

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

ScienceDirect

journal homepage: [www.elsevier.com/locate/cose](http://www.elsevier.com/locate/cose)Computers  
&  
Security

# A survey of empirical performance evaluation of permissioned blockchain platforms: Challenges and opportunities



Mohammad Dabbagh<sup>a,\*</sup>, Kim-Kwang Raymond Choo<sup>b</sup>, Amin Beheshti<sup>c</sup>,  
Mohammad Tahir<sup>a</sup>, Nader Sohrabi Safa<sup>d</sup>

<sup>a</sup> School of Science and Technology, Sunway University, 47500 Selangor, Malaysia

<sup>b</sup> Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249, United States

<sup>c</sup> Department of Computing, Macquarie University, Sydney, Australia

<sup>d</sup> School of Computing, Electronics and Mathematics, Coventry University, United Kingdom

## ARTICLE INFO

### Article history:

Received 3 May 2020

Revised 11 August 2020

Accepted 5 October 2020

Available online 9 October 2020

### Keywords:

Permissioned blockchain

Performance evaluation

Distributed ledger

Consensus Algorithms

Decentralized Platforms

## ABSTRACT

Blockchain-based platforms, particularly those based on permissioned blockchain, are increasingly popular in a broad range of settings. In addition to security and privacy concerns, organizations seeking to implement such platforms also need to consider performance, especially in latency- or delay-sensitive applications. Performance is generally less studied in comparison to security and privacy, and therefore in this paper we survey **existing empirical performance evaluations of different permissioned blockchain platforms published between 2015 and 2019**, using a comparative framework. The framework comprises ten criteria. We then conclude the paper with a number of potential future research directions.

© 2020 Elsevier Ltd. All rights reserved.

## 1. Introduction

We are living in an era where data is being generated at a significantly fast pace (IBM 2017), and this trend is likely to continue in the foreseeable future. Such a trend also necessitates the availability of platforms where data can be stored and exchanged reliably. Information exchange is typically controlled and centralized by some third-party entity. For instance, a financial transaction between two entities (e.g. consumer and the resource provider such as supermarket), a financial institution or credit card provider plays an intermediary role

to complete the transaction and the two involved entities in the transaction must trust the third-party. In providing the trusted service, the financial institution or credit card provider imposes a transaction processing fee. This process is typical in many other domains, such as healthcare and real estate. There are, however, situations where reliance on a centralized third-party entity is not desirable. Thus, there have been attempts to explore the potential of utilizing blockchain technology.

Blockchain technology provides a decentralized platform where information can be exchanged without relying on an external third-party entity. As a distributed ledger, blockchain

\* Corresponding author.

E-mail addresses: [mdabbagh@sunway.edu.my](mailto:mdabbagh@sunway.edu.my), [mohamad\\_dbgh@yahoo.com](mailto:mohamad_dbgh@yahoo.com) (M. Dabbagh), [raymond.choo@fulbrightmail.org](mailto:raymond.choo@fulbrightmail.org) (K.-K.R. Choo).

<https://doi.org/10.1016/j.cose.2020.102078>

0167-4048/© 2020 Elsevier Ltd. All rights reserved.

stores a copy of completed transactions on top of a peer-to-peer network (M. Dabbagh et al., 2019). The information about each completed transaction in the blockchain is immutable and also available to every involved party. These key features make blockchain technology a potential solution to facilitate enhanced security, reliability, transparency, efficiency, and cost-effectiveness of information processing. Although blockchain was initially applied to digital currency (i.e., Bitcoin) (Nakamoto, 2008), it has the potential to be applied in other settings and sectors, such as smart cities (Makhdoom et al., 2020), healthcare (Zhang et al., 2016; Esposito et al., 2018), insurance (Gatteschi et al., 2018), education (Sharples and Domingue, 2016), supply chain management (Kshetri, 2018), Internet of Things (IoT) (M. Dabbagh et al., 2019; Ali et al., 2019), 5 G networks (Tahir et al., 2020), etc. For example, over \$1 billion investment was reportedly made by technical and financial firms in 2016 to deploy blockchain technology into their businesses (Harty, 2018).

As its name suggests, blockchain is a chain of connected blocks where each block comprises multiple records of transactions. The security of these records depends mainly on the underlying cryptographic algorithms. Each block in a blockchain is associated with a unique identifier (i.e., hash), which is produced based on the data stored in that particular block, and the hash of the previous block. This allows the detection of any manipulation to a block of data, since the modification will result in the alteration of the hash of the modified block, as well as the other subsequent linked blocks. In other words, the hash identifiers play a critical role in ensuring the security and immutability of the blockchain. The decentralized feature of blockchain also makes it more resistant to attacks, for example minimizing the risk of single point of failure since multiple copies of data are replicated among all peers in the network. However, other conditions and factors need to be taken into consideration prior to deploying a blockchain in practice. There are two types of blockchain platforms, namely: permissionless and permissioned. Although both types of platforms share some similar characteristics, their key differences can influence the security requirements they can fulfill. In permissionless (also known as public) blockchain platforms, no permission is required to join a network. Therefore, anyone is able to freely join the network and perform read/write operations on the ledger. The most popular or widely known permissionless blockchain is Bitcoin. However, when additional layers of control and privacy are needed, we can consider permissioned blockchain platforms. These platforms require specific permissions prior to the user accessing and performing any read/write operation on the ledger. In other words, permissioned blockchain platforms provide an additional layer of security, where an access control layer is deployed in order to handle permissions for performing some specific operations by authorized participants (Su et al., 2020).

There are a number of popular blockchain platforms (e.g., Ethereum, Hyperledger Fabric, Multichain, Quorum, and Libra), and the options for consumers are growing. Hence, it can be challenging for consumers to determine which platform(s) is/are more appropriate for a given use case without an unbiased evaluation of the platform's key properties and features. Performance is generally a key concern in the adoption

of blockchain platforms, but it is generally less studied compared to other concerns such as security (Yli-Huumo et al., 2016; Swan, 2015). This reinforces the importance of unbiased evaluation of popular blockchain platforms' performance, despite the small number of empirical studies that attempt to quantify and measure performance characteristics of different blockchain platforms under divergent settings. Nonetheless, key differences, significant similarities, and potential opportunities for improvement can be overlooked unless existing empirical performance studies are also critically reviewed and analyzed. This is the focus of this paper.

Specifically, we will analyze, assess, and compare recent empirical studies focused on the performance evaluation of different permissioned blockchain platforms. By doing so, we answer the following research questions.

**RQ1: What are the existing empirical studies in the literature focusing on performance evaluation of permissioned blockchain platforms?**

Answering RQ1 would help the blockchain community, as well as the broader community, to understand the current state-of-play.

**RQ2: What are the key aspects and achievements of the identified empirical studies?**

Answering RQ2 would provide the blockchain community, as well as the broader community, a comparative overview of these empirical studies, such as key differences and common performance characteristics of different permissioned blockchain platforms.

**RQ3: What are the existing limitations and potential future research opportunities?**

Answering RQ3 would provide the blockchain and broader communities' potential research agenda.

This is the first attempt to conduct a thorough review of the empirical performance evaluation of permissioned blockchain platforms. Findings would potentially benefit business leaders (seeking to adopt a given blockchain platform for a specific application), blockchain developers (seeking to identify and optimize performance issues and bottlenecks of different platforms), and blockchain researchers (seeking to recognize potential opportunities for future enhancements).

The rest of this paper is organized as follows. A literature review on popular consensus algorithms and recent blockchain platforms is given in Section 2. Section 3 presents a detailed review of recent empirical studies focusing on the performance evaluation of permissioned blockchain platforms. Section 4 explains the comparison criteria. Section 5 presents a discussion on the results. Section 6 describes potential future research opportunities, based on the discussion in the preceding section. Finally, conclusions and future work are given in Section 7.

## 2. Literature review

This section reviews existing blockchain literature from two main perspectives. First, in Section 2.1, we explain the concept of consensus algorithm and give an overview of some popular consensus algorithms. Second, in Section 2.2, we review recent blockchain platforms.

**Table 1 – Summary of surveys on consensus algorithms.**

Source	Publication Year	Key contribution
(Mingxiao et al., 2017)	2017	It discusses the <b>most popular consensus algorithms</b> such as PoW, PoS, DPoS, PBFT, and Raft along with a theoretical analysis of their performance characteristics.
(Cachin and Vukolić, 2017)	2017	It provides an in-depth review on the consensus algorithms deployed in permissioned blockchain platforms.
(Bano et al., 2017)	2017	It introduces an evaluation framework to assess performance and security attributes of classical consensus algorithms.
(Nguyen and Kim, 2018)	2018	<b>It classifies the consensus algorithms into two main categories of proof-based such as PoW and voting-based such as PBFT.</b> Then, it highlights the main differences between various consensus algorithms under each category.
(Alsunaidi and Alhaidari, 2019)	2019	It provides a comparative analysis on well-known consensus algorithms. It also discusses the main factors that can influence the performance and security of consensus algorithms.
(Wang et al., 2019)	2019	It presents a detailed review on the recent design schemes used in developing different permissionless consensus algorithms.
(Sharma and Jain, 2019)	2019	It gives a summary of popular consensus algorithms along with their advantages and disadvantages.
(Sadek Ferdous et al., 2020)	2020	It presents different taxonomies of consensus algorithms from different perspectives to establish a baseline for a detailed comparison. This study also categorizes consensus algorithms into two types of incentivized and non-incentivized.

### 2.1. Consensus algorithms

Blockchain technology has attracted significant attention from a broad range of stakeholders since the success of Bitcoin (Nakamoto, 2008). **Blockchain is a decentralized ledger with the ability to record transactions between two entities without the intervention of an intermediary.** Instead, a set of nodes connect with each other via a peer-to-peer network. **When a transaction is made between two nodes, the other nodes would validate the transaction by some form of “proof” or to vote for an agreement between the nodes** (Nguyen and Kim, 2018). This validation process is achieved through a mechanism (also referred to as the consensus algorithm). Once the transaction has been verified among all the nodes, the transaction is kept in a block and then appended to a chain of blocks; thus, the name “blockchain”. Several variations of consensus algorithms have been proposed, each of which offers varying performance efficiency, security and reliability. Several surveys on consensus algorithms have been conducted in recent years (Nguyen and Kim, 2018; Mingxiao et al., 2017; Cachin and Vukolić, 2017; Bano et al., 2017; Alsunaidi and Alhaidari, 2019; Wang et al., 2019; Sharma and Jain, 2019; Sadek Ferdous et al., 2020). Several of these surveys are summarized in Table 1. Some of the popular consensus algorithms are explained below.

**Proof of Work (PoW)** (Nakamoto, 2008): This is the first consensus mechanism that gained widespread adoption due to its implementation in Bitcoin. In this algorithm, a new block is added to blockchain just after a complex mathematical puzzle is solved. This puzzle usually is a hashing function that involves finding a particular hash that is a sequence of consecutive 0's at the end of the hash. Finding the hash becomes difficult over time and requires immense computing capability. All nodes in the network try to solve the puzzle, i.e., finding the hash. Any node that finds a solution to the complex puzzle,

disseminate the solution to the network, which is then verified by the other nodes. The node that solves the puzzle receives rewards for its efforts. In the case of the Bitcoin network, the node will receive Bitcoin as a reward. PoW has become quite unpopular because of performance bottleneck (low transactions rate) and sustainability issues (high power consumption).

**Proof of Stake (PoS)** (Bitcoin forum 2011): PoS is the most popular alternative to PoW. The number of stakes a node holds in the network characterizes this consensus mechanism. This mechanism does not require huge computational power as in PoW. Unlike PoW where the nodes earn the reward by solving a complex puzzle, PoS let the nodes to acquire the coins upfront before any transaction even happens. After the coins have been acquired by the node, the nodes earn the reward by helping in the transaction, i.e., adding new blocks to the blockchain. In PoS there is no race to solve the complex puzzle; instead, the node that has more stakes has higher chances of creating a new block in the blockchain. After the block has been created, the way it is added to the blockchain may vary.

**Proof of Activity (PoA)** (Ethcore): PoA can be considered to be the combination of PoW and PoS. In this mechanism, the process of adding a block to the blockchain starts in a fashion similar to the PoW. The nodes race to solve the computationally complex mathematical problem. However, the new block created by the winning node does not constitute a transaction; it merely contains the header (relevant information of the block) and the address of the node that created the block first. To add to the blockchain, the system switches to PoS, where the nodes with higher stakes are sent the blocks to be signed. Once enough signs required by the system are reached, the transaction value is distributed among the node that created the block and the signing nodes.

**Practical Byzantine Fault Tolerance (PBFT)** (Castro and Liskov, 1999): PBFT is proposed to make or reach a decision in a

distributed network, such as blockchain, in the presence of the malicious nodes. The concept of PBFT was proposed in 1999 (Castro and Liskov, 1999). In PoW, the node that solves the puzzles gets to add the block, whereas, in PoS, the node that put in the highest stake has the chances to add the block. In PBFT, no such special requirement is needed, the block that will be added to the chain would be the one validated by the majority of the nodes. PBFT results in higher throughput (transaction rate), scalability and low power consumption. In PBFT, a proposer (block generator) is chosen in a round-robin manner. This proposer then collects the transaction details to create a block and broadcasts the created block to the network. Upon receiving the block, nodes will validate the received blocks and commit the block to the blockchain.

*Istanbul Byzantine Fault Tolerance (IBFT)* (Istanbul, 2020): IBFT is similar to PBFT with an added functionality of the way nodes are added or removed from the validation group. IBFT uses pool validating nodes to determine whether the proposed block is suitable for addition to the chain. One of the validator nodes is chosen randomly to create a block after committing the previous block. If the majority (66% of the nodes) of validating nodes approve the block, then it is added to the blockchain. If a block cannot be added to the chain, the proposer is changed, and the process starts again. The nodes in the validation group may change over time. The new nodes are added via a voting mechanism where members can be added or removed through a majority vote. Once a node reaches majority votes, they immediately join/leave the validator group. Each block in IBFT each block goes through several voting rounds by the set of validators to reach a bilateral settlement before adding it to the blockchain.

*Raft* (Ongaro and Ousterhout, 2014): This represents another distributed consensus protocol that is utilized in blockchain. The defining feature of this protocol is that each node is assigned three states, namely leader, follower or candidate. The leader node is chosen by voting rule and is allowed to append-only operation. During normal operations, there is only one leader node and other nodes are followers. Followers simply respond to requests from leaders and candidate nodes. In Raft, the time is divided into arbitrary length term. At the beginning of each term, a leader is selected via an election mechanism. One or more candidate nodes attempt to become the leader. If a candidate node receives majority votes, then it will operate as a leader until the term expires. The Raft is envisioned for use in a comparatively small cluster of nodes. In Raft, the followers blindly trust their leader and follow what is instructed by the leader. Raft consensus results in significant storage savings because it does not create a block if there are no transactions. This is significantly different from other consensus algorithms where blocks containing zero transactions can be added.

*LibraBFT* (Baudet et al., 2019): This consensus algorithm was proposed by Facebook. It is similar to other BFT-based networks, such as PBFT and IBFT, where the nodes acting as validators decide which block can be appended to the chain. However, LibraBFT reduces the complexity associated with the PBFT consensus protocol. It makes sure that all honest nodes agree on the block and continuously adds the block to the chain. LibraBFT protocol proceeds in rounds and a leader is chosen among the set of validators in each round. The leader

broadcasts new block requests to all validator nodes and waits for enough votes before proceeding. Once enough votes are collected, it forms a Quorum Certificate (QC). This QC is broadcasted to all the validators for verification. After verifying the QC, the validators commit the proposed block to its local storage and the round finishes.

## 2.2. Blockchain platforms

Bitcoin (Nakamoto, 2008) is the first and most popular blockchain platform which facilitates a reliable, cheap, and fast mechanism to perform digital financial transactions without the need for a central bank. Bitcoin enables the implementation of smart contracts using a scripting language. However, the limitation of scripting language makes Bitcoin an inappropriate choice to create smart contracts.

Ethereum (Buterin, 2013) is a blockchain platform that has had a profound effect on the evolution of blockchain technology in recent years. Ethereum has proven itself as the most well-known platform to support smart contracts using a built-in scripting language, named Solidity. The simplicity of creating smart contracts using Ethereum enables the blockchain technology to be applied not only on cryptocurrencies but also on different application domains. This has made Ethereum as the most popular solution to develop blockchain applications. Parity and Geth are the two leading Ethereum clients' implementation.

Hyperledger Fabric (Androulaki et al., 2018) is an open-source blockchain platform, developed under the Linux foundation to be used in enterprise context. Hyperledger Fabric facilitates the creation of smart contracts using general-purpose scripting languages such as Go, Java and Node.js rather than constrained domain-specific languages (DSL). This simplifies the implementation of blockchain solutions in enterprises, as there is no need for developers to learn a new scripting language for developing smart contracts. Another key attribute of Hyperledger Fabric is to support the pluggable consensus protocols that provide the platform with the ability to be tailored to specific industry use cases.

MultiChain (MultiChain Private Blockchain - Whitepaper 2015) is a blockchain platform, which enables the users to set up private blockchains in organizations with speed. It provides a command-line interface for network interaction and extends the core functionality of Bitcoin API using a simple API. MultiChain allows different clients such as C#, Go, Java, Node.js, PHP, Python and Ruby to interact with the network through JSON-RPC API.

Lisk (The Lisk Protocol 2017) provides an open-source blockchain platform that simplifies the creation and deployment of decentralized applications in JavaScript. It deploys Delegated Proof of Stake (DPoS) as its consensus protocol. Lisk enables the users to establish personal blockchain, called sidechain, which can be simply developed and tailored with Lisk tools. Sidechain stores all the data generated by a decentralized application. With a special focus on accessibility and usability, Lisk intends to be the most prominent platform for developing blockchain applications.

Quorum (Quorum Whitepaper 2016) is an Ethereum-based blockchain platform established to simplify the development of Ethereum's blockchain applications in enterprises. Quo-



rum is an ideal solution for applications for which transactions' processing time and throughput are major concerns. The functionality of Quorum is almost the same as Ethereum. However, it has few differences such as permissions management of both network and peer, improved transaction and contract privacy, voting-based consensus protocols and higher performance.

HydraChain (Ethereum and Bot, 2015) is an open-source extension of Ethereum blockchain platform that supports the development and deployment of permissioned distributed ledgers. HydraChain is entirely compatible with the Ethereum protocol and allows the creation of smart contracts using Python. The key feature of HydraChain is that it enables different components of the system to be easily customized based on customer needs. It supports many tools, which enable reducing development time while improving debugging capabilities.

Libra Blockchain (Amsden et al., 2019) provides an open-source, programmable, and decentralized database that aims at offering a powerful and efficient platform to accommodate the daily financial demands of all people around the world. It facilitates the creation of smart contracts using a new and user-friendly programming language, called Move. In line with its main goal, which is to support billions of people, high scalability, extreme security, and flexibility are amongst the highest-priority features of this platform.

### 3. Empirical studies on performance evaluation of blockchain platforms

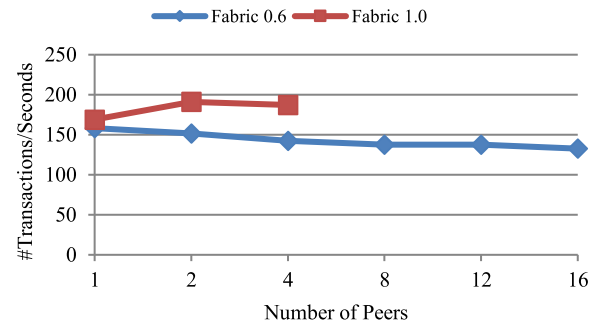
In order to answer the first research question (RQ1), we conducted an exhaustive review of the blockchain literature using popular scientific research databases such as Web of Science, Scopus, IEEE Xplore, Google Scholar, etc. We selected "performance" AND ("evaluation" OR "analysis") AND ("blockchain" OR "blockchains") as a search query string, covering the time span between 2015 and Dec 2019. The search process resulted in locating more than 1000 articles, focusing on the performance evaluation of blockchain platforms, and we used the following inclusion/exclusion criteria to further filter the located articles.

- We included studies that focused on the empirical performance evaluation of permissioned blockchain platforms. In other words, the performance evaluation of permissionless (public) blockchain platforms was not considered.
- We considered empirical studies that compared the performance of at least two permissioned blockchain platforms.
- We excluded studies that explored the performance evaluation of blockchain platforms through analytical modeling.

Eventually, we ended up with seven articles that will be examined in the following sub-sections.

#### 3.1. Hyperledger fabric 0.6 vs. fabric 1.0

In (Nasir et al., 2018), two experiments were conducted to have a comparative evaluation of two versions of Hyperledger Fabric



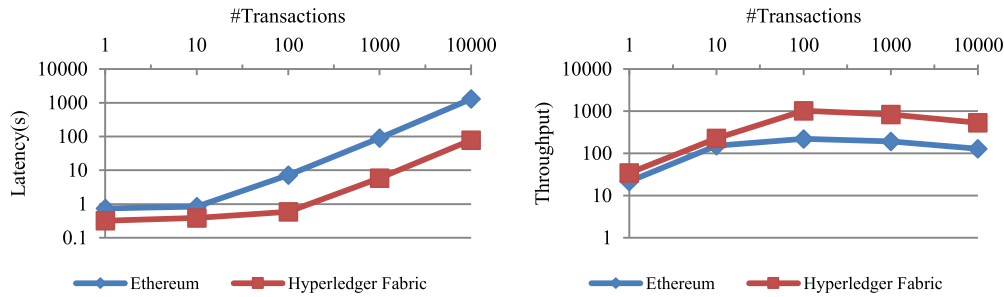
**Fig. 1 – Scalability (increasing the number of peers) of Hyperledger Fabric 1.0 and 0.6 (adapted from (Nasir et al., 2018)).**

ric, Fabric v0.6 and Fabric v1.0 to assess different metrics including execution time, latency, throughput, and scalability. In the first experiment, the execution time, latency, and throughput of the two platforms were analyzed on a single-peer network by changing the number of transactions up to 10,000 for each platform. The results proved the superiority of Hyperledger Fabric v1.0 over Hyperledger Fabric v0.6 across the three assessment metrics. In the second experiment, the same metrics were measured but on a multiple-peer network to measure the scalability of the two platforms. Here also Hyperledger Fabric v1.0 outperformed Hyperledger Fabric v0.6, although increasing the number of peers up to 20 nodes resulted in longer execution time, more latency, and lower throughput for both platforms. This study also reported that the highest number of peers that Hyperledger Fabric v1.0 and Hyperledger Fabric v0.6 can handle is 4 and 16 peers, respectively, for the execution of 10,000 transactions (see Fig. 1). However, this number would be 26 and 16 peers for Hyperledger Fabric v1.0 and Hyperledger Fabric v0.6, respectively, when executing 1000 transactions. This shows that Hyperledger Fabric v1.0 is more scalable for a lower number of transactions (up to 1000) and Hyperledger Fabric v0.6 is a better choice in terms of scalability for higher number of transactions (up to 10,000).

**Issues.** Consensus algorithms play a critical role in addressing the performance of blockchain platform. Given this, a detailed discussion on the type of consensus algorithm used in experiments and its effects on the performance results are overlooked in this research.

#### 3.2. Hyperledger fabric 0.6 vs. ethereum 1.5.8

In other research (Pongnumkul et al., 2017), execution time, latency, and throughput of two well-known blockchain platforms, Hyperledger Fabric 0.6 and Ethereum 1.5.8, have been assessed while the number of transactions was increased from 1 to 10,000. The results highlighted that Hyperledger Fabric attains lower execution time, higher throughput and lower latency compared to Ethereum when the number of transactions are increased up to 10,000. However, it is reported that Ethereum is able to execute a higher number of concurrent transactions compared to Hyperledger Fabric (i.e., 50,000 and 20,000, respectively).



**Fig. 2 – Latency (left) and throughput (right) of Ethereum and Hyperledger Fabric (taken from (Hao et al., 2018)).**

**Issues.** The presented results in this research are obtained in a situation where consensus protocol is discarded by configuration. In other words, the results only highlight the difference between the execution layers of two platforms. Therefore, the impact of the consensus algorithm on the performance evaluation of the two platforms is not investigated in this research. The experiments were executed on a single-node network. Another drawback is concerned with the lack of scalability evaluation of both platforms. In this sense, the effects of varying the number of nodes on performance metrics of both platforms are not addressed in this research.

### 3.3. Ethereum 1.7.3 vs. hyperledger fabric 1.0

In (Hao et al., 2018), the authors compared two permissioned blockchain platforms, namely Ethereum 1.7.3 and Hyperledger Fabric 1.0, in terms of throughput and latency. The major difference between this work and (Pongnumkul et al., 2017) is that different consensus algorithms, i.e., PBFT and PoW, were taken into consideration to explore their impact on performance evaluation of both platforms. The experiments were performed on a multiple-node network, including four servers and four clients. The results, as sketched in Fig. 2, indicated that Hyperledger Fabric outperforms Ethereum in terms of average throughput and latency under different transaction rates (from 1 to 10,000 transactions). These results are in line with the results presented in (Vukolić, 2015) as BFT-based blockchain platform, i.e., Hyperledger Fabric, shows a better performance compared to the PoW-based blockchain platform, i.e., Ethereum.

**Issues.** The execution time of transactions, as one of the important performance metrics, is not measured and discussed in this research. Moreover, the scalability of both platforms is not explored in the presence of consensus algorithms. It is worth scrutinizing the impacts of changing the number of nodes on the performance metrics of both platforms. The ability of both platforms to handle concurrent transactions is also not addressed in this work.

### 3.4. Ethereum 1.4.18 vs. parity 1.6 vs. hyperledger fabric 0.6

In (Dinh et al., 2017), the authors proposed a performance benchmarking framework, called BLOCKBENCH, to compare three permissioned Blockchain platforms, namely Ethereum

1.4.18, Parity 1.6, and Hyperledger Fabric 0.6. The evaluation was based on measuring throughput, latency, scalability, and fault-tolerance. The experimental results highlighted that Hyperledger outperforms Ethereum and Parity in terms of throughput. However, Parity achieved the lowest latency while Ethereum showed the highest. The scalability was measured by fixing the transaction rate while increasing the number of servers and clients. The results illustrated that throughput and latency of Parity remained at the same level where for Ethereum both throughput and latency reduced continually over 8 servers. Surprisingly, Hyperledger stopped responding over 16 servers. From fault tolerance point of view, it was described that Ethereum and Parity remain unaffected against failing four servers while Hyperledger Fabric fails to generate any more block once failure occurs. In summary, although the similarities and differences among three blockchain platforms are well-investigated in this research, the authors concluded that the investigated blockchain platforms are not able to show a good performance in case of large-scale data processing workloads.

**Issues.** Although the effects of changing the number of nodes on performance evaluation of three platforms were well-explored in this research, the impacts of modifying the number of transactions on performance metrics were not addressed. Furthermore, the transaction execution time and the ability of both platforms to handle concurrent transactions are ignored to be measured and reported in this research.

### 3.5. Geth vs. parity

Another research (Rouhani and Deters, 2017) analyzed two popular Ethereum clients, called Parity and Geth, to determine which one is faster in processing a different number of transactions, ranging from 1000 to 10,000 transactions. The results demonstrated that Parity is able to process transactions, on average 89.9%, faster than Geth.

**Issues.** No information and discussion are given about the consensus algorithm deployed in the experiments. Latency and throughput of transactions are not measured. Although the influence of changing the amount of RAM on execution times of both Parity and Geth has been explored in this research, the impacts of changing the number of nodes on execution times of both platforms were not investigated. The ability of both platforms to handle concurrent transactions is also not taken into consideration.

### 3.6. Quorum (Raft-Based vs. ibft-based)

In another research (Baliga et al., 2018), the performance characteristics of another blockchain platform, called Quorum, have been investigated through an empirical study. In practice, the focus was on comparing the performance of two versions of Quorum, Raft deployment and IBFT deployment, in terms of throughput and latency. The experiments were performed on a three-node and four-node network for Raft-based and IBFT-based Quorum respectively, based on the least requirements of both platforms. After executing the experiments and collecting the results, the authors deduced that IBFT-based Quorum has higher latency compared to Raft-based Quorum. Unexpectedly, the difference was not significant in terms of throughput. Nevertheless, for input transaction rates of 1650 per second and above, Raft-based Quorum outperformed IBFT-based Quorum to a small extent and for lower transaction rates IBFT-based Quorum performed slightly better than Raft-based Quorum.

**Issues.** This research failed to notice how increasing the number of peers would affect the throughput and latency of both platforms. The number of peers remained fixed throughout the experiments. In other words, scalability and its impact on both platforms are not addressed in this research. The number of transactions also kept rather low compared to other similar empirical studies. There is no information behind making such a design decision. Moreover, the ability of both platforms to handle concurrent transactions and transaction execution time are overlooked in this study.

### 3.7. Libra vs. hyperledger fabric 1.4.1

In (Zhang et al., 2019), the authors performed an experimental study to evaluate the performance and scalability of two permissioned blockchain platforms, called Libra and Hyperledger Fabric 1.4.1. The performance evaluation was based on measuring two properties, execution time and throughput. A four-node network was utilized to run experiments. The results, as sketched in Fig. 3, ascertained the superiority of Hyperledger Fabric over Libra against both properties. However, in terms of scalability, it has been shown that increasing the number of peers, from 1 to 16, resulted in execution time increment and throughput reduction for both platforms. By analyzing the performance and scalability of Libra blockchain platform, the authors figured out that Libra in its current version is still

far from a decent level to satisfy the requirements of a given heavy-load real application.

**Issues.** Although this research applied a rigorous methodology to compare both platforms under different circumstances, measuring performance metrics of both platforms beyond 16 nodes are not addressed. Tackling such a point could certainly strengthen the findings of this study.

## 4. Comparison of the empirical studies

To answer the second research question (RQ2), we performed a comparative analysis of the empirical studies introduced in the preceding section. This kind of analysis would enable us to provide a broad overview of all empirical studies by discovering the similarities and differences among those studies.

### 4.1. Comparison criteria

Before presenting the results of our analysis, it is initially important to describe the comparison criteria we have identified and used in order to characterize the empirical studies. These criteria are described below.

**Blockchain platforms.** This represents the blockchain platforms that have been empirically evaluated by a given experimental study.

**Blockchain type.** As previously discussed, there are two categories of blockchain platforms (Sadek Ferdous et al., 2020; Vukolić, 2017):

**Permissionless blockchain** is a fully decentralized network that allows any peer to freely join the network and participate in a transaction and consensus process without any kind of permission.

**Permissioned blockchain** requires every unknown peer to be verified and permissioned by a set of known peers before joining and accessing to a blockchain network.

**Consensus algorithm.** It represents a specific process that a blockchain platform under evaluation might have been deployed in order to reach agreement among peers before letting any block to be added into a blockchain network. Currently, various types of consensus algorithms exist, each of which has its own fundamental process. Further details about different types of consensus algorithms are given in Section 2.1.

**Performance metrics.** The performance of a given blockchain platform can be measured in terms of execution time, latency, and throughput.

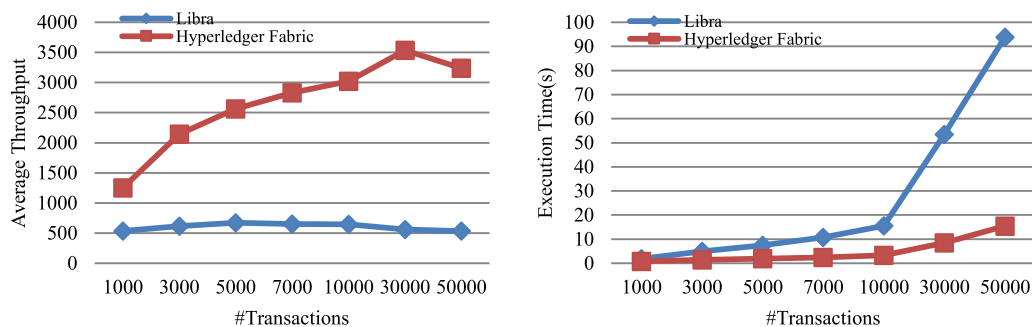


Fig. 3 – Throughput (left) and execution time (right) of Libra and Hyperledger Fabric (adapted from (Zhang et al., 2019)).

**Execution time** is the amount of time required by a blockchain platform to successfully confirm and execute a transaction.

**Latency** is the amount of time taken by a blockchain platform to respond to a transaction.

**Throughput** is the number of transactions that can be executed successfully by a blockchain platform per second.

**Evaluation workload.** It represents the number of transaction requests sent to a given blockchain platform while monitoring its effect on the performance metrics.

**Network type.** It determines whether the performance metrics of a given blockchain platform have been measured on a single or multi-node network.

**Scalability.** The scalability of a given blockchain platform is measured by increasing the number of nodes while still measuring the performance metrics.

**Fault tolerance.** The fault tolerance of a given blockchain platform is measured as to how a node failure can affect the performance metrics. In other words, it determines how resilient a blockchain platform is against failures.

**Evaluation framework.** It describes which testing framework was deployed in order to produce workload and collect performance results. It can be classified into three categories:

**Hyperledger Caliper** (Hyperledger Caliper 2020) is a performance benchmarking tool, developed by the Linux Foundation that can be used to quantify different performance properties, such as success rate, latency, throughput, and resource consumption, of multiple blockchain solutions. The key component of Hyperledger Caliper's architecture is the adaption layer, which enables practitioners to customize this benchmarking tool by adding a plugin to accommodate the requirements of a particular blockchain implementation.

**Blockbench** (Dinh et al., 2017) is a performance testing framework for evaluating and comparing the performance of different permissioned blockchain platforms. The extensible architecture of Blockbench allows the integration of any permissioned blockchain platform into it using APIs. It measures the performance of a blockchain platform in terms of latency, throughput, fault tolerance, and scalability.

**Synthetic application** is involved with deploying a synthetic application and evaluation framework for measuring the performance of permissioned blockchain platforms. In other words, no specific performance benchmarking tool has been utilized for testing purposes, which makes it cumbersome for the other researchers to replicate such empirical studies.

**Maximum concurrent transactions.** It specifies the capacity of a certain blockchain platform to handle concurrent transactions.

• **Impact of different consensus algorithms:** Consensus algorithms play a critical role in justifying the performance characteristics of different permissioned blockchain platforms. From Table 3, it is perceptible that Hyperledger Fabric consistently outperforms Ethereum in terms of performance metrics, as also demonstrated in (Hao et al., 2018) and (Dinh et al., 2017). This can be justified by the underlying consensus algorithm deployed by each platform, i.e., PBFT for Hyperledger Fabric and PoW for Ethereum. This justification is also in consonance with the results given in (Vukolić, 2015) where it has been proved that BFT-based consensus algorithms show better performance compared to PoW-based ones. It is worth highlighting that even with the exclusion of the consensus algorithm and only by assessing the execution layer of both platforms, Hyperledger Fabric still performs better than Ethereum (Pongnumkul et al., 2017). This implies the fact that the smart contract infrastructure of Hyperledger Fabric is developed more efficiently than Ethereum.

As discussed earlier in Section 3.4, Parity has the lowest latency and is more fault-tolerant compared to Hyperledger Fabric and Ethereum. The main reason behind this is the deployment of PoA consensus algorithm in Parity as it has been proposed to offer better performance and fault tolerance compared to other BFT-based consensus algorithms (De Angelis et al., 2018). Nevertheless, due to its fixed block generation rate, Parity's throughput appears to be less than the other two platforms.

- **Better performance of Raft-based Quorum over IBFT-based Quorum** in terms of latency (which is discussed in Section 3.6) is also rooted in the consensus algorithm deployed in each version of Quorum blockchain platform. Raft, as a variation of Crash Fault Tolerance (CFT) consensus algorithm, is designed and developed to process transactions faster than BFT-based consensus algorithms like IBFT. The same reason justifies the superiority of Hyperledger Fabric 1.4.1 over Libra (as discussed in Section 3.7) in terms of execution time and throughput as a CFT-based consensus algorithm, i.e., Raft, has been deployed in Hyperledger Fabric 1.4.1 while a BFT-based consensus algorithm, i.e., LibraBFT, is exploited in Libra blockchain platform.
- **Influence of varying number of peers:** Not many empirical studies have addressed how performance characteristics of blockchain platforms can be affected by increasing the number of peers in the network. From Table 2, we can figure out that only studies (Nasir et al., 2018; Dinh et al., 2017), and (Zhang et al., 2019) have undertaken such issue. An in-depth analysis of those empirical studies implies the fact that growing the number of peers will negatively affect the performance of blockchain platforms. Even in the worst case, it causes some platforms such as Hyperledger Fabric to stop responding beyond a certain number of peers. This is in contrast with the hypothesis that increasing the number of peers would be a logical solution to improve performance. This is mainly because Hyperledger Fabric is a communication bound network and adding more number of peers inflicts more network communication overheads.

## 5. Results and discussion

The comparison results of the identified empirical studies with respect to the comparison criteria are summarized in Table 2. The results of performance metrics are also elaborated in Table 3.

Based on the results characterized in Table 2 and Table 3, here, we discuss the different factors that lead to different performance results for different blockchain platforms.



**Table 2 – Comparison of the empirical studies addressing performance evaluation of permissioned blockchain platform.**

Paper	Blockchain Platforms	Blockchain Type	Consensus Algorithm	Performance Metrics			Evaluation Workload	Network Type	Scalability (node vs client)	Fault tolerance	Evaluation Framework	Maximum Concurrent Transactions
				Execution time	Latency	Throughput						
(Nasir et al., 2018)	Hyperledger Fabric 0.6	Permissioned	Not reported	✓	✓	✓	Up to 10,000	Single-node	Node (up to 20 nodes)	×	Hyperledger Caliper	20,000 (for single-node)
(Pongnumkul et al., 2017)	Hyperledger Fabric 1.0	Permissioned	Disabled	✓	✓	✓	Up to 10,000	Single -node	×	×	Synthetic application	20,000 50,000
(Hao et al., 2018)	Hyperledger Fabric 0.6	Permissioned	PBFT	×	✓	✓	Up to 10,000	Multiple nodes (4 clients, 4 Servers)	×	×	Synthetic application	×
(Dinh et al., 2017)	Ethereum 1.7.3	Permissioned	PoW	×	✓	✓	Up to 1000	Multiple nodes (8 clients, 8 servers)	Node and clients (up to 32 for both)	✓	BlockBench	×
(Rouhani and Deters, 2017)	Hyperledger Fabric 0.6	Permissioned	PBFT	×	✓	✓	Up to 10,000	Single - node	×	×	Not reported	×
(Baliga et al., 2018)	Ethereum 1.4.18	Permissioned	PoW	×	✓	✓	Up to 2100	Multiple nodes (3 for RAFT and 4 for IBFT)	×	×	Hyperledger Caliper	×
(Zhang et al., 2019)	Parity 1.6	Permissioned	Not reported	✓	×	×	Up to 50,000	Multiple-nodes (4 Servers)	Node (up to 20 nodes)	×	S ynthetic application	×
	Geth	Permissioned	Not reported	✓	×	×	Up to 50,000	Multiple-nodes (4 Servers)	Node (up to 20 nodes)	×	S ynthetic application	×
	Quorum (RAFT deployment)	Permissioned	Raft	×	✓	✓	Up to 2100	Multiple nodes (3 for RAFT and 4 for IBFT)	×	×	Hyperledger Caliper	×
	Quorum (IBFT deployment)	Permissioned	IBFT	×	✓	✓	Up to 2100	Multiple nodes (3 for RAFT and 4 for IBFT)	×	×	Hyperledger Caliper	×
	Hyperledger Fabric 1.4.1	Permissioned	Raft	✓	×	✓	Up to 50,000	Multiple-nodes (4 Servers)	Node (up to 20 nodes)	×	S ynthetic application	×
	Libra	Permissioned	LibraBFT	✓	×	✓	Up to 50,000	Multiple-nodes (4 Servers)	Node (up to 20 nodes)	×	S ynthetic application	×

**Table 3 – Summary of performance metrics results of the empirical studies along with their experimental infrastructures.**

Paper	Execution Time	Latency	Throughput	Experimental Infrastructure
(Nasir et al., 2018)	Fabric 1.0 < Fabric 0.6	Fabric 1.0 < Fabric 0.6	Fabric 1.0 > Fabric 0.6	HPC server with Intel(R) Xeon(R) 24 core CPU E5-2690, 2.60 GHz, 64 GB RAM, running Ubuntu 16.04
(Pongnumkul et al., 2017)	Hyperledger Fabric < Ethereum	Hyperledger Fabric < Ethereum	Hyperledger Fabric > Ethereum	Amazon AWS EC2 with Intel E5-1650 8 core CPU, 15GB RAM, running Ubuntu 16.04
(Hao et al., 2018)	×	Hyperledger Fabric < Ethereum	Hyperledger Fabric > Ethereum	Yahoo Cloud System Benchmark, 8GB RAM, 128 G SSD HD, running Ubuntu16.04
(Dinh et al., 2017)	×	Parity < Hyperledger Fabric < Ethereum	Hyperledger Fabric > Ethereum > Parity	48-node commodity cluster with E5-1650 3.5 GHz CPU, 32GB RAM, 2 TB HD, running Ubuntu 14.04
(Rouhani and Deters, 2017)	Parity < Geth	×	×	Core i7-6700 CPU, 24GB RAM
(Baliga et al., 2018)	×	Raft-based < IBFT-based	IBFT-based > Raft-based (for lower transaction rates) Raft-based > IBFT-based (for higher transaction rates)	8 vCPUs (4 cores at 3.6 GHz), 16 GB RAM, running Ubuntu 14.04
(Zhang et al., 2019)	Hyperledger Fabric < Libra	×	Hyperledger Fabric > Libra	2 Intel E5-2680 v4 CPU (14 cores), 384 G RAM, running Ubuntu 18.04

Only Parity's performance appears to be unaffected against increasing the number of peers (Dinh et al., 2017), because of its fixed block generation rate.

- *Impact of evaluation workload:* Increasing the number of input transactions to be processed by a blockchain platform affects the performance metrics. In general, it has to be said that longer execution time and more latency are also caused by growing the number of transactions. However, the scenario is different for throughput where increasing the number of transactions usually results in higher throughput but up to a certain point from which throughput also declines as the number of transactions grows.

Overall, after a thorough analysis of the empirical studies, it has to be said that, although each blockchain platform has its own performance characteristics under different circumstances and one may outperform the other in terms of a specific performance metric, the evaluated blockchain platforms with their current performance specifications are still faraway to accommodate the needs of large-scale data processing systems.

## 6. Open issues

Based on our findings reported in the preceding sections, we will now describe several potential research opportunities (addressing RQ3).

- *Further empirical performance studies:* Performance of blockchain platforms has been identified as a major issue that needs to be investigated more by blockchain research community (Yli-Huumo et al., 2016; Swan, 2015; Jiang et al., 2020). Although the significance of considering performance evaluation of blockchain platforms, it can be observed, from Table 2, that only a mere handful of research studies have been dedicated to addressing this important challenge. This shows that the current body of

research in this area is still at its nascent stage and further empirical studies need to be conducted in order to empirically evaluate the performance of recently established blockchain platforms.

- *Performance evaluation of other permissioned blockchain platforms:* From Table 2, we can easily distinguish that only a limited number of permissioned blockchain platforms have been empirically compared together in terms of performance. From the other side, an in-depth review of the blockchain literature implies that blockchain platforms are evolving rapidly. Some popular permissioned blockchain platforms, other than those were investigated in Table 2, are Corda (Brown, 2018), MultiChain (MultiChain Private Blockchain - Whitepaper 2015), Hyperledger Sawtooth (Olson et al., 2018), HydraChain (Ethereum and Bot, 2015), BigchainDB (McConaghy et al., 2016), Ripple (Schwartz et al., 2014), Hyperledger Iroha (Schwartz et al., 2014), OpenChain (Openchain 2020), Kadena (Martino, 2016), and so forth. As the number of blockchain permissioned platforms grows, it is essential to empirically evaluate the key performance properties of each platform before applying on a real use case. This gap creates an interesting line of research for scholars to conduct more empirical performance analysis on the aforementioned platforms. This kind of studies would also reveal the key differences that might exist among different consensus algorithms deployed in each blockchain platform. *Scalability:* Scalability, like performance, is considered as a significant non-functional requirement of any blockchain-based system. Despite its importance, only a few empirical studies measured the scalability of different blockchain platforms (see Table 2). This brings out the necessity for the blockchain research community to conduct more experiments to measure the scalability of different permissioned blockchain platforms. Those few studies, (Nasir et al., 2018; Pongnumkul et al., 2017), and (Zhang et al., 2019) also demonstrate that the evaluated blockchain platforms are suffering from

scalability issues. Although some mechanisms such as sharding, directed acyclic graph, side chain, and off-chain have been proposed in recent years to address the scalability issue of blockchain platforms, it is essential for researchers to conduct rigorous empirical studies to explore and

- report the effects of such solutions on the performance and scalability of permissioned blockchain platforms.
- **Block size:** Block size, as the number of transactions that can be fitted into a block, has a significant impact on the performance of a blockchain platform (Dinh et al., 2017; Schäffer et al., 2019). Despite this, only one empirical work in Table 2 measured the impact of such a factor on performance characteristics of permissioned blockchain platforms, concluding that it has no effect on improving performance. In contrast, another study evidences the positive impact of block size on the performance of permissioned blockchain platforms (Schäffer et al., 2019). Hence, further empirical studies need to be carried out in order to provide more rigorous evidence on either advocating or negating the positive impact of block size on the performance of permissioned blockchain platforms.
- **Node configuration:** The other parameter that can affect the performance of a permissioned blockchain platform is the computational power of a node, which is called node configuration, such as RAM and CPU's amount and type (Schäffer et al., 2019). The effect of this parameter on performance is partially addressed in only one empirical study (Rouhani and Deters, 2017). In this sense, it is worth conducting more empirical studies to explore the effect of varied node configuration on the performance of different permissioned blockchain platforms.

## 7. Concluding remarks

In this research, we performed a detailed review of empirical studies that focused on the performance evaluation of different permissioned blockchain platforms, as performance is often understudied in the blockchain literature.

Using our proposed comparative framework (that comprises ten criteria), we were able to not only highlight the similarities and differences between different performance empirical studies but also identify the performance characteristics of each permissioned blockchain platform under different settings. This comparative framework also facilitates in the identification of key factors that could impact on performance on the different platforms (e.g., consensus algorithm, number of peers, and number of transactions).

An extension of this research is to empirically evaluate the performance of the MultiChain permissioned blockchain platform and other newer blockchain platforms, as well as addressing the open issues described in Section 6.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution statement

**Mohammad Dabbagh:** Conceptualization, Methodology, Investigation, Writing - original draft, Visualization, Validation. **Kim-Kwang Raymond Choo:** Conceptualization, Writing - review & editing. **Amin Beheshti:** Writing - review & editing. **Mohammad Tahir:** Writing - review & editing. **Nader Sohrabi Safa:** Visualization, Validation.

## REFERENCES

- Ali G, Ahmad N, Cao Y, Asif M, Cruickshank H, Ali QE. Blockchain based permission delegation and access control in Internet of Things (BACI). *Computers & Security* 2019;86:318–34.
- Alsunaidi SJ, Alhaidari FA. A Survey of Consensus Algorithms for Blockchain Technology. In: *2019 International Conference on Computer and Information Sciences (ICCIS)*. IEEE; 2019. p. 1–6.
- Amsden Z, Arora R, Bano S, Baudet M, Blackshear S, Bothra A, and Cabrera G, “The Libra Blockchain,” pp. 1–29, 2019. [Online]. Available: <https://developers.libra.org/docs/the-libra-blockchain-paper>
- Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, De Caro A, Muralidharan S. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: *Proceedings of the Thirteenth EuroSys Conference*. ACM; 2018. p. 30.
- Baliga A, Subhod I, Kamat P, & Chatterjee S. (2018). Performance evaluation of the quorum blockchain platform. *arXiv preprint arXiv:1809.03421*.
- Bano S, Sonnino A, Al-Bassam M, Azouvi S, McCorry P, Meiklejohn S, & Danezis G. (2017). Consensus in the age of blockchains. *arXiv preprint arXiv:1711.03936*.
- Baudet M, Ching A, Chursin A, Danezis G, Garillot F, Li Z, Malkhi D, Naor O, Perelman D, and Sonnino A, “State Machine Replication in the Libra Blockchain,” pp. 1–41, 2019. [Online]. Available: <https://developers.libra.org/docs/state-machine-replication-paper>
- Bitcoin forum, “Topic: proof of stake instead of proof of work,” 2011 [Online]. Available: <https://bitcointalk.org/index.php?topic=27787.0>.
- Brown RG. (2018). The corda platform: an introduction. Retrieved, 27, 2018.
- Buterin V, Ethereum white paper, 2013. Available online: <https://github.com/ethereum/wiki/wiki/White-Paper> (Accessed 2 April 2018).
- Cachin C, & Vukolić M. (2017). Blockchain consensus protocols in the wild. *arXiv preprint arXiv:1707.01873*.
- Castro M, Liskov B. Practical Byzantine fault tolerance. *OSDI 1999*;99(1999):173–86.
- Dabbagh M, Sookhak M, Safa NS. The Evolution of Blockchain: a Bibliometric Study. *IEEE Access*. 2019a.
- Dabbagh M, Kakavand M, Tahir M. Towards Integration of Blockchain and IoT: a Bibliometric Analysis of State-of-the-Art. In: *International Congress on Blockchain and Applications*. Springer; 2019b. p. 27–35.
- De Angelis S, Aniello L, Baldoni R, Lombardi F, Margheri A, & Sassone V. (2018). PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain.
- Dinh TTA, Wang J, Chen G, Liu R, Ooi BC, Tan KL. Blockbench: a framework for analyzing private blockchains. In: *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM; 2017. p. 1085–100.

- Esposito Christian, Santis Alfredo De, Tortora Genny, Chang Henry, Choo Kim-Kwang Raymond. Blockchain: a Panacea for Healthcare Cloud-Based Data Security and Privacy? *IEEE Cloud Comput.* 2018;5(1):31–7.
- Ethcore Parity: next generation ethereum browser. <https://ethcore.io/parity.html> [accessed on Mar 23, 2020].
- Ethereum and Bot B, Hydrachain open source code. 2015. Book <https://github.com/HydraChain/hydrachain>.
- Gatteschi V, Lamberti F, Demartini C, Pranteda C, Santamaría V. Blockchain and smart contracts for insurance: is the technology mature enough? *Future Internet* 2018;10(2):20.
- Hao Y, Li Y, Dong X, Fang L, Chen P. Performance analysis of consensus algorithm in private blockchain. In: 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE; 2018. p. 280–5.
- Declan Harty. <https://www.bloomberg.com/news/articles/2016-06-23/finance-firms-seen-investing-1-billion-in-blockchain-this-year> (accessed on 25 October 2018).
- Hyperledger Caliper: <https://www.hyperledger.org/use/caliper>, [accessed 10 August 2020].
- Hyperledger Iroha: <https://www.hyperledger.org/projects/iroha>, accessed 30 April 2020.
- IBM, “10 Key Marketing Trends for 2017 and Ideas for Exceeding Customer Expectations”. 2017 Available online: <https://www.ibm.com/downloads/cas/XKBEABLN>
- Istanbul BFT <https://github.com/ethereum/EIPs/issues/650> [Online; accessed Mar 12, 2020].
- Jiang L, Chang X, Liu Y, Mišić J, Mišić VB. Performance analysis of Hyperledger Fabric platform: a hierarchical model approach. *Peer-to-Peer Networking and Applications* 2020:1–12.
- Kshetri Nir. 1 Blockchain’s roles in meeting key supply chain management objectives. *Int J Inf Manage* 2018;39:80–9.
- Makhdoom I, Zhou I, Abolhasan M, Lipman J, Ni W. PrivySharing: a blockchain-based framework for privacy-preserving and secure data sharing in smart cities. *Computer. Secur.* 2020;88.
- Martino W. (2016). Kadena: the first scalable, high performance private blockchain. Kadena, Okinawa, Japan, Tech. Rep.
- McConaghy T, Marques R, Müller A, De Jonghe D, McConaghy T, McMullen G, ... & Granzotto, A. (2016). Bigchaindb: a scalable blockchain database. white paper, BigChainDB.
- Mingxiao D, Xiaofeng M, Zhe Z, Xiangwei W, Qijun C. A review on consensus algorithm of blockchain. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE; 2017. p. 2567–72.
- MultiChain Private Blockchain - Whitepaper, 2015. Available online: <https://www.multichain.com/download/MultiChain-White-Paper.pdf> (Accessed 11 March 2020).
- Nakamoto, Satoshi. “Bitcoin: a peer-to-peer electronic cash system.” (2008).
- Nasir Q, Qasse IA, Abu Talib M, Nassif AB. Performance Analysis of Hyperledger Fabric Platforms. *Security and Communication Networks*; 2018. 2018.
- Nguyen GT, Kim K. A Survey about Consensus Algorithms Used in Blockchain. *J Inf. process. sys.* 2018;14(1).
- Olson K, Bowman M, Mitchell J, Amundson S, Middleton D, Montgomery C. Sawtooth: An Introduction. The Linux Foundation; 2018. Jan.
- Ongaro D, Ousterhout J. In search of an understandable consensus algorithm. In: 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14); 2014. p. 305–19.
- Openchain: <https://www.openchain.org/>, accessed 30 April 2020.
- Pongnumkul S, Siripanpornchana C, Thajchayapong S. Performance analysis of private blockchain platforms in varying workloads. In: 2017 26th International Conference on Computer Communication and Networks (ICCCN). IEEE; 2017. p. 1–6.
- Quorum Whitepaper, 2016. Available online: <https://github.com/jpmorganchase/quorumdocs/blob/master/Quorum%20Whitepaper%20v0.1.pdf>. (Accessed 1 February 2019).
- Rouhani S, Deters R. Performance analysis of Ethereum transactions in private blockchain. In: 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). IEEE; 2017. p. 70–4.
- Sadek Ferdous M, Javed Morshed Chowdhury M, Hoque MA, & Colman A. (2020). Blockchain Consensus Algorithms: a Survey. arXiv, arXiv-2001.
- Schäffer M, di Angelo M, Salzer G. Performance and scalability of private Ethereum blockchains. In: International Conference on Business Process Management. Cham: Springer; 2019. p. 103–18.
- Schwartz D, Youngs N, Britto A. The Ripple Protocol Consensus Algorithm, 5. Ripple Labs Inc White Paper; 2014.
- Sharma K, Jain D. Consensus Algorithms in Blockchain Technology: a Survey. In: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE; 2019. p. 1–7.
- Sharples M, Domingue J. The blockchain and kudos: a distributed system for educational record, reputation and reward. In: European Conference on Technology Enhanced Learning. Springer; 2016. p. 490–6.
- Su Q, Zhang R, Xue R, Li P. Revocable Attribute-Based Signature for Blockchain-Based Healthcare System. *IEEE Access* 2020;8:127884–96. doi:10.1109/ACCESS.2020.3007691.
- Swan M. Blockchain: Blueprint for a New Economy. O’Reilly Media, Inc.; 2015.
- Tahir M, Habaebi MH, Dabbagh M, Mughees A, Ahad A, Ahmed KI. A Review on Application of Blockchain in 5 G and Beyond Networks: taxonomy, Field-Trials, Challenges and Opportunities. *IEEE Access* 2020;8:115876–904. doi:10.1109/ACCESS.2020.3003020.
- The Lisk Protocol, 2017. Available online: <https://docs.lisk.io/docs/the-liskprotocol>. (Accessed 1 February 2019).
- Vukolić M. The quest for scalable blockchain fabric: proof-of-work vs. BFT replication. In: International workshop on open problems in network security. Springer; 2015. p. 112–25.
- Vukolić M. Rethinking permissioned blockchains. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts; 2017. p. 3–7.
- Wang W, Hoang DT, Hu P, Xiong Z, Niyato D, Wang P, Kim DI. A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access* 2019;7:22328–70.
- Yli-Huoma J, Ko D, Choi S, Park S, Smolander K. Where is current research on blockchain technology?—A systematic review. *PLoS ONE* 2016;11(10).
- Zhang Jie, Xue Nian, Huang Xin. “A secure system for pervasive social network-based healthcare. *IEEE Access* 2016;4:9239–50.
- Zhang J, Gao J, Wu Z, Yan W, Wu Q, Li Q, & Chen Z. (2019). Performance Analysis of the Libra Blockchain: an Experimental Study. arXiv preprint arXiv:1912.05241.

**Mohammad Dabbagh** received the Ph.D. degree in Computer Science, specialization in Software Engineering, from University of Malaya (UM), Malaysia in 2015. He is currently a Senior Lecturer and Programme Leader with the Department of Computing and Information Systems at Sunway University, Malaysia. Prior to joining Sunway University, he has acquired enormous working experiences as a lecturer as well as a researcher in the Computer Science discipline. He has published several research papers in prestigious international journals and conference proceedings. His research interests span a broad range of interesting topics including but not limited to Blockchain, Requirements Engineering, Empirical Software Engineering, Big Data Analytics, and Internet of Things. Dr. Dabbagh has been recognized as a Certified Professional in Requirements Engineering (CPRE) by International Requirements Engineering Board (IREB). He is also a member of IEEE Computer Society and Malaysian Software Testing Board.



**Kim-Kwang Raymond Choo** holds the Cloud Technology Endowed Professorship with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio. In 2015, he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen–Nuremberg. He is the recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, British Computer Society's 2019 Wilkes Award Runner-up, 2019 EURASIP JWCN Best Paper Award, Korea Information Processing Society's JIPS Survey Paper Award (Gold) 2019, IEEE Blockchain 2019 Outstanding Paper Award, Inscrypt 2019 Best Student Paper Award, IEEE TrustCom 2018 Best Paper Award, ESORICS 2015 Best Research Paper Award, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008.

**Amin Beheshti** is the Director of AI-enabled Processes (AIP) Research center and the head of the Data Analytics Research Lab, Department of Computing, Macquarie University. He is also the Senior Lecturer in Data Science (Macquarie University) and Adjunct Academic in Computer Science (UNSW Sydney). Amin completed his Ph.D. and Postdoc in Computer Science and Engineering in UNSW Sydney and holds a Master and Bachelor in Computer Science both with First Class Honours. In addition to his contribution to teaching activities, Amin extensively contributed to research projects; where he was the R&D Team Lead and Key Researcher in the 'Case Walls & Data Curation Foundry' and 'Big Data for Intelligence' projects. Amin has been recognized as a high-quality re-

searcher in Big-Data/Data/Process Analytics and has been invited to serve and served as Keynote Speaker, General-Chair, PC-Chair, organisation-Chair and program committee member of top international conferences. He is the leading author of the book entitled "Process Analytics", co-authored with other high-profile researchers in UNSW and IBM research, recently published by Springer. Amin was able to secure over \$2 million research grants for AI-Enabled and Intelligence-Led projects.

**Mohammad Tahir** received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Department of Electrical and Computer Engineering, International Islamic University Malaysia, in 2011 and 2016, respectively. Prior to joining academics, he worked in the R&D Division of industry for seven years on several projects related to the Internet of Things and cognitive radio. His-research interests include 5 G, the Internet of things, game theory for wireless networks, wireless security, Blockchain, and autonomic computing.

**Nader Sohrabi Safa** is an academic member in School of Computing, Electronics and Mathematics, Coventry University, UK. Nader was member of Cyber Security center, WMG, University of Warwick. He successfully finished his postdoctoral in domain of information security at the center for Research in Information and Cyber Security (CRICS) in Nelson Mandela Metropolitan University (NMMU), South Africa. He received his Ph.D. degree in Information Systems from Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. He has teaching experience in several universities. He is a member of IFIP TC 11 Working Group 12, a member of committee in several annual conferences and reviewer in several journals. The outputs of his research have been published in several high impact factor journals.