes. These nodes conduct the writing and verification process. The leader only commits the message when all the nodes reach to an agreement about writing the message and sending a response to the leader. At this point, the follower nodes also commit the message, and a consensus is reached. RAFT is tolerant for handling network partition failure.

C. Apache Kafka

Apache Kafka is not a traditional consensus protocol; it is a publish-subscribe solution that supports topics, wherein published messages are serialized in the same order for all subscribers. Apache Kafka is crash fault tolerant (CFT), which prevents the system from failing when the nodes crash or go offline [31]. However, it is not BFT, which prevents the system from reaching an agreement in the case of malicious or faulty nodes. This protocol is mostly employed in a Hyperledger Fabric ordering service. Apache Kafka is a permissioned voting-based distributed streaming platform that is aimed at high throughput and low latency [32]. Apache Kafka consists of two types of nodes: broker nodes and consistency nodes. The network messaging is processed by a set of redundant brokers. To address the network issues and crashes, a Zookeeper instance is employed in the consistency nodes that coordinate the brokers. The ordering process is carried out by the leader and only in-sync replicas can be voted as leader.

D. PoA

Proof of authority (PoA) refers to a reputation-based consensus algorithm that introduces a practical and efficient solution for the private implementation of blockchain networks [33]. This consensus can be utilized in the Ethereum network for private deployment purposes. In PoA, the block validators are staking their reputation or identity instead of coins which makes this algorithm different from proof of stake (PoS). PoS is not always suitable for certain businesses and corporations due to the decentralized nature of most blockchain networks [34]. Conversely, PoA systems may represent a better solution for private blockchains because it can provide higher throughput. The preapproved participants act as moderators of this system and are responsible for verifying the blocks and transactions.

E. QuorumChain

QuorumChain uses a smart contract to validate blocks [35]. The model has a set of “voter” and “block-maker” nodes, whose identities are known to the network. This protocol implies the standard peer to peer gossip layer of Ethereum to form the blocks. The logic is formulated as a smart contract that is deployed with the genesis block. All the messages are digitally signed by the nodes. The block-maker nodes propose the block to be added in the network, while the voter nodes are responsible for the approval and validation process. By default, there is one block-maker node in this network.

IV. Deployment of Blockchain Platforms in Cloud Computing Service

This section describes the deployment of permissioned blockchain in cloud computing services. The perks of integrating blockchain are similar to other distributed systems, such as booting, accessing, and scaling a system with ease compared to an on-premises network. For developing a proof of concept, we chose the Microsoft Azure Platform. Azure provides a different service model, such as Infrastructure as a service (IaaS) and platform as a service (PAAS), for provisioning a fully configured blockchain network topology. Cloud services are used to develop the proof of concept to mitigate the challenges in performing a complex performance evaluation in a local machine. Moreover, establishing a system configuration for blockchain-based solutions is complicated.

1) PoA Ethereum Deployment: Azure Blockchain Service provides an Ethereum PoA consortium template, which facilitates deploying, configuring, and provisioning a network. This template can be utilized by any consortium member nodes to provision a block network. There is a set of activities that need to be performed to deploy a PoA network, which includes VMs for running PoA validators; using Azure load balancer for distributing RPC, peering, and governance DApp requests; and managing Azure Monitor for aggregating logs and performance statistics. First, we need an Azure subscription to access the Azure portal. Second, we

<table>
<thead>
<tr>
<th>Blockchain Platforms</th>
<th>Ethereum</th>
<th>Hyperledger Fabric</th>
<th>Quorum</th>
<th>Corda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Public</td>
<td>Enterprise</td>
<td>Enterprise</td>
<td>Enterprise</td>
</tr>
<tr>
<td>Purpose</td>
<td>Cross-Industry</td>
<td>Cross-Industry</td>
<td>Cross-Industry</td>
<td>Financial Services</td>
</tr>
<tr>
<td>Purpose</td>
<td>Business to Client (B2C)</td>
<td>Business to Business (B2B)</td>
<td>Financial Service Industry</td>
<td>Financial Service Industry</td>
</tr>
<tr>
<td>Smart Contract Programing Language</td>
<td>Solidity</td>
<td>GoLang, NodeJs</td>
<td>Solidity</td>
<td>Kotlin, Java</td>
</tr>
<tr>
<td>Currency</td>
<td>Ether</td>
<td>Can be built using chain-codes</td>
<td>Ether</td>
<td>No native Cryptocurrency (Corda coin)</td>
</tr>
<tr>
<td>Governance</td>
<td>DAO</td>
<td>Linux Foundation</td>
<td>Ethereum Foundation</td>
<td>R3</td>
</tr>
<tr>
<td>Consensus Algorithm</td>
<td>PoW</td>
<td>PBFT</td>
<td>RAFT</td>
<td>Pluggable Consensus</td>
</tr>
<tr>
<td>Throughput</td>
<td>A few 100s</td>
<td>more than 200 tps</td>
<td>200 tps</td>
<td>200 tps</td>
</tr>
</tbody>
</table>
deploy the Ethereum PoA consortium template by specifying some parameters, such as resource groups, subscription, authentication type, VM size, network size, etc. Fig. 1 shows the deployed Ethereum PoA overview, which includes blocks, transactions, pending transactions, etc.

2) Quorum Deployment: In Azure, the Quorum consortium can be deployed by providing some parameters. However, it is mandatory to understand the network architecture of the Quorum blockchain platform. The implementation of Quorum consists of QuorumChain, constellation, and peer security. Quorum platform uses the constellation as a tool for peer to peer encrypted message exchange. Peer security refers to the node permissions using a smart contract. Similar to the deployment of a previous permissioned blockchain network, we need to initiate the VM using an Azure subscription. Azure Marketplace has a deployment template known as the ‘Quorum Consortium Network’. This template will generate the VMs after specifying the standard parameters such as subscription, resource group, and basic VMs properties. The infrastructure deployed in Azure will deploy all virtual machines a single virtual network and a single subnet inside this network.

3) Corda Deployment: Azure Blockchain service facilitates the ability to provision Corda nodes in a private Corda network. It simplifies the process by providing easy configuration and deployment of the Corda node within the Azure portal or programmatically via REST APIs, CLI, or PowerShell. In addition to provisioning and deploying Corda nodes, Azure Blockchain Service provides managed APIs to help you manage your Corda nodes and Corda Distributed Applications (CorDapps). By leveraging the Azure monitor within Azure Blockchain Service provides facilities, such as CorDapps, health monitoring and logging information, customizing alerts and actions, and creates custom visualization of health monitoring data.

4) Hyperledger Fabric Deployment: Hyperledger Fabric network can be deployed and configured on Azure using the Azure Kubernetes Service (AKS) template. We deployed an ordering service and organizations with peers to build the Hyperledger Fabric network using the AKS. This template consists of components such as orderer nodes, peer nodes, a certificate authority (CA), and a world state database for the peer nodes. The Orderer node is responsible for carrying out transaction ordering in the ledger. Peer nodes host smart contracts and ledgers. Fabric CA allows us to manage certificates and identities. For storing chaincode data as simple key/value pairs and supporting composite key queries, LevelDB and CouchDB are applied as a default state database that is embedded in the peer nodes.

V. RESULTS & DISCUSSION

This segment identifies the performance matrices for evaluating permissioned blockchain platforms. In this section, the performance and scalability of Ethereum, Quorum, Corda and Hyperledger Fabric are demonstrated based on the network latency, and throughput. An experiment was carried out by assessing the performance of both a single-peer network and a multiper network. Further, a smart contract and the experimental setup to evaluate the platforms are discussed.

A. Performance Metrics

1) Latency: Latency is a network-wide view of the amount of time taken for a transaction’s effect to be usable across the network.

2) Throughput: Transaction throughput is the rate at which valid transactions are committed to a blockchain in a defined time period.

B. Workloads & Smart Contract

We have deployed a simple marketplace smart contract or chaincode to evaluate the performance of blockchain platforms. A smart contract is a piece of code that is intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract while triggering a transaction in the blockchain network [36]. This application expressed a workflow for simple transactions between an owner and a buyer in a marketplace. There are two roles in this application: owner and buyer. The owner is an individual who wants to sell on the marketplace, whereas the buyer wants to buy the products. The network consists of three sample states: ItemAvailable, OfferPlaced, and Accepted. ItemAvailable indicates that the owner has made his product available in the marketplace to sell. OfferPlaced refers to the offer made by the buyer to purchase the item from the marketplace. Accepted state defines that the owner has accepted the buyer’s
bidding for the product.

As shown in Fig. 2, the application’s workflow starts when the ItemAvailable state is initiated by the owner. In this state, the owner specifies the details and price of a product and makes it available for sale. Once the item is listed in the marketplace, a buyer can make a bid for this item by specifying the amount that he wants to pay. The state is changed from ItemAvailable to OfferPlaced due to this action. If the owner accepts the bid then the owner calls the function to accept the offer, the workflow reaches a successful conclusion and the state is changed to Accepted. Conversely, if the owner is not satisfied with the offer from the buyer, then the owner can call a function to reject the offer. At this stage, the state is changed to ItemAvailable again, which indicates that the item is still up for sale. The transitions between these states can continue until the buyer and seller reach to an agreement. This smart contract is deployed in the permissioned blockchain platforms before evaluating the performance of the platforms.

C. Evaluation Environment Configuration

All these permissioned platforms are deployed and run on Microsoft Azure VMs. For this experiment, we utilized Standard D4sv3 instances with 4 vCPUs at 3.7 GHz and 16 GB RAM. All the nodes have the Ubuntu 18.04 LTS operating system. To integrate the workloads (concurrent transactions and nodes) in the network and evaluate the performance and scalability of the private blockchain platforms, each network was configured with a maximum cluster of 24 peer nodes. The maximum number of concurrent transactions that are deployed as workloads is 10000.

We used a blockchain benchmark tool named Hyperledger Caliper. This tool allows users to measure the performance of a blockchain network with a set of predefined use cases. Hyperledger project has now officially incubated Caliper. The performance of a blockchain network can be measured by using this tool with a set of predefined use cases. The results generated by Caliper facilitates some insights on the performance and scalability to the Hyperledger projects, especially in supporting the choice of a blockchain implementation that is suitable for a user’s specific needs. For analyzing the performance of the Quorum blockchain network, we have employed the Caliper benchmarking tool integrated with a Quorum plugin. This plugin helps Caliper to record the throughput and transaction latencies by sending controlled workloads to the Quorum network. Caliper’s measurement framework operates on the client machines and sends transactions to peers in the Quorum network. Next, we chose BLOCKBENCH, which is an evaluation framework for analyzing private blockchains to evaluate the Ethereum platform. This tool is capable of measuring metrics such as latency, throughput, fault-tolerance, and scalability. A simple set of APIs can be used to integrate the blockchain platforms and workloads. The performance evaluation of the Corda platform has been carried away by installing the Corda Enterprise performance test suite for deploying a single node or a small set of nodes including a notary. This test suite uses Apache JMeter to start flows on nodes via RPC calls and capture the latency and throughput.

D. Performance & Scalability Analysis

This section analyzed the performance and scalability aspects of the permissioned blockchain platforms considering metrics such as latency and throughput.
From Fig. 3 and Fig. 4 we can easily distinguish the performance gap between PoA-based Ethereum deployment and PoW-based Ethereum deployment. The PoA-Ethereum outperforms PoW-Ethereum in terms of the network latency and throughput, due to the deployed consensus protocol. PoA is a much simpler and efficient protocol compare to PoW. The throughput in PoW-Ethereum is significantly lower than PoA-Ethereum due to the tradeoff between the decentralization and transaction throughput. PoW-Ethereum prefers decentralization by not trusting a small set of peers to orchestrate the block production and achieve a consensus by allowing any node to participate in the network. Conversely, PoA-Ethereum acts similar to a centralized consortium that processes transactions faster by relying on the fewer and known participants in the network.

Another interesting observation is that PoW-Ethereum’s throughput remain steady, because PoW is not dependent on the number of participating nodes. However, our results show that the performance initially rises but starts to decrease once a peak is reached, when nodes are added to the PoA-Ethereum setting.

Fig. 5 depicts that Corda Enterprise 4.5 can achieve a significantly higher throughput when compared to Corda Enterprise 4.3 for two reasons. First, In Corda Enterprise 4.5, flows that execute in parallel have significantly lower latency. This reduced latency means that nodes are able to complete more flows in the same amount of time, which achieves higher throughput. Second, the peer-to-peer messages between nodes can be compressed, which can lead to more efficient use of network bandwidth.

Fig. 6 shows that Corda Enterprise 4.3 shows exponential growth in the latency as the number of nodes that participate in the transaction are increased. Corda Enterprise 4.3 processes a transaction resolution in bulk, instead of one state at a time. Compared to Corda Enterprise 4.3, Corda Enterprise 4.4 shows a large decline in latency because the execution of the flows is sequential across the nodes; however, the cost per node is significantly smaller. Finally, Corda Enterprise 4.5 demonstrates the lowest latency flow in comparison with other versions due to the parallelized flow. Therefore, Corda Enterprise 4.5 can scale best with the growth in network size (nodes). The presence of additional participants has minimal impact on its latency.

Fig. 7 depicts the latency and throughput measurement of the Quorum platform with the growth in network size. We observe that the latency grows linearly with an increase in the number of peer nodes in the network. The latency reached its peak with 8 nodes and then experiences an exceptional depletion when the network grows further. This finding could be due to the overhead communications and experimental setup of the Quorum platform, which cannot handle more than 10 nodes in the network. On the other hand, the throughput is higher and experiences constant growth of the network up to 4 nodes; then it shows a steep decline.

Fig. 8 presents the scalability evaluation of the Hyperledger Fabric platform by varying the number of nodes up to 20. It can be observed that the scalability analysis is performed using the same metrics such as latency and throughput while using two different sets of transactions (1000Tx and 10000Tx). The results showed that the network can scale up to 16 nodes due to the overhead communication among nodes in the consensus protocol. Moreover, the platform cannot operate 10000 transactions when the number of nodes in the network exceeds 4 with the deployed experiment setup. Therefore, it can be seen that the performance of this platform deteriorates if the number of peer nodes in the network increases.

Fig. 9 and Fig. 10 demonstrates the comparison of latency and throughput, respectively, among Ethereum, Quorum, Corda, and Hyperledger fabric platforms. We can observe that Hyperledger Fabric out-performs Ethereum and Quorum with a large margin but performed slightly better than Corda. Hyperledger Fabric facilitates a short latency period while processing transactions compared to other permissioned platforms. Therefore, this platform is capable of providing
better throughput as well. Our experimental observations reveal that Hyperledger Fabric is performing better than other private platforms due to its simple and efficient modular consensus approach.

VI. CONCLUSION & FUTURE SCOPE

This research presents a performance and scalability analysis of permissioned blockchain platforms, including PoA based Ethereum, Quorum, Corda, and Hyperledger. These platforms are assessed by varying the deployed concurrent transactions as workloads and increasing the network size (nodes) to determine the scalability. Overall, the performance analysis results across all evaluation metrics, such as throughput and
latency, illustrate that Hyperledger Fabric performs better than other permissioned platforms. These private blockchains are deployed in the Microsoft Azure cloud computing platform to facilitate the computational resources that are required to perform the evaluation. The approach of the experimental setup in cloud service is a major strength of this research regarding performance benchmarking, instead of fragile deployments on local machines or speculative simulations. However, this experiment uses different benchmarking tools, such as Caliper, Blockbench and Corda enterprise test suite, to evaluate each of these platforms, which can be considered as a drawback. A single framework and test suite that supports these platforms could render the analysis more efficient and fair. In addition, the security concerns and energy consumption are not considered evaluation metrics. Addressing these limitations and optimizing the consensus approaches of these permissioned blockchain platforms can be considered in future work.

ACKNOWLEDGMENT

This study was supported by the Swedish Energy Agency under grant 43090-2, Cloudberry Datacenters.

REFERENCES


