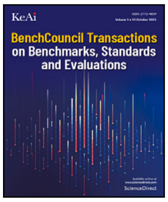
Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/



Research article

Performance comparison of permissioned and permissionless blockchain by varying workload transaction

Madhav Ajwalia [a],[*], Parth Shah [b]

[a] Chandubhai S. Patel Institute of Technology (CSPIT), Faculty of Technology & Engineering (FTE), Charotar University of Science and Technology (CHARUSAT), Changa, Gujarat, India
[b] Department of Studies in Strategic Technologies, School of National Security Studies (SNSS), Central University of Gujarat, Vadodara, Gujarat, India

ARTICLE INFO

Keywords:
Benchmarking
Blockchain
Ethereum
Hyperledger Caliper
Hyperledger Fabric
Performance evaluation
Transaction workload

ABSTRACT

Blockchain technology has fueled exponential growth across various industries, including finance, supply chain management, and healthcare, enabling greater transparency in transaction management and supporting decentralized implementations. This paper presents a comprehensive performance analysis of permissioned and permissionless blockchain platforms, specifically Hyperledger Fabric and Ethereum. The study evaluates these platforms with varying transaction workloads (100 to 1000 transactions) with a consistent network. Our objective is to measure key performance metrics such as send rate, throughput, latency, resource utilization, and transaction success rate using established benchmarking tools and methodologies. The findings offer valuable insights into the comparative strengths, limitations, and optimal use cases of these blockchain platforms across different performance parameters. The results indicate that Hyperledger Fabric achieves, on average, 3.5–4.5 times higher throughput and 10–12 times lower latency than Ethereum, while consuming 2.5–3 times less memory across tested workloads. In contrast, Ethereum demonstrates a higher send rate and lower CPU demand in some operations. Overall, the study suggests that Hyperledger Fabric is better suited for enterprise applications that demand high scalability and performance.

1. Introduction

The arrival of blockchain technology has reformed various sectors, including Finance [1], Governance [2], Internet of Things (IoT) [3], Healthcare [4], Agriculture [5], Supply chain [6], Energy sector [7], Education [8], Public sector [9], Business and Industry [10], and more [11,12]. This technology is used for integrity verification, privacy and security, identity management, data management, and more [13,14]. It introduces decentralized and transparent systems for recording and verifying transactions. It has attracted the significant attention of researchers from both academia and industry.

The growth of blockchain technology has been exponential, with an increasing number of projects, applications, and users entering the ecosystem. Recent market analyses present varying projections for the adoption of blockchain technology. Fortune Business Insights (2023) estimates that the global blockchain market will expand from $17.57 billion in 2023 to $469.49 billion by 2030, representing a compound annual growth rate (CAGR) of 59.9%. In addition, Grand View Research (2023) projects more aggressive growth, forecasting market expansion from $17.46 billion to $1.43 trillion over the same period, indicating a CAGR of 87.7%. Similarly, Statista (2023) anticipates substantial growth at a CAGR of 82.8% from 2021 to 2030, with a market valuation of $1.24 trillion by 2030.

Based on SlashData's "State of the Developer Nation" report, a notable portion of the programming community shows strong engagement with blockchain technology. The research found that 25% of developers are currently learning about or building blockchain applications, while an additional 28% have expressed interest in working with blockchain, dApps, and related development frameworks. This data suggests that more than half of the developer population is either actively involved in or considering blockchain development work. Fig. 1 illustrates the percentage of developers learning about or working on various blockchain platforms, reflecting their comfort with the technology across different application areas.

Over the years, the evolution of blockchain technology has led to four primary categories: public, private, permissioned, and permissionless [16]. These classifications help differentiate how participating nodes interact with the blockchain, achieve consensus among participants, and access or validate data (Fig. 2). Permissioned blockchains, platforms such as Hyperledger Fabric [17], and Corda [18], require participants to be explicitly granted access to the blockchain network. These blockchains are commonly adopted in enterprise settings,

[*] Corresponding author.
E-mail addresses: madhavajwalia.it@charusat.ac.in (M. Ajwalia), parth.shah@cug.ac.in (P. Shah).

https://doi.org/10.1016/j.tbench.2025.100251
Received 2 July 2025; Received in revised form 8 December 2025; Accepted 9 December 2025
Available online 11 December 2025

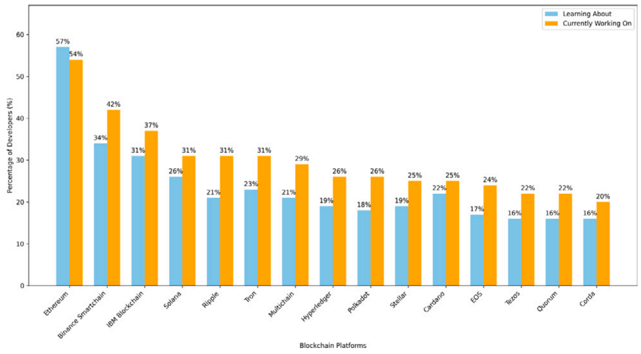


Fig. 1. % of developers working on or learning of the platform [15].

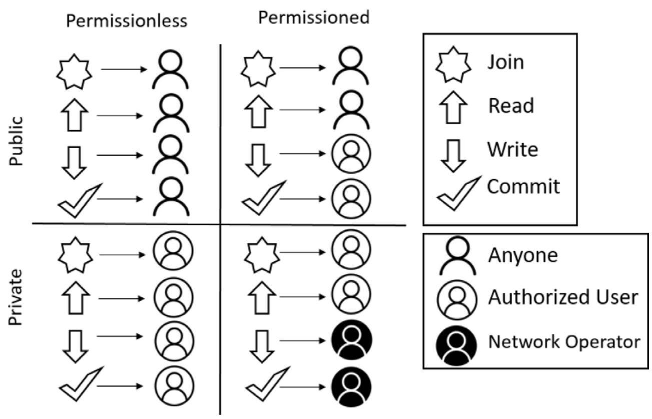


Fig. 2. Join, read, write, and commit access control in public, private, permissionless, and permissioned blockchain categories [16].

offering enhanced privacy, scalability, and control over network governance [19]. In contrast, permissionless blockchains, platforms such as Bitcoin [20] and Ethereum [21], allow anyone to join the blockchain network, participate in transaction validation, and access transaction data.

This difference between permissioned and permissionless blockchains concerns to both public and private networks. Permissionless blockchains permit any member to play the role of validator without constraints, but permissioned blockchains limit this role to known and trusted entities. This dissimilarity substantially affects the performance, security model, and suitability of the network for several applications [22]. Combining public access with authorized involvement in consensus to strike a balance between openness, performance, and control. Some blockchain systems may also show hybrid features [23].

Public blockchains are permissionless in nature, providing unrestricted access, and any participant can join the network, authenticate transactions, and observe the ledger. Such systems are inherently permissionless because no prior permission is needed to join or participate. Well-known examples include Bitcoin and Ethereum, which use consensus protocols such as Proof of Work (PoW) or Proof of Stake (PoS) to ensure security and trust within an entirely open system. While public and permissionless blockchains offer high degrees of transparency and decentralization [24], they typically suffer from limitations in scalability, transaction rates, and energy efficiency as a result of their open and wide participation model [14].

Private blockchains are designed for limited access and are usually controlled by one organization or a consortium. Such networks are naturally permissioned, i.e., only approved participants can read, write, or authenticate transactions. Private blockchains provide more control, quicker transaction processing, better privacy, and are thus increasingly used in enterprise applications like supply chain management, finance, and health care [25,26]. Hyperledger Fabric is a very commonly used example of a private, permissioned blockchain that supports modular architecture, configurable consensus protocols, and fine-grained access control mechanisms [27].

Understanding these various types of blockchain architectures is essential for determining their suitability to specific use cases, when performance, security, and scalability vary per the use case. By analyzing their behavior under varying transactional loads, this study explores the performance of different blockchain architectures. This highlights key architectural design trade-offs and their impact on system performance and efficiency in real-world scenarios.

With the rapid expansion of blockchain use in all sectors, there is an increasing demand for insights into the performance trade-offs for permissioned and permissionless platforms, especially performance under realistic workloads. Although earlier studies have noted architectural and functional differences, a gap remains in quantitative performance comparisons under controlled experimental setups. This research addresses that gap by evaluating and comparing the performance of Hyperledger Fabric and Ethereum under varying transaction workloads while maintaining a constant network size.

Furthermore, the rapid growth of the blockchain market emphasizes the need for comprehensive performance evaluations to measure the suitability of blockchain platforms for specific use cases [28, 29]. Understanding the performance characteristics of these platforms is crucial for stakeholders, as it facilitates informed decision-making and fosters the continued advancement and adoption of blockchain technology across various domains. Bitcoin, the initial application of the blockchain, can be assessed through four performance parameters: mining reward, total forks, transaction throughput, and network latency [30]. Due to the potential for integration with blockchain, people may consider its application beyond cryptocurrency. However, efficiency, stability, scalability, and performance should not be overlooked [31]. [32] delves into various performance modeling techniques used to analyze blockchains. These techniques are categorized into analytical modeling, empirical analysis, simulation, and benchmarking. In this work, a benchmarking method is employed to evaluate the performance of permissionless and permissioned types of blockchain using the Hyperledger Caliper tool [33].

If a blockchain network experiences persistent high load, such as high transaction volume or complex smart contract executions, it may begin to experience performance degradation [34]. This degradation can lead to lower system availability, where nodes become slower to respond or fail to respond due to excessive resource consumption. Resource exhaustion, such as CPU and memory overload, can cause node failures or instability, thereby affecting the overall reliability of the blockchain network. Overloaded systems also face node crashes or synchronization issues, which can compromise data consistency across the blockchain network. These issues also limit scalability, as the blockchain network struggles to maintain performance to handle growing demand. These consequences highlight the importance of performance evaluation, particularly when comparing permissioned and permissionless models under dynamic load conditions.

In preview of our findings, the comparative evaluation reveals clear distinctions between the two platforms. Hyperledger Fabric demonstrated consistently higher throughput and lower latency under controlled workloads, along with greater efficiency in resource utilization. Ethereum, by contrast, sustained higher transaction submission rates but at the cost of increased resource consumption. Fabric's resource efficiency and responsiveness make it suitable for permissioned, enterprise-grade use cases such as timebanking applications, whereas Ethereum reflects the trade-offs inherent in public, permissionless environments. These findings help in selecting platforms for real-world deployments.

This paper contributes a structured comparative study of permissioned (Hyperledger Fabric) and permissionless (Ethereum) blockchains under systematically varied transaction workloads. It benchmarks multiple transaction types (open, query, transfer) and examines a broader set of performance indicators beyond conventional throughput and latency, including send rate, transaction success rate, and resource utilization. The experimental design employs standard Dockerized deployments to ensure reproducibility and to isolate the impact of workload intensity on system behavior. The findings provide empirical evidence of the fundamental trade-offs between permissioned and permissionless models, demonstrating Fabric's strengths in throughput, latency, and efficiency, while Ethereum demonstrates higher submission rates, offering practical insights for blockchain platform selection. Through this empirical investigation, this study examines where the performance of permissioned and permissionless blockchain networks excels and limits.

In the subsequent sections of this paper, we examine the existing work, discuss the methodology employed for performance evaluation, present and analyze our findings, and outline paths for future research in the dynamic landscape of blockchain technology.

2. Literature survey

Many recent research works have identified the performance issue of blockchain-based systems as a promising research topic. They have investigated the performance using a practical and theoretical method that shows potential developments in the field. This section discusses the performance research of various application areas related to supply chain [35,36], network service federation [37], cryptocurrency [38], finance [39], trading [40], e-voting [41], IoT [42], and mining [43].

"LogisticChain" [35] introduces a proof of concept model that leverages the Hyperledger Fabric blockchain infrastructure to model shipping logistics workflows. System performance assessment involves manipulating various parameters including client volume, simultaneous transaction processing, and per-second transaction frequency, while monitoring metrics such as latency, send rate, and throughput. Read operations exhibit the lowest latency, while Update operations show the highest latency due to the complex computations and validations involved. The LogisticChain implementation reveals lower latency when utilizing a constant-rate controller compared to a linear-rate controller.

The work [36] presents a prototype of BloodChain based on the private blockchain Hyperledger Fabric. It evaluates the performance and effectiveness of application claims enhanced security, transparency, and traceability. Execution observes requests per second and latency for create, request, and update functions under different test loads. The system can handle a large number of requests and maintain a relatively stable performance. However, there are some fluctuations in latency, mainly when the system is under a high load, from 1000 to 10,000 transactions.

The research work [37] compares Proof of Work (PoW), Proof of Authority (PoA), Practical Byzantine Fault Tolerant (PBFT), and Proof of Stake (PoS) by analyzing their performance on platforms like Ethereum, Tendermint, and Cosmos in the context of Network Service Federation (NSF). It emphasizes the negotiation and implementation aspects of multi-cloud federation, offering insights into the appropriateness of each consensus mechanism for NSF applications. The research evaluates latency, send rate, and throughput to assess the effectiveness of each consensus method. It investigates the performance of Blockchain hosts considering CPU, memory, disk, and network usage. PoA and PBFT mechanisms show lower latency and higher throughput, making them more suitable for NSF applications.

The paper [38] introduces a reputation-based blockchain protocol in Bitcoin networks to assign a controler node for each cluster, which propagates transactions to other clusters. The protocol, called Master Node Based Clustering (MNBC), aims to reduce the propagation delay of transactions by grouping nodes based on their physical internet proximity. The study evaluates the proposed methods through simulation experiments and shows that they can optimize the transaction propagation delay compared to the Bitcoin protocol. Nevertheless, MNBC's efficiency declined when the proportion of compromised nodes rose from 5% to 30%.

[39] Evaluate the performance of the ConsenSys Quorum blockchain platform, permissioned blockchain designed for financial applications. This study aims to analyze the throughput, latency, and scalability. It summarizes that Istanbul Byzantine fault tolerant (IBFT) represents the best choice for Quorum in financial applications. The paper also discusses the challenges faced by permissioned blockchains in achieving high performance and the potential solutions to overcome these challenges.

The performance of the livestock application based on the Hyperledger Iroha blockchain framework is evaluated [40]. The study assesses the framework based on three key parameters: total requests per second (RPS), response times in milliseconds over time, and the number of users in the network over time. The findings suggest that Hyperledger Iroha can effectively support at least 200 participants without errors in the network. It can handle up to 40.6 requests per second, and the response times are rapid, typically less than a second.

[41] develops a private blockchain infrastructure designed for democratic voting systems, utilizing blockchain network for data storage and processing operations. Smart contracts are implemented to handle transaction execution within the system. The authors conduct a comparative performance evaluation examining two widely-used Ethereum client implementations, Geth and Parity, analyzing their capabilities across throughput, latency, and scalability parameters. On average, transactions are 91% faster in the Parity client compared to the Geth client.

[42] compares the performance of their solution with existing access management solutions in IoT. The paper conducts realistic experiments to evaluate the performance of the proposed system in terms of latency, throughput, and scalability. It is observed that, in the case of only single management hub, the proposed solution achieves better performance than the optimized centralized IoT systems. The proposed implementation offers significant scalable advantages over traditional scenarios in the case of multiple management hubs.

"MobiChain" [43] experiments were performed on a mobile node. It evaluates the performance of the proposed model in terms of computation time, energy consumption, and memory utilization for chain verification. It observes that the chain verification process is executed faster and consumes less energy when transactions are grouped in a block. Moreover, the execution time reduces insignificantly as the number of threads increases.

The reviewed literature (Table 1) shows that performance evaluation across blockchain applications is essential. Performance evaluation trend shifts from theoretical analysis to practical implementation studies. That emphasis on domain-specific performance metrics and real-world usability factors. Scalability remains the primary challenge across all application domain. Energy efficiency and resource constraints are concerns, specifically for mobile and IoT applications. However, most studies focus on individual platforms rather than comprehensive cross-architecture comparisons. This study addresses these limitations by systematically comparing the Hyperledger Fabric and Ethereum platforms under varying workload scenarios, offering valuable insights for informed deployment decisions in both enterprise and public contexts.

3. Methodology

This study includes configuring Hyperledger Fabric as a permissioned blockchain network and Ethereum as a permissionless blockchain network. It implements chaincode and smart contracts and runs several predefined test scenarios with the benchmarking tool Hyperledger Caliper. That measuring various performance metrics and analyzing the results. To evaluate performance, the tool runs a comprehensive set of test scenarios that include a variety of workloads, transactions, and network sizes. For the study, workload transaction per second (TPS) is systematically varied from 100 to 1000 while keeping the network size constant.

Table 1
Comparison of performance evaluation studies in blockchain applications.

Paper	Study Focus	Network/Platform	Consensus Mechanism	Tools Used	Key Performance Metrics	Key Observations	Limitations
[35]	Analyze performance of a blockchain-based system for maritime logistics	Hyperledger Fabric	Raft	Hyperledger Caliper	Throughput, latency, scalability, transaction processing speed	Lower latency with fixed-rate controllers than linear-rate controllers	Simulation-only results; lacks of real-world deployment data; scalability and interoperability issues
[36]	Scalability performance of permissioned network in Healthcare	Hyperledger Fabric	Not Mentioned	Hyperledger Caliper	Data integrity, traceability, system availability, user adoption rates	Stable performance under high load; fluctuations in latency occur, particularly under high load from 1000 to 10,000 transactions	Conceptual; lacks full metrics and implementation; privacy compliance concerns
[37]	Evaluation of blockchain consensus mechanisms for federated network services	Ethereum, Tendermint, Cosmos	Proof of Work, Proof of Authority, PBFT, Proof of Stake	Not Mentioned	Consensus latency, energy use, fault tolerance, network overhead	PoW and PBFT show lower latency and higher throughput making them more suitable for NSF applications	Limited scalability analysis; limited consensus algorithms tested
[38]	Investigate the performance and security of a reputation-based blockchain in cryptocurrency network	Bitcoin network	Rapid-Chain/MNBC	Bitcoin simulator, Metis graph partition toolkit	Security metrics, reputation scoring, network resilience, transaction speed	Proposed Master Node based Clustering (MNBC) protocol offers improvement in information propagation delay compared to Bitcoin protocol. However, MNBC performance decreased as malicious nodes increased from 5% to 30%	Simulation focus; Bitcoin-specific; centralization and vulnerability concerns
[39]	Performance evaluation of Permission blockchain in Financial Services	ConsenSys Quorum	Raft, Clique Proof of Authority, Istanbul BFT	Hyperledger Caliper	Transaction throughput, privacy, compliance, cost-efficiency	Istanbul Byzantine fault-tolerant (IBFT) represents the best choice for Quorum in financial applications	Single-platform focus; lacks cross-platform comparison
[40]	Evaluate performance of permissioned network in livestock management and trading platform	Hyperledger Iroha	Byzantine Fault Tolerant	Not Mentioned	Transaction processing, data provenance, system reliability, adoption	Supports 200+ participants, 40.6 tx/sec throughput	Small-scale testing only; lacks stress/fault tolerance
[41]	Benchmark public blockchain in E-Voting scenario	Ethereum (Geth and Parity)	Proof of Work	Not Mentioned	Vote speed, system security, voter privacy, electoral integrity	Parity is 91% faster than Geth for transactions	Ethereum-only; limited security analysis; privacy vs. transparency trade-offs
[42]	Analyze performance of Blockchain-based access control for IoT devices	Ethereum	Not Clear	CoapBench	Device auth speed, scalability, energy efficiency	Scalable advantage in multi-Hub scenario; single-Hub underperforms centralized systems	Private blockchain only; no constrained device tests; energy/resource limits
[43]	Performance analysis of Blockchain in Mobile device for mining	Bitcoin network	Proof of Work	VideoOptimizer program	Battery consumption, processing efficiency, network usage, user experience	Grouped transactions improve speed and reduce energy use	Energy consumption and bandwidth not fully evaluated; mobile device variability

3.1. Experimental design description

The experimental design follows a structured workflow where business logic is developed independently for each platform before performance evaluation. The process consists of four main phases: development of business logic, configuration for benchmarking, benchmark execution and measurement, and result analysis.

Development of Business Logic: Application-level business logic was developed independently for each platform. For Hyperledger Fabric, chaincode was implemented in the Node.js, while for Ethereum, smart contracts were written in Solidity. These implementations covered three core functions (Open, Query, Transfer) and were validated through unit testing prior to benchmarking. The code was deployed on private test networks for each platform, each hosted in an isolated Dockerized environment to ensure fairness and control across experiments.

Configuration for Benchmarking: Hyperledger Caliper served as the benchmarking framework. It was configured using an adapter and network files referencing the deployed business logic, specifying workload parameters, transaction types, and network details. Caliper does not develop or execute code internally; instead, it interfaces with predeployed smart contracts or chaincode via standardized blockchain APIs. This ensures that benchmarking reflects actual system behavior rather than simulated execution.

Benchmark Execution and Measurement: Once configured, Caliper initiated benchmarking by generating workloads against the deployed networks. Each experimental run followed a systematic sequence: initially, blockchain networks are deployed using validated business logic; Caliper then connects to each network via standardized APIs. Once connected, workload modules initiate transaction patterns defined by experimental parameters, prompting the execution of business logic on the networks. Throughout this process, Caliper captures transaction-level metrics through analysis of API interactions and timestamps. Parallel system-level monitoring captures resource utilization data, providing a comprehensive view of system performance. Each scenario was executed three times, and Caliper aggregates and stores all results for statistical and comparative analysis. Scalability was assessed by varying transaction rates across the workload range, while keeping the network size fixed to isolate workload effects.

Result Analysis and Reproducibility: Caliper outputs structured JSON and performance reports, which were analyzed for performance

Table 2
Test environment for performance evaluation of blockchain networks.

Component	Platform/Network	Version
Benchmarking Platform	Hyperledger Caliper	v0.5.0
Blockchain Network	Ethereum Hyperledger Fabric	1.0 2.5.0
Business Logic	Smart Contract with Solidity Chaincode with Node.js	0.8.23 16.13.1, 12.22.9

trends and platform comparisons. Control variables, including hardware specifications, software versions, Docker container settings, and environment parameters, remained constant across all experiments to ensure comparability. Detailed documentation of all configuration parameters, software versions, and experimental steps enables reproducibility.

3.2. Experimental setup

Table 2 shows the test environment prepared for this performance study. Hyperledger Caliper v0.4.2 was used as a benchmarking tool. Chaincode, smart contract development, and execution were carried out using the programming languages Go, Solidity, and JavaScript. Visual Studio Code (VSCode) worked as a development environment for code creation and debugging. For the CPU configuration, we used an Intel Core i5-1135G7 CPU operating at 2.40 GHz for the Ubuntu v22.04.3 LTS system.

We used Hyperledger Fabric v2.5.0 and Ethereum as permissioned and permissionless networks, respectively, and deployed them with Docker engine versions 20.10.22 and 24.0.7. We used Docker Compose versions 2.15.1 and 1.29.2 to manage containerized blockchain environments.

3.3. Benchmarking tool

Hyperledger Caliper is a benchmarking tool designed explicitly to evaluate the performance of blockchain networks. It provides a framework for executing different scenarios, measuring various performance metrics, and analyzing the results. Caliper's general framework [44] is used to run benchmarks against various blockchain frameworks. That generates a workload for a particular system under test (SUT) and keeps track of how it responds all the time. Finally, a report will be produced based on the observed SUT responses. This simplistic view is depicted in Fig. 3.

Caliper needs many inputs to execute a benchmark, regardless of the selected SUT. The following subsets provide a quick summary of these inputs.

- Benchmark configuration file: This file specifies the benchmarking process's parameters, including the blockchain platform, number of participating nodes, consensus algorithms, and other test-specific parameters.
- Network configuration file: This file is dedicated to the blockchain network setup and contains information about network nodes, their addresses, and other relevant network-related details.
- Workload modules: It is the brain of the benchmark. It customizes the benchmarking workload, allowing users to tailor test scenarios to their specific use cases and evaluate the blockchain platform's performance under realistic conditions.
- Benchmark artifacts: Artifact or result generated by the benchmarking process, encapsulating key performance metrics, transaction details, and other relevant data collected during the experiment.

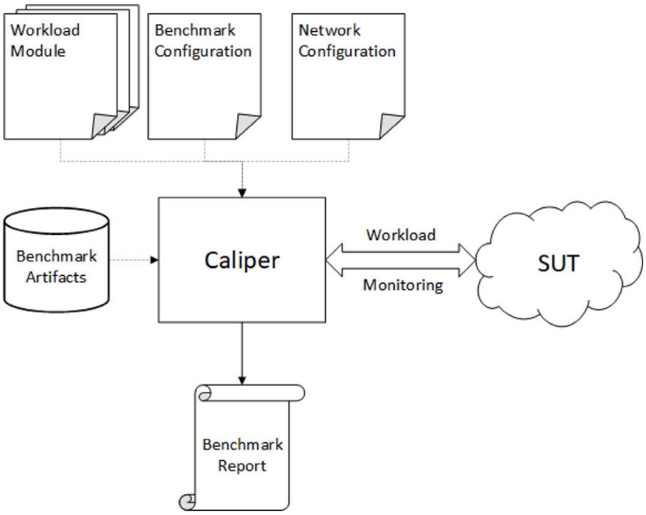


Fig. 3. Architecture of Hyperledger Caliper [44].

- Caliper core: This central component is responsible for the workload module and orchestrating the benchmarking process. It includes the core logic for test execution, result aggregation, and communication with blockchain networks.
- Benchmark Process:

- Initialize the Caliper core, load the specified workload module, and connect to the target blockchain network using the provided configuration.
- Caliper generates and submits transactions to the blockchain network based on the workload module.
- The network monitor records relevant metrics, and the result writers store the collected data for later analysis.
- The Caliper produces detailed reports and analysis, allowing users to evaluate the performance of the tested blockchain platform.

Performance metrics such as send rate, throughput, latency, resource utilization, and transaction success rates can be measured and analyzed. These metrics assess the blockchain network's scalability, efficiency, and reliability. Many research articles present theoretical insights, but may lack empirical evidence or real-world case studies to support their findings.

3.4. Parameter selection rationale

In the context of performance evaluation for blockchain networks, we used several key metrics to assess the network's efficiency and effectiveness. The performance metrics in this study were carefully selected to align with both prior blockchain performance evaluation and benchmarking standards and the specific objectives of this research. Our chosen metrics align with established performance modeling techniques categorized into analytical modeling, empirical analysis, simulation, and benchmarking approaches [32]. Previous systematic surveys [28, 29,45–47] consistently identify throughput, latency, and resource utilization as the core indicators of blockchain performance, as they directly reflect scalability, efficiency, and user experience. [28] identifies throughput, latency, and resource efficiency as the most critical indicators of blockchain usability, while [29] emphasizes the need to capture both user-perceived and system-level performance characteristics. In parallel, [45] classifies throughput, latency, and resource consumption as the fundamental evaluation criteria applied across multiple approaches, and [46] highlights the necessity of analyzing consensus through transaction latency and success rate. [47] reinforces this view, noting that among the various metrics, send rate, throughput,

Table 3
Symbolic definitions for blockchain performance metrics.

Symbol	Meaning
Tx	Transactions
Tx_p	Successfully processed transactions
T_t	Transaction processing duration
T_{init}	Transaction initiation time
T_{conf}	Transaction confirmation time
R_u	Actual resource usage
R_{total}	Total available resources

latency, CPU utilization, memory usage, energy efficiency, and security are consistently highlighted as the most important and widely adopted empirical performance evaluation parameters for blockchain systems.

Other vital parameters, energy efficiency and security metrics, were excluded from this baseline study due to scope constraints. Energy efficiency evaluation requires specialized power measurement infrastructure and extended observation periods that would significantly complicate the controlled experimental design. Security assessment requires separate methodologies, including penetration testing and cryptographic analysis, representing a distinct research domain beyond the performance benchmarking scope. This study establishes foundational performance baselines that inform subsequent specialized energy and security evaluations.

3.5. Performance parameters

These metrics collectively address usability, adoption, and sustainability of blockchain technologies and are therefore central to any comprehensive performance evaluation. These metrics provide a solid and comprehensive foundation for assessing the performance trade-offs between permissioned and permissionless blockchain platforms across different workload scenarios. Their combined result also proves the network's scalability, efficiency, and practical deployment considerations. Different blockchain networks may prioritize these metrics over others based on their use cases and requirements. Table 3 shows the meaning of the symbols utilized in the definitions of the blockchain performance metrics.

Send Rate: That represents the number of transactions sent in the network by the participant node per unit of time. It measures the rate at which transactions are submitted to the blockchain network.

$$\text{Send Rate} = \frac{\text{Total number of transactions sent}}{\text{Time duration}} = \frac{\sum Tx}{T_t} \tag{1}$$

Throughput: It measures the processing capacity of the blockchain network, i.e., the number of transactions successfully processed per unit of time. It reflects the network's ability to handle a certain volume of transactions within a given timeframe.

$$\begin{aligned}\text{Throughput} &= \frac{\text{Total number of transactions successfully processed}}{\text{Total time period}}\\ &= \frac{\sum Tx_p}{\sum T_t}\end{aligned} \tag{2}$$

Latency: It refers to the time delay between initiating a transaction and its final confirmation or inclusion in the blockchain network. It measures the time taken for transactions to be propagated, validated, and included in a blockchain.

$$\begin{aligned}\text{Latency} &= \text{Time taken for transaction confirmation}\\ &\quad - \text{Time of transaction initiation}\\ &= T_{\text{conf}} - T_{\text{init}}\end{aligned} \tag{3}$$

Success Rate: It indicates the proportion of transactions or operations that are executed successfully without errors or failures.

$$\begin{aligned}\text{Success Rate} &= \frac{\text{Total number of transactions successfully processed}}{\text{Total number of transactions}}\\ &= \frac{\sum Tx_p}{\sum Tx}\end{aligned} \tag{4}$$

Resource Utilization: Resource utilization assesses the efficiency of the blockchain network's resources, such as computing power and memory usage, in processing and validating transactions. It measures how effectively the network utilizes its resources to maintain its operation.

$$\text{Resource Utilization} = \frac{\text{Actual resource usage}}{\text{Total available resources}} = \frac{R_u}{R_{\text{total}}} \tag{5}$$

3.6. Benchmarking operations

In the context of performance evaluation, the 'open','query', and 'transfer' functions are essential tasks carried out during benchmarking scenarios to evaluate blockchain network performance [48]. Every function has a specific role: 'open' initializes and sets up the blockchain network, 'query' collects data from the network, and 'transfer' carries out transactions to move assets or information. By incorporating these functions into benchmarking scenarios, Caliper allows for the measurement and analysis of various performance metrics. This supports a thorough evaluation of the blockchain network's efficiency, scalability, and reliability across varying workloads.

- Open function: The open function sets the blockchain network up for benchmarking by initializing it. This could involve tasks such as setting up or creating accounts or participants, deploying smart contracts, configuring network parameters, loading initial data onto the blockchain, and so forth. The function initializes the starting state of the blockchain network before running any benchmark transactions.
- Query Function: The query function fetches data from the blockchain network. This could involve checking transaction records, smart contract status, account balances, or other relevant information stored on the blockchain. Evaluating the effectiveness of data retrieval systems and the responsiveness of the blockchain network to read queries depends on query operations.
- Transfer Function: Transfer function is essential for the transfer of value or assets in the blockchain network. It consists of submitting transactions into the network to transfer tokens, assets, or data between smart contracts or accounts. Transfer operations are essential for assessing transaction-processing capacity.

With these functions, Caliper enables users to simulate real-world scenarios by benchmarking and evaluating their performance under various workloads and conditions across various blockchain networks.

3.7. Platform selection rationale

Various surveys and comparative analyses highlight Ethereum and Hyperledger Fabric as leading examples of permissionless and permissioned blockchains, respectively, due to their consensus algorithms and performance capabilities. These two systems are the most widely adopted and documented in both academic research and industry deployments, making them ideal for comparative studies.

Examples of public permissionless blockchains include Bitcoin, Ethereum, Litecoin, Cardano, and Polkadot, all of which enable open participation. By contrast, permissioned blockchains such as Hyperledger Fabric, R3 Corda, Quorum, Ripple, and IBM Blockchain emphasize controlled access, enhanced privacy, and enterprise-grade functionality. Many permissionless platforms, including Binance Smart Chain

(BSC), Polygon, Avalanche, Fantom, and Tron, extend Ethereum-like designs, leveraging the Ethereum Virtual Machine (EVM) or Ethereum-inspired smart contracts while maintaining public participation. Similarly, permissioned platforms like Quorum, Corda, Hyperledger Sawtooth, and IBM Blockchain adopt Hyperledger Fabric's modular, privacy-oriented, and enterprise-driven design principles. In addition, several leading platforms, including Ethereum 2.0, Cardano, Polkadot, Tezos, and Algorand, are transitioning toward or have already adopted PoS. Research [32] focused on performance modeling of a public permissionless blockchain. It suggests that, PoW and PoS are widely used and the most popular consensus algorithms in this category. Although Algorand has emerged as a promising alternative, Ethereum represents this class majorly. A detailed study [49] evaluates the performance of consensus algorithms and blockchain platforms using twelve assessment indexes. It also discusses eight common blockchain platforms, including Bitcoin, Ethereum, Hyperledger Fabric, Quorum, Ripple, Corda, EOS, and IOTA, analyzed by these indexes to highlight their distinct strengths. Comparative analyses of permissioned blockchain, such as Cosmos, Hyperledger Fabric, Quorum, and XRPL, demonstrate Fabric's superiority in consensus, smart contracts, custom tokens, privacy, latency, and throughput, despite limitations in interoperability [50]. Researchers [51] provide a comparative study of permissioned frameworks with regard to the community activities, performance, scalability, privacy and adoption criteria. Study shows fabric is promising. In the area of blockchain cloud integration, while providers such as AWS and Google Cloud offer Blockchain-as-a-Service (BaaS), Fabric remains the preferred choice for private deployments [52]. Survey [53] investigated 12 most popular blockchain platforms and elaborated six platforms that are widely applied in finance. This study observes that the proportion of companies using Ethereum and Hyperledger Fabric for application development is 24% and 38%, respectively, among the top 100 companies. It demonstrates their market dominance in financial use cases. This highlights why Ethereum and Hyperledger Fabric are regarded as foundational benchmarks for permissionless and permissioned blockchains, respectively, since studying them sufficiently captures the core architectural and performance traits of each category. Thus, focusing comparative performance studies on Ethereum and Hyperledger Fabric is sufficient to capture the essential dynamics of these two prominent blockchain domains. Including additional platforms for a performance comparison study introduces redundant comparisons, as many permissionless or permissioned blockchains often derive from or closely align with these core models. [49] highlights that the optimal blockchain platform depends on the particular application and desired performance criteria such as throughput, scalability, energy consumption, cost, and security. So the final selection of a framework for a specific use case is always a trade-off.

3.8. Methodological contribution

While Hyperledger Caliper serves as the underlying benchmarking tool, this study goes beyond its default usage by proposing a structured evaluation framework that introduces methodological innovations to enhance both analytical depth and practical relevance. Rather than relying solely on the tool's default configurations, the study introduces distinct experimental controls and performance metrics, positioning the work beyond generic benchmarking practices (See Fig. 4).

This study intentionally employs Hyperledger Caliper to ensure consistency and comparability with prior research. Unlike conventional applications of Caliper, we extend its application through a structured framework that incorporates systematic workload variation, multi-function benchmarking, extended performance metrics, and standardized deployments. While the development of entirely new benchmarking tools lies beyond the scope of this work, the framework established here offers a reliable baseline and can serve as a foundation for future research exploring more adaptive or context-specific evaluation methodologies.

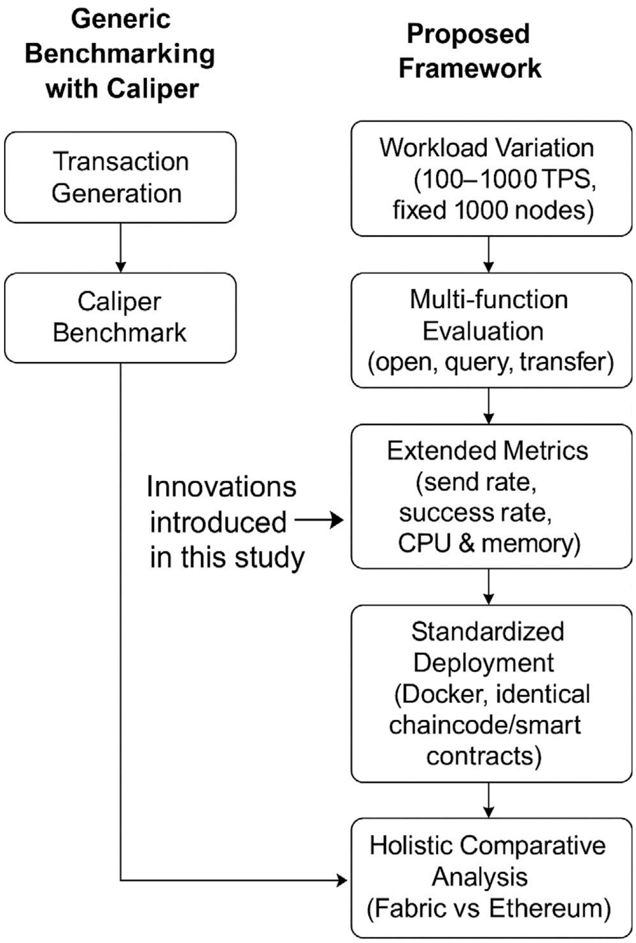

Generic Benchmarking with Caliper
Proposed Framework
Transaction Generation
Caliper Benchmark
Workload Variation (100–1000 TPS, fixed 1000 nodes)
Multi-function Evaluation (open, query, transfer)
Innovations introduced in this study
Extended Metrics (send rate, success rate, CPU & memory)
Standardized Deployment (Docker, identical chaincode/smart contracts)
Holistic Comparative Analysis (Fabric vs Ethereum)

Fig. 4. Generic caliper vs proposed framework.

First, transaction workloads were systematically varied between 100 and 1000 TPS while maintaining a fixed network size. This design isolates the impact of workload intensity without conflating it with network scaling, a limitation observed in prior benchmarking studies. Second, the research develops a 3-dimensional analysis approach examining platform performance across distinct functional operations (Open, Query, Transfer) rather than aggregate metrics. This function-specific methodology reveals performance characteristics that aggregate evaluations obscure, providing granular insights for application-specific deployment decisions. Finally, both Ethereum and Hyperledger Fabric are deployed in standardized Dockerized environments with equivalent business logic implemented in smart contracts and chaincode. This ensures fairness, reproducibility, and guarantees that observed differences arise from platform characteristics rather than deployment variations.

Collectively, these contributions move the study beyond a routine use of existing tools and establish a replicable evaluation framework for blockchain performance assessment. The approach not only strengthens the validity of comparative analysis between permissioned and permissionless systems but also provides a methodological template for future benchmarking studies in the specific use case.

4. Result discussion

The result illustrates how Ethereum and Hyperledger Fabric (HLF) performed when we tested different functions like 'Open', 'Query', and 'Transfer'. For this result, the transaction load gradually increased from 100 to 1000 transactions per second (TPS) while keeping the network size constant to see how each platform would handle the workload.

Table 4
Performance results for ‘open’ function.

TPS	Send Rate (TPS)		Throughput (TPS)		Max Latency (s)		CPU Utilization (%)		Memory Usage (MB)	
	Ethereum	HLF	Ethereum	HLF	Ethereum	HLF	Ethereum	HLF	Ethereum	HLF
100	100.2	121.4	31.1	120.5	22.77	1.96	1.96125	7.56750	1239.04	505.08
200	200.4	145.9	31.1	140.0	27.50	1.96	2.16250	7.99500	1505.28	520.84
300	300.9	150.5	31.9	149.4	28.28	1.92	0.17625	8.28125	1546.24	490.57
400	400.8	175.2	31.2	155.7	29.76	1.99	0.23375	6.47375	1525.76	516.42
500	501.0	154.6	31.2	148.2	30.18	1.95	0.04875	7.60125	1525.76	517.06
600	600.2	158.8	31.9	153.2	32.89	1.97	0.45607	8.04947	1520.73	505.85
700	700.2	167.9	31.4	155.3	34.45	2.05	0.57142	9.67462	1555.92	515.79
800	800.3	163.8	31.5	157.4	36.10	2.22	0.79558	9.69863	1555.22	516.57
900	900.4	157.7	31.1	159.1	38.07	2.31	0.39635	9.64107	1550.10	511.98
1000	1000.2	162.7	31.3	160.4	39.86	2.35	0.3383	10.15398	1545.31	518.39

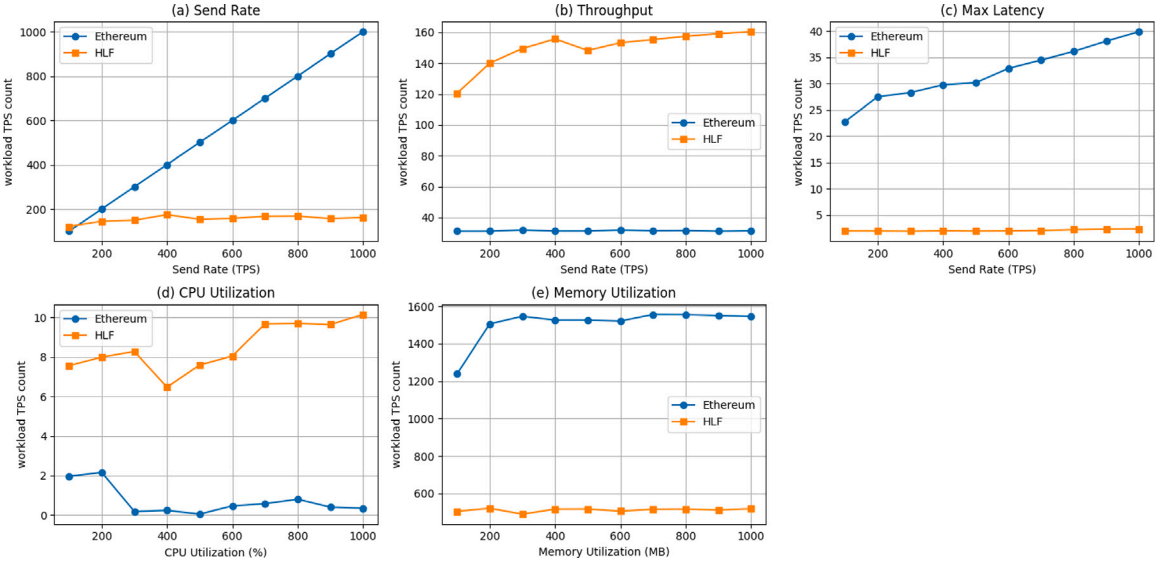


Fig. 5. Performance results for ‘open’ function with the varying workload from 100 to 1000.

Metrics include send rate, throughput, maximum latency, CPU utilization, and memory usage. This work provides valuable insights into platform activity under various transaction workloads, offering a subtle understanding of efficiency and scalability.

As shown in Fig. 5 and Table 4, the result for the ‘open’ function, reveal the following trends: The open function highlights key differences in how Ethereum and Hyperledger Fabric process write-intensive workloads. Across the full workload spectrum (100–1000 TPS), Fabric consistently maintains throughput close to the incoming send rate, achieving up to 160 TPS. In contrast, Ethereum’s throughput is around 31 TPS irrespective of higher send rates. These numbers reflect Ethereum’s consensus constraints that restrict how many ‘open’ transactions can be committed per unit time. Ethereum’s maximum latency increases linearly with workload, exceeding 39 s at 1000 TPS. This outcome is typical in a PoW-based system, where transaction congestion occurs in the mempool when the volume of submitted transactions exceeds the network’s processing capacity. Fabric, in contrast, maintains latency within 2–3 s even at peak load, as its modular consensus (ordering service with endorsement policies) is optimized for rapid block finality in permissioned settings. Resource utilization aligns with these patterns. Ethereum exhibits a steadily rising CPU and memory usage as workload increases, reflecting the computational burden of block validation and state updates under PoW. Fabric, while consuming CPU and memory at lower levels, shows slightly increased CPU usage at higher TPS as the ordering service manages more frequent block commits. Overall, the open function analysis highlights Fabric’s superior suitability for enterprise contexts requiring reliable throughput and bounded latency, while Eth-ereum demonstrates the scalability limits imposed by its permissionless consensus.

Fig. 6 and Table 5 illustrate several observable trends regarding the ‘query’ function, summarized as follows: The query operation demonstrates a significantly different pattern. Here, both Ethereum and Fabric achieve throughput nearly equal to the send rate across all workloads up to 1000 TPS. Queries are read-only operations that do not alter the blockchain state, meaning they bypass the heavy consensus bottlenecks associated with write transactions. As a result, Ethereum handles queries at near-linear scalability, achieving 1000 TPS throughput when pushed to that workload. Fabric mirrors this performance, consistently matching the query submission rate. Latency for queries remains negligible on both platforms, with Ethereum maintaining under 0.03 s and Fabric slightly higher but still well under 0.1 s across all load levels. This clear difference with the open function highlights the architectural difference between read and write operations in blockchain. Since queries mainly involve state lookups, consensus delays and block validation overheads are avoided. CPU and memory utilization reflect the lightweight nature of queries. Ethereum’s CPU consumption is somewhat higher than Fabric’s, which aligns with its execution environment (EVM-based contract logic and PoW validation, even for read calls). Fabric remains comparatively more efficient, with CPU usage below 1% and a stable memory usage across workloads. The query analysis confirms that both platforms can scale read-heavy workloads effectively, but Fabric achieves this with lower resource overhead.

As shown in Fig. 7 and Table 6, the behavior of the ‘transfer’ function exhibits the following trends: The transfer function presents the most demanding workload by combining state updates with consensus enforcement. Fabric maintains throughput above 130 TPS across all load levels, reaching at 160 TPS, whereas Ethereum again saturates at 38 TPS. This limit in Ethereum reflects the same architectural

Table 5
Performance results for ‘query’ function.

TPS	Send Rate (TPS)		Throughput (TPS)		Max Latency (s)		CPU Utilization (%)		Memory Usage (MB)	
	Ethereum	HLF	Ethereum	HLF	Ethereum	HLF	Ethereum	HLF	Ethereum	HLF
100	100.1	699.3	100.1	697.8	0.01	0.09	0.55500	0.31125	1249.28	486.28
200	200.3	708.2	200.2	706.7	0.03	0.08	0.47875	0.42125	1361.92	491.44
300	300.4	686.8	300.3	685.4	0.02	0.15	0.19125	0.42750	1372.16	481.57
400	401.0	714.8	400.8	713.8	0.01	0.05	0.25125	0.22000	1341.44	503.12
500	501.0	705.7	501.0	703.2	0.02	0.05	0.15000	0.40875	1341.44	506.56
600	600.2	705.6	599.3	704.6	0.018	0.06	0.22	0.60485	1347.15	500.00
700	700.2	706.5	700.0	707.2	0.019	0.08	0.234	0.79843	1346.9	494.6
800	800.3	711.2	799.5	713.5	0.014	0.07	0.3421	0.7649	1347.1	497.31
900	900.4	710.3	898.9	713.2	0.014	0.08	0.352	0.81362	1348.7	508.32
1000	1000.2	711.1	999.6	708.6	0.024	0.06	0.3403	0.83039	1349.0	508.0

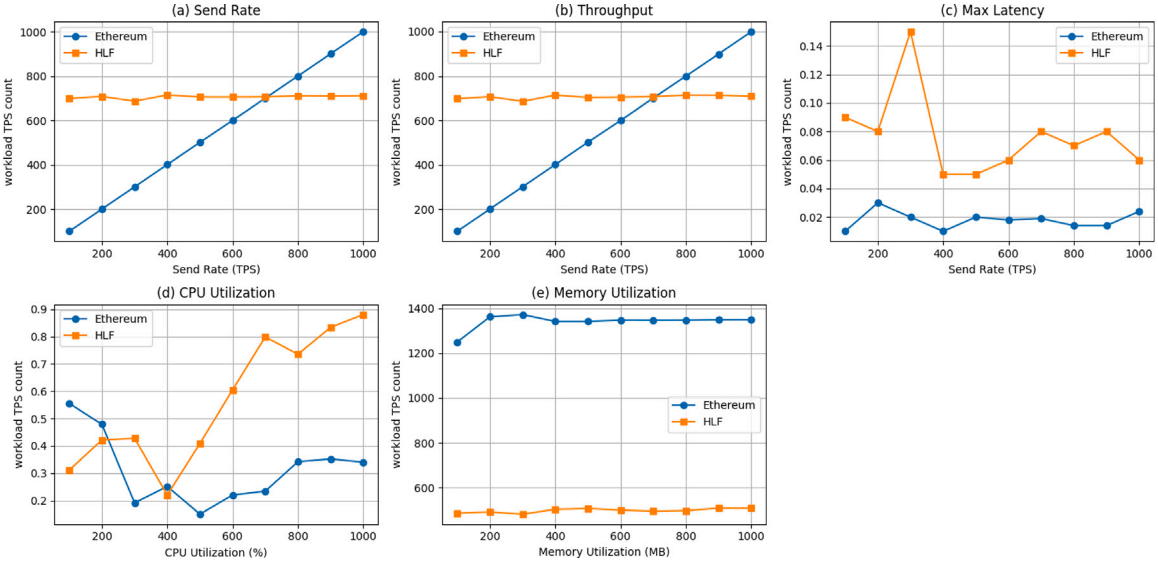


Fig. 6. Performance results for ‘query’ function with the varying workload from 100 to 1000.

bottlenecks seen in the open function: block gas limits and the sequential nature of PoW consensus. Fabric’s higher throughput results from its efficient ordering and endorsement process. This allows for parallelism and trust assumptions in a controlled, permissioned environment. Latency patterns further highlight this gap. Ethereum’s maximum latency increases sharply with increasing load, from 17 s at 100 TPS to nearly 32 s at 1000 TPS. This Growth reflects the backlog of pending transactions in the mempool during sustained load. Fabric maintains latency under 3.5 s even at peak workloads, reinforcing its strength in providing predictable responsiveness. Resource utilization trends are consistent with the throughput findings. Ethereum’s CPU usage steadily increases beyond 40% as the system struggles with backlogged transactions and block production overhead. Memory consumption also increases, exceeding 1600 MB at the highest workload. Fabric, in contrast, consumes minimal additional resources, with CPU usage increasing modestly but still below 7% and memory remaining near 540 MB. These results show Fabric’s advantage for financial or transactional systems that need reliable settlement speed and efficiency.

5. Conclusion and future work

This study provides a detailed performance comparison of permissioned and permissionless blockchains under workloads ranging from 100 to 1000 TPS. Ethereum consistently achieves a higher send rate, reflecting its suitability for scenarios where transaction initiation speed and public participation are essential. Hyperledger Fabric offers, on average, 3.5–4.5 times higher throughput and 10–12 times lower latency across tested functions, demonstrating its efficiency for enterprise-grade workloads. Resource utilization further highlights the trade-offs: Fabric consumes 2.5–3 times less memory than Ethereum but requires higher CPU usage for some operations, such as open transactions. Ethereum, on the other hand, often shows lower CPU demand but at the cost of significantly higher memory consumption and reduced confirmed throughput. Both systems scale with increasing workloads, showing robustness across the tested range. Together, the analyses of open, query, and transfer functions reveal consistent architectural trade-offs. Ethereum’s permissionless design emphasizes openness and security but sacrifices throughput and latency, limiting its suitability for high-volume enterprise use. Hyperledger Fabric, tailored for permissioned environments, achieves higher throughput, bounded latency, and more efficient resource use. Queries scale well in both systems, though Fabric remains more efficient.

The results of this study also have direct practical implications for the planned application. Since such systems demand predictable throughput, low latency, and efficient resource utilization to support frequent credits, debits, and balance queries, Hyperledger Fabric emerges as the more suitable platform. Its ability to sustain consistent performance up to 1000 TPS ensures reliability for community-scale deployments, where transaction volumes are moderate but require high integrity. Ethereum, while offering openness and higher transaction submission rates, introduces latency and resource inefficiencies that

Table 6
Performance results for ‘transfer’ function.

TPS	Send Rate (TPS)		Throughput (TPS)		Max Latency (s)		CPU Utilization (%)		Memory Usage (MB)	
	Ethereum	HLF	Ethereum	HLF	Ethereum	HLF	Ethereum	HLF	Ethereum	HLF
100	100.2	160.4	37.6	158.5	17.41	1.94	0.78375	4.69875	1505.28	519.98
200	200.4	173.1	36.7	165.1	22.41	1.94	1.08125	3.7875	1576.96	540.44
300	300.4	190.5	40.7	131.9	21.52	2.49	1.80875	2.905	1587.20	514.17
400	401.0	202.5	39.4	139.3	23.07	2.25	0.97625	1.495	1556.48	541.42
500	501.3	200.3	38.4	139.6	24.19	2.22	0.04875	0.610	1556.48	540.46
600	600.2	200.4	38.6	141.5	26.27	2.84	0.45607	4.71	1553.30	539.85
700	700.2	202.5	37.9	140.3	27.16	3.05	0.57142	5.21	1555.92	540.79
800	800.3	201.6	38.7	139.8	28.47	3.00	0.79558	5.617	1561.61	541.57
900	900.4	203.3	38.2	139.4	30.35	3.26	0.39635	6.01	1590.00	540.98
1000	1000.2	201.9	38.4	137.1	31.97	2.94	0.3383	6.401	1602.31	541.60

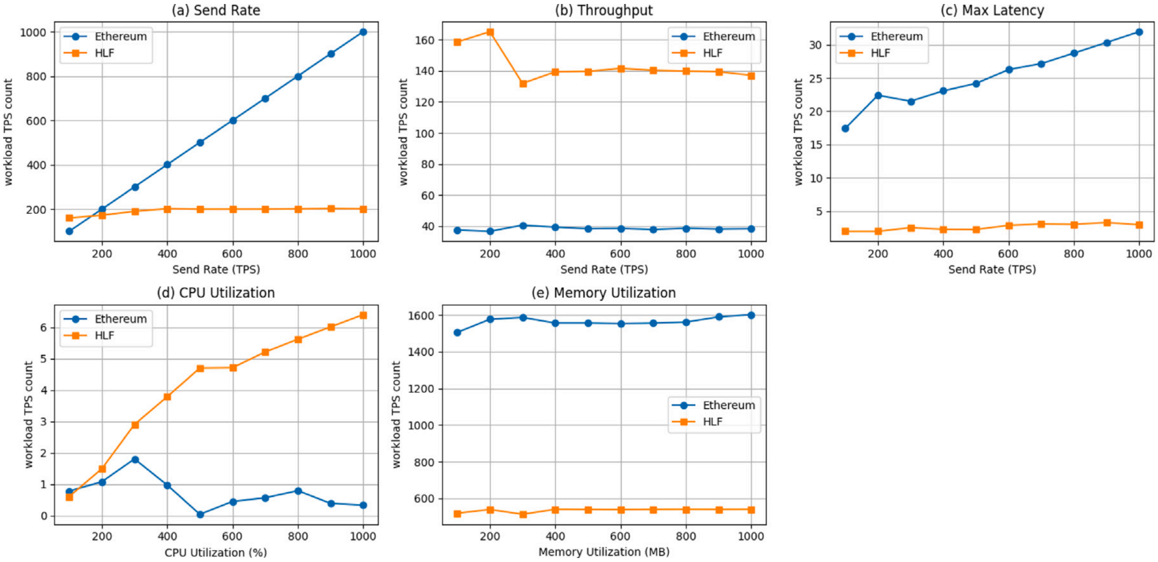


Fig. 7. Performance results for ‘transfer’ function with the varying workload from 100 to 1000.

could hinder responsiveness in a timebanking context. These insights provide a strong foundation for selecting Fabric as the underlying platform for the forthcoming application.

5.1. Future work

While this study evaluated blockchain performance by varying transaction workloads under a fixed network size, future research could extend the analysis by exploring how network size impacts system behavior. As the number of participating nodes increases, consensus mechanisms experience additional communication overhead, which may affect performance differently than transaction load. These investigations may reveal performance patterns that cannot be observed simply by adjusting the system’s workload. They can also help explain real-life situations where the performance of platforms changes depending on the number of users involved, not just the system’s overall activity.

Given the rapid evolution of blockchain technologies, this study may not cover every emerging enhancement or innovation in platform design and interaction. Future work should include new consensus protocols, updated framework versions, and wider deployment configurations to stay relevant. Building on these findings, we plan to adopt Hyperledger Fabric and move toward developing and testing a real-world application, validating the platform’s performance and usability in a live deployment scenario.

CRediT authorship contribution statement

Madhav Ajwalia: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Data curation, Conceptualization. **Parth Shah:** Writing – review & editing, Writing – original draft, Supervision.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors declare that they do not have any conflict of interest.

References

[1] M. Javaid, A. Haleem, R.P. Singh, R. Suman, S. Khan, A review of blockchain technology applications for financial services, BenchCouncil Trans. Benchmarks, Stand. Eval. 2 (3) (2022) 100073.

[2] J. Clavin, S. Duan, H. Zhang, V.P. Janeja, K.P. Joshi, Y. Yesha, L.C. Erickson, J.D. Li, Blockchains for government: use cases and challenges, Digit. Gov.: Res. Pr. 1 (3) (2020) 1–21.

[3] M. Ajwalia, K. Mer, R. Bhatia, P. Shah, P. Prajapati, Security and challenges of blockchain-based IoT use cases, in: International Conference on ICT for Sustainable Development, Springer, 2024, pp. 91–103.

[4] S. Cihan, N. Yılmaz, A. Ozsoy, O.D. Beyan, A systematic review of the blockchain application in healthcare research domain: toward a unified conceptual model, Med. Biol. Eng. Comput. (2025) 1–24.

[5] C. Zheng, X. Peng, Z. Wang, T. Ma, J. Lu, L. Chen, L. Dong, L. Wang, X. Cui, Z. Shen, A review on blockchain applications in operational technology for food and agriculture critical infrastructure, Foods 14 (2) (2025) 251.

[6] N. Kumar, K. Kumar, A. Aeron, F. Verre, Blockchain technology in supply chain management: Innovations, applications, and challenges, Telemat. Informatics Rep. (2025) 100204.

[7] T. Sivaram, et al., Recent developments and challenges using blockchain techniques for peer-to-peer energy trading: A review, Results Eng. (2024) 103666.

[8] X. Wang, M. Younas, Y. Jiang, M. Imran, N. Almusharraf, Transforming education through blockchain: A systematic review of applications, projects, and challenges, IEEE Access (2025).

[9] G. Piccardo, L. Conti, A. Martino, Blockchain technology and its potential to benefit public services provision: A short survey, Futur. Internet 16 (8) (2024) 290.

[10] N.S. Sizan, D. Dey, M.A. Layek, M.A. Uddin, E.-N. Huh, Evaluating blockchain platforms for iot applications in industry 5.0: A comprehensive review, Blockchain: Res. Appl. (2025) 100276.

[11] K. Zīle, R. Strazdiņa, Blockchain use cases and their feasibility, Appl. Comput. Syst. 23 (1) (2018) 12–20.

[12] D. B. Rawat, V. Chaudhary, R. Doku, Blockchain technology: Emerging applications and use cases for secure and trustworthy smart systems, J. Cybersecur. Priv. 1 (1) (2020) 4–18.

[13] J. Yu, X. Li, Y. Guo, A secure and verifiable blockchain-based framework for personal data validation, Computers 13 (9) (2024) 240.

[14] H. Alanzi, M. Alkhatib, Towards improving privacy and security of identity management systems using blockchain technology: A systematic review, Appl. Sci. 12 (23) (2022) 12415.

[15] D.N. Community, State of the developer nation Q4 2019, 2020, URL https://www.developernation.net/resources/reports/state-of-the-developer-nation-q4-2019/. (Accessed 11 May 2025).

[16] A.Z. Junejo, M.A. Hashmani, A.A. Alabdulatif, A survey on privacy vulnerabilities in permissionless blockchains, Int. J. Adv. Comput. Sci. Appl. (IJACSA) 11 (9) (2020) 130–139.

[17] H. Fabric, A blockchain platform for the enterprise—hyperledger-fabricdocs main documentation, 2022.

[18] R3, Corda technical whitepaper, 2025, https://r3.com/blog/corda-technical-whitepaper/. (Accessed 20 May 2025).

[19] J.K. Mudhar, J. Malhotra, S. Rani, Blockchain-based decentralized access control framework for enhanced security and privacy for consumer electronic devices, IEEE Trans. Consum. Electron. (2024).

[20] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Satoshi Nakamoto (2008).

[21] V. Buterin, et al., A next-generation smart contract and decentralized application platform, White Pap. 3 (37) (2014) 2–1.

[22] V. Capocasale, D. Gotta, G. Perboli, Comparative analysis of permissioned blockchain frameworks for industrial applications, Blockchain: Res. Appl. 4 (1) (2023) 100113.

[23] H.M. Kim, H. Turesson, M. Laskowski, A.F. Bahreini, Permissionless and permissioned, technology-focused and business needs-driven: understanding the hybrid opportunity in blockchain through a case study of insolar, IEEE Trans. Eng. Manage. 69 (3) (2020) 776–791.

[24] J. Zarrin, H. Wen Phang, L. Babu Saheer, B. Zarrin, Blockchain for decentralization of internet: prospects, trends, and challenges, Clust. Comput. 24 (4) (2021) 2841–2866.

[25] C. Ma, X. Kong, Q. Lan, Z. Zhou, The privacy protection mechanism of hyperledger fabric and its application in supply chain finance, Cybersecurity 2 (1) (2019) 1–9.

[26] G. Al-Sumaidaee, R. Alkhudary, Z. Zilic, A. Swidan, Performance analysis of a private blockchain network built on hyperledger fabric for healthcare, Inf. Process. Manage. 60 (2) (2023) 103160.

[27] H.F. Documentation, Introduction, 2025, URL https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html. (Accessed 25 May 2025).

[28] C. Fan, S. Ghaemi, H. Khazaei, P. Musilek, Performance evaluation of blockchain systems: A systematic survey, Ieee Access 8 (2020) 126927–126950.

[29] M. Touloupou, M. Themistocleous, E. Iosif, K. Christodoulou, A systematic literature review toward a blockchain benchmarking framework, IEEE Access 10 (2022) 70630–70644.

[30] F. Liu, S. He, Z. Li, Z. Li, An overview of blockchain efficient interaction technologies, Front. Blockchain 6 (2023) 996070.

[31] A.H. Lone, R. Naaz, Demystifying cryptography behind blockchains and a vision for post-quantum blockchains, in: 2020 IEEE International Conference for Innovation in Technology, INOCON, IEEE, 2020, pp. 1–6.

[32] M. Esmaili, K. Christensen, Performance modeling of public permissionless blockchains: A survey, ACM Comput. Surv. 57 (7) (2025) 1–35.

[33] H. Caliper, Hyperledger caliper, 2025, URL https://hyperledger.github.io/caliper/. (Accessed 20 May 2025).

[34] C. Melo, F. Oliveira, J. Dantas, J. Araujo, P. Pereira, R. Maciel, P. Maciel, Performance and availability evaluation of the blockchain platform hyperledger fabric, J. Supercomput. 78 (10) (2022) 12505–12527.

[35] L. Ni, E. Irannezhad, Performance analysis of LogisticChain: A blockchain platform for maritime logistics, Comput. Ind. 154 (2024) 104038.

[36] H.T. Le, T.T.L. Nguyen, T.A. Nguyen, X.S. Ha, N. Duong-Trung, Bloodchain: a blood donation network managed by blockchain technologies, Network 2 (1) (2022) 21–35.

[37] K. Antevski, C.J. Bernardos, Applying blockchain consensus mechanisms to network service federation: Analysis and performance evaluation, Comput. Netw. 234 (2023) 109913.

[38] M. Sallal, G. Owenson, D. Salman, M. Adda, Security and performance evaluation of master node protocol based reputation blockchain in the bitcoin network, Blockchain: Res. Appl. 3 (1) (2022) 100048.

[39] M. Mazzoni, A. Corradi, V. Di Nicola, Performance evaluation of permissioned blockchains for financial applications: The ConsenSys quorum case study, Blockchain: Res. Appl. 3 (1) (2022) 100026.

[40] K. Ntolkeras, H. Sharif, S.D. Salmasi, W. Knottenbelt, Performance analysis of a hyperledger iroha blockchain framework used in the UK livestock industry, in: 2021 IEEE International Conference on Blockchain (Blockchain), IEEE, 2021, pp. 456–461.

[41] P.M. Dhulavvagol, V.H. Bhajantri, S. Totad, Blockchain ethereum clients performance analysis considering E-voting application, Procedia Comput. Sci. 167 (2020) 2506–2515.

[42] O. Novo, Scalable access management in IoT using blockchain: A performance evaluation, IEEE Internet Things J. 6 (3) (2018) 4694–4701.

[43] K. Suankaewmanee, D.T. Hoang, D. Niyato, S. Sawadsitang, P. Wang, Z. Han, Performance analysis and application of mobile blockchain, in: 2018 International Conference on Computing, Networking and Communications, ICNC, IEEE, 2018, pp. 642–646.

[44] H. Caliper, Architecture, 2025, URL https://hyperledger.github.io/caliper/v0.4.2/architecture/. (Accessed 30 May 2025).

[45] S. Smetanin, A. Ometov, M. Komarov, P. Masek, Y. Koucheryavy, Blockchain evaluation approaches: State-of-the-art and future perspective, Sensors 20 (12) (2020) 3358.

[46] S.M.H. Bamakan, A. Motavali, A.B. Bondarti, A survey of blockchain consensus algorithms performance evaluation criteria, Expert Syst. Appl. 154 (2020) 113385.

[47] M. Ajwalia, P. Shah, Performance evaluation of blockchain systems: Parameters, criteria and modeling techniques, in: 2024 IEEE/ACM 17th International Conference on Utility and Cloud Computing, UCC, IEEE, 2024, pp. 256–258.

[48] Y. Ucbas, A. Eleyan, M. Hammoudeh, M. Alohaly, Performance and scalability analysis of ethereum and hyperledger fabric, IEEE Access 11 (2023) 67156–67167.

[49] N. Anita, M. Vijayalakshmi, S.M. Shalini, K.D. Lakshmi, Blockchain consensus algorithms and platforms: a survey, J. Manag. Anal. (2025) 1–37.

[50] P.H.B. Correia, M.A. Marques, M.A. Simplicio, L. Ermlivitch, C.C. Miers, M.A. Pillon, Comparative analysis of permissioned blockchains: Cosmos, hyperledger fabric, quorum, and XRPL, in: 2024 IEEE International Conference on Blockchain (Blockchain), IEEE, 2024, pp. 464–469.

[51] J. Polge, J. Robert, Y. Le Traon, Permissioned blockchain frameworks in the industry: A comparison, Ict Express 7 (2) (2021) 229–233.

[52] S. Sarker, A.K. Saha, M.S. Ferdous, A survey on blockchain & cloud integration, in: 2020 23rd International Conference on Computer and Information Technology, ICCIT, IEEE, 2020, pp. 1–7.

[53] H. Wu, Q. Yao, Z. Liu, B. Huang, Y. Zhuang, H. Tang, E. Liu, Blockchain for finance: A survey, IET Blockchain 4 (2) (2024) 101–123.