


Article

Blockchain Bottleneck Analysis Based on Performance Metrics Causality

Wei hu Song ^{1,2} , Mengxiao Zhu ³, Dong Lu ¹, Chen Zhu ¹, Jiejie Zhao ², Yi Sun ⁴, Lei Li ^{2,*} and Haogang Zhu ^{1,2,5,*}

¹ State Key Laboratory of Complex & Critical Software Environment, Beihang University, Beijing 100191, China; weihusong@buaa.edu.cn (W.S.)

² Zhongguancun Laboratory, Beijing 100086, China

³ School of Information Science and Technology, North China University of Technology, Beijing 100144, China

⁴ Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100045, China

⁵ Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Institute of Artificial Intelligence, Beihang University, Beijing 100191, China

* Correspondence: lil@mail.zgclab.edu.cn (L.L.); haogangzhu@buaa.edu.cn (H.Z.)

Abstract: With the widespread application of blockchain technology across various industries, detecting and analyzing performance bottlenecks is crucial for evaluating and optimizing blockchain system performance. However, current research needs general performance metrics for detecting and analyzing bottlenecks. Only some studies focus on this aspect within blockchain systems. To address this, this paper first proposes 18 fine-grained performance metrics to evaluate performance across various layers of blockchain systems comprehensively. Subsequently, we introduce a generalized loosely coupled performance measurement framework to capture these metrics and construct the causal relationship between them, i.e., the mesoscopic performance structure. This approach allows for the detection and analysis of performance bottlenecks. Finally, numerous experimental results demonstrate that the causality between the relevant performance metrics disappears when the system reaches a performance bottleneck. Additionally, the framework has a performance impact of less than 15% on ChainMaker.

Keywords: blockchain; consortium; causal inference; bottleneck analysis



Citation: Song, W.; Zhu, M.; Lu, D.; Zhu, C.; Zhao, J.; Sun, Y.; Li, L.; Zhu, H. Blockchain Bottleneck Analysis Based on Performance Metrics Causality. *Electronics* **2024**, *13*, 4236. <https://doi.org/10.3390/electronics13214236>

Academic Editor: Mehdi Sookhak

Received: 9 September 2024

Revised: 17 October 2024

Accepted: 24 October 2024

Published: 29 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the widespread application of blockchain technology across various industries, the evaluation and optimization of blockchain systems have become particularly important. Performance metrics for blockchain systems are critical for understanding system behavior, identifying potential bottlenecks, and supporting performance optimization. Therefore, while individual blockchain performance metrics are not independent, but often correlated, and causal inference of blockchain performance metrics is imperative.

In previous studies, research on blockchain performance metrics has primarily focused on data collection, specifically monitoring the performance of individual blockchain platforms or conducting horizontal comparisons across multiple blockchain platforms.

In previous studies, the research on blockchain performance metrics has primarily focused on data collection, specifically monitoring the performance of individual blockchain platforms [1] or conducting horizontal comparisons across multiple blockchain platforms [2–5]. However, challenges still need to be addressed due to the limitations of existing data collection tools. First, current data collection frameworks have certain limitations. They usually only capture core and limited performance metrics that cannot fully reflect the complexity and diversity of blockchain systems. Second, performance metrics collection frameworks tailored for different blockchain platforms have deployment difficulties that limit data collection and analysis. In addition, there has yet to be prior work on causal inference of blockchain performance metrics, i.e., the analysis of mesoscopic

performance structure. These challenges hinder our understanding of system behavior and potential bottlenecks. Our experiment is indicated in Figure 1.

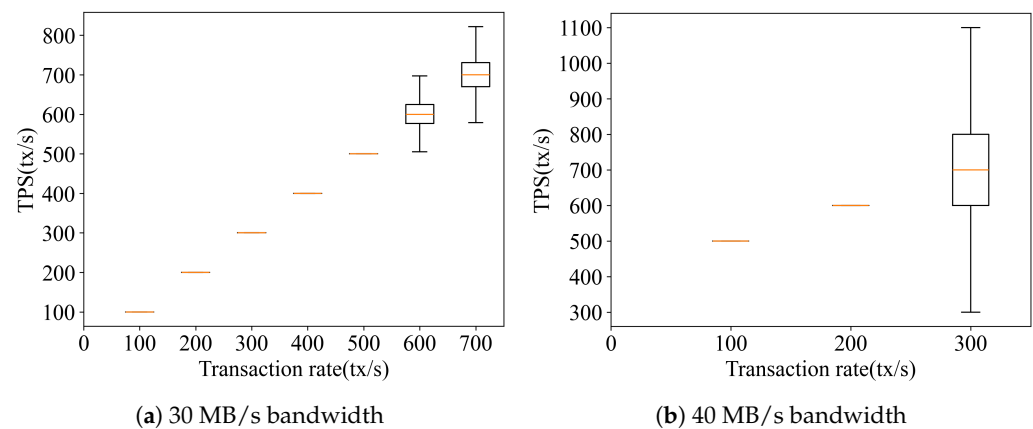


Figure 1. TPS for different bandwidths with different transaction rates.

When using ChainMaker [6] at 30 MB/s bandwidth, the median, upper quartile, and lower quartile of the Transactions per Second (TPSs) are the same when the transaction rate is less than 500, indicating that the current system can efficiently process all the transactions at the lower transaction rate, and that the system begins to encounter a bottleneck when it is over 500. The rate of growth begins to become slower, the TPSs' median is still increasing, and this trend does not imply a substantial improvement in system efficiency, instead signaling that the system is approaching its bottleneck. When the bandwidth increases to 40 MB/s, the transaction rate bottleneck rises to 600.

Causal inference represents a burgeoning field, extending across diverse domains, including medicine [7,8], climate science [9,10], and its integration with machine learning techniques [11,12] and Graph Neural Networks (GNNs) [13,14]. However, research on causal inference within blockchain technology remains scarce, primarily focusing on delineating causal relationships between different cryptocurrencies [15,16] and their causal relationships with other factors [17,18]. Notably, to our knowledge, we are the first to introduce causal inference into blockchain performance metrics causality for elucidating causal relationships among different performance metrics.

To address the aforementioned challenges, this paper proposes a novel approach that integrates the underlying implementation of blockchain, abstracting a set of fine-grained performance metrics that are relatively common among different blockchain platforms and capable of reflecting the working states of various level modules. Additionally, we establish a generalized loosely coupled measurement framework that is not restricted to specific blockchain platforms. Our method enables performance measurement for different blockchain applications, and provides a comprehensive set of fine-grained performance metrics to evaluate blockchain system performance under various conditions. We can better identify potential system bottlenecks and provide quantitative support for blockchain performance optimization by employing a measurement framework to obtain fine-grained performance metrics.

In this paper, we conduct experiments on the causality between performance metrics through a detailed analysis of performance metric data and further delve into the bottleneck analysis of blockchain systems. By gaining an in-depth understanding of the interactions and impact of performance metrics, we can better understand the state of the blockchain system, paving the way for targeted optimization strategies.

The main contributions of this paper can be summarized as follows:

- This paper proposes 18 fine-grained performance metrics on contract, network, data, consensus, and system layers for comprehensively evaluating the performance of blockchain systems under different conditions.

- Given the limitations of existing performance collection tools, this paper provides a generalized loosely coupled measurement framework to obtain comprehensive, fine-grained blockchain performance metrics for different blockchain implementations, including ChainMaker [6], Ethereum [19], and FISCO BCOS [20], and the framework's impact on ChainMaker is less than 15%.
- This paper conducts causal inference analysis on performance metrics, constructing the mesoscopic performance structure between performance metrics, and delves into the relationships, providing a different perspective for understanding the behavior of blockchain systems and potential bottleneck issues.
- Extensive experiments demonstrate that our approach can identify causality between performance metrics when the system reaches a bottleneck.

The organizational structure of this paper is as follows: The second part will review related research and existing performance metric frameworks. The third part will introduce our fine-grained performance metrics, the universal loosely coupled measurement framework, and causality analysis between performance metrics. The fourth part will discuss our research findings on blockchain bottleneck analysis. Finally, the sixth part will summarize the paper and propose prospects for future work.

2. Related Work

Anomaly detection remains a ubiquitous concern across diverse domains, and the blockchain field is no exception. Ref. [21]'s solution based on Ethereum analyzes summarized block data structures to identify suspicious accounts, effectively reducing time complexity while maintaining high accuracy. Ref. [22] employs a dynamic attribute graph network construction method to model each transaction, utilizing edges to provide additional learnable transaction attribute information. This approach facilitates graph representation learning in blockchain networks, offering a novel and effective solution for anomaly detection on the blockchain. This research in this area has predominantly focused on anomaly detection in the financial activities of public blockchain networks. However, consortium blockchain platforms do not generate the kind of financial irregularities observed in public chains, because no mining is required. Instead, the core functionality of consortium blockchain platforms is closely tied to system stability, as these systems prioritize operational efficiency. Therefore, in this case, behaviors that affect the stability and efficiency of the consortium blockchain system can be considered abnormal. System-level performance bottlenecks are one of them.

2.1. Performance Bottlenecks

There are some current research on blockchain performance bottlenecks. Initially, some macro-level studies of the blockchain protocol layer were conducted. In the storage layer, ref. [23] research focuses on the architecture of Hyperledger Fabric Blockchain Systems (HFBS), revealing that the performance of the state database is highly dependent on read transactions, rather than write transactions. The study identifies that the performance bottleneck in read operations is attributed to CouchDB's inability to maintain high performance under a large volume of query operations. Within the consensus layer, ref. [24] proposes a method to evaluate the performance of consensus algorithms in private blockchain platforms, such as Ethereum and Hyperledger Fabric. The study yields performance assessment results of the consensus algorithms under varying transaction volumes through quantitative analysis of latency and throughput. The findings indicate that consensus mechanisms can lead to performance bottlenecks. Ref. [25] investigates the performance and scalability of Byzantine Fault-Tolerance (BFT) consensus protocols widely used in permissioned blockchain systems. The study compares the performance of these protocols under identical conditions through theoretical analysis (load formulas), as well as practical implementation and evaluation. The findings reveal that scalability could be better as the number of validators increases, identifying communication complexity as the primary cause of this issue. Ref. [26] highlights that the insufficient performance of current

blockchain systems is one of the critical limitations hindering the global realization of the Web 3.0 revolution. The design of ALDER leverages the presence of multiple potential leaders to alleviate bottlenecks in various aspects of consensus protocols. Ref. [27] analyzes how Bitcoin's current peer-to-peer overlay network is constrained by fundamental and technical bottlenecks, limiting its capacity to support higher throughput and lower latency. The findings indicate that merely adjusting block size and interval can only serve as an initial step toward achieving the next generation of high-load blockchain protocols; significant progress requires a fundamental rethinking of the technological approach. Due to the performance complexity of distributed systems, many performance characteristics of the latest version of Hyperledger Fabric—such as the performance features at various stages, the impact of the ordering service, bottlenecks, and scalability—remain inadequately understood. Ref. [28] finds that the verification stage is likely to become a system bottleneck due to the relatively low speed of chain code verification. Ref. [29] conducts a comprehensive empirical study of Hyperledger Fabric to characterize its performance and identify potential bottlenecks. The research identifies three primary performance bottlenecks: endorsement policy verification, ordering policy verification for transactions within blocks, and state validation and commitment (when using CouchDB). Additionally, ref. [30] proposes a novel theoretical model for calculating transaction latency under various network configurations, such as block size and interval. The study identifies several performance bottlenecks and provides insights from a developer's perspective. ref. [31] investigates the performance of private Ethereum blockchains through an in-depth analysis of functional-level bottlenecks and conducts a series of experiments to identify bottleneck functions invoked each time a transaction reaches an Ethereum node. ref. [32] systematically examines how the performance of private Ethereum blockchains scales with variations in various parameters and identifies which parameters constitute bottlenecks. The findings indicate that the effect of changes in one parameter is highly dependent on the configuration of other parameters, mainly when the system is operating near its limits.

2.2. Blockchain Performance Framework

Numerous studies have extensively examined the measurement and analysis of blockchain performance. The increasing complexity of blockchain systems has heightened the challenge of analysis. Therefore, research typically concentrates on specific blockchain platforms or components or employs simulation methods to quantify performance analysis. Among these, BlockBench [2] is the first evaluation framework designed for analyzing private blockchains. It measures both overall performance and the performance of individual components, including throughput, latency, scalability, and fault tolerance. The results of horizontal comparisons of Ethereum, Parity, and Hyperledger Fabric indicate that these systems still need to replace current database systems for traditional data processing workloads. Hyperledger Caliper [33] is a more comprehensive and sophisticated tool that focuses on fundamental performance metrics of blockchain, such as throughput and latency. It allows users to construct complex testing scenarios to simulate real-world business logic. ConsenSys [34] delineates crucial considerations for blockchain performance testing and benchmarking, including methodologies, performance metrics, and benchmarking tools. Blockmeter [3] is an application-agnostic performance benchmarking framework designed for private blockchain platforms. This framework can measure the critical performance indicators of any application deployed on external private blockchain applications, enabling enterprises to understand better how private blockchain platforms perform in their specific application scenarios. Hyperledger Fabric [35] and Hyperledger Sawtooth [36], across varied applications and configuration parameters. Alsahan [37] introduced a new blockchain network simulator focusing primarily on Bitcoin's original reference implementation. This simulator employs lightweight virtualization techniques to create finely tuned local testing networks. The simulator can adjust Bitcoin mining difficulty to levels below the default minimum to facilitate rapid simulation of large-scale networks without shutting down mining services. BTCSpark [38] is introduced as an open-source tool for

Bitcoin analysis, and it provides researchers and developers with an easy-to-use, flexible, and high-performance environment for querying the blockchain and building blockchain analysis tools.

3. Method

This section describes this paper's methodology, including microscopic performance metrics, a generalized loosely coupled measurement framework, and mesoscopic performance metrics.

3.1. Microscopic Performance Metrics

This paper introduces microscopic performance metrics to evaluate and optimize blockchain system performance. Performance metrics enable an in-depth understanding of system behavior, identification of bottlenecks, and support for performance optimization. Existing performance metrics need to capture the complexity and diversity of systems. Therefore, we propose a set of fine-grained performance metrics aimed at comprehensively evaluating the performance of blockchain systems while focusing on bottlenecks. By introducing these metrics, we can accurately identify potential bottlenecks and provide quantitative support.

To take into account the universality and comprehensiveness of the performance metrics, we expand on [2,33,34,39] to include 18 micro-performance metrics from the consensus, network, data, contract, and system layers of the blockchain. Most of these metrics are applicable across different blockchain implementations, such as CPU utilization, memory utilization, TPS, latency, commit block delay, transaction queue delay, transaction pool input throughput, block verification efficiency, average consensus time per round, proposal time, pre-vote time, pre-commit time, commit time, peer message throughput, average transmission latency, state data read throughput, state data write throughput, and transaction conflict rate. Several performance metrics related to the consensus methodology, the time spent in the four stages of the TBFT [40] consensus proposal, pre-vote, pre-commit, and commit.

3.1.1. Consensus Layer

Transactions per Second. TPSs represent the rate at which transactions are packaged and stored in a block per unit of time, constituting the most core performance metric in the blockchain.

Latency, **Transaction Confirmation Delay (TCD)** refers to the duration from the time a transaction (tx) enters the transaction pool at the time to the time the transaction is stored in the block at the time, representing the entire lifecycle of the transaction from start to completion.

Commit Block Delay. CBD indicates the time interval from a successful block proposal to block storage. This metric is a key factor affecting latency.

Transactions Queue Delay. TQD is the elapsed time a transaction spends in the pool waiting to be processed and packaged, specifically from the transaction that enters the pool until it leaves. TQD is the determining factor that affects latency.

Transaction Pool Input Throughput. TPIT denotes the rate at which transactions enter the transaction pool per unit time, representing the pace at which transactions enter the blockchain network and await processing.

Block Verification Efficiency. BVE denotes the ratio of the time required to validate a block to the number of transactions contained in the block. This metric evaluates the efficiency of validation nodes in processing transactions and validating blocks in the blockchain network.

Average Consensus Time Per Round. ACTPR represents the average time held for each stage of consensus per unit of time and the overall average time.

The duration of consensus phases. They refer to the time taken for each step in the consensus algorithm. Using TBFT as an example, it comprises four main stages:

proposal, pre-vote, pre-commit, and commit. The calculation method for each stage involves subtracting the start time from the end time.

3.1.2. Network Layer

Average Transmission Latency. ATL is the average time required for two-way communication between two nodes and represents the duration from sending a message to receiving it. First, node A sends a message to node B and records the sending moment (t_1). Node B receives the message and records the receiving moment (t_2). At this point, the transmission elapsed time of node A is ($t_2 - t_1$). The roles of nodes A and B are swapped, and repeat the above process, at which time the transmission elapsed time of node B is ($t_4 - t_3$). Finally, the average transmission delay is obtained by summing the transmission elapsed time of node A and node B and taking the average value.

Peer Message Throughput. PMT The sum of message sizes sent and received by a node over a fixed period. Specifically, it is the ratio of cumulative message size sum and elapsed time over a fixed period.

3.1.3. Storage Layer

State Data Read Throughput. SDRT refers to the amount of status data read during a specific time interval. This metric is derived by accumulating the ratio of status data read in each read operation to the time taken.

State Data Write Throughput. SDWT is the amount of status data written in a given time interval. The metric is a cumulative ratio of state data written in each write operation to the time spent.

3.1.4. Contract Layer

Transaction Conflict Rate. TCR represents the probability or frequency of transaction conflicts in a blockchain network. Transaction conflict occurs when multiple transactions are for inclusion in the same block, resulting in only one being included while others may be delayed or rejected.

3.1.5. System Layer

Resource utilization. CPU and memory utilization are crucial metrics for assessing the usage of computational resources in a computer system. This value indicates the extent of CPU and memory resource usage, where higher percentages signify more efficient utilization, while lower percentages denote relatively idle resources.

3.2. Generalized Loosely Coupled Measurement Framework

The measurement framework proposed in this paper aims to overcome the limitations of existing frameworks and fulfill the need for fine-grained performance metrics data collection. Existing data collection frameworks usually only capture core performance metrics, which cannot fully reflect the complexity and diversity of blockchain systems. In addition, difficulties in deploying different blockchain platforms limit data collection and analysis. Thus, our proposed measurement framework can capture comprehensive and diverse performance metrics data for different blockchain implementations.

The proposed generalized loosely coupled measurement framework, depicted in Figure 2, comprises four components: the definition layer, data storage layer, configuration layer, and processing layer.

- The definition layer defines the micro-performance metrics measurements in an object-oriented manner and maps them to the actual stored files. Suppose the corresponding performance metrics file does not exist in the specified directory. In that case, the framework generates the corresponding performance metrics file and initializes the table header according to its object definition.
- The data storage layer stores vast raw data points for calculating performance metrics, facilitating rapid data input sequences. Asynchronous write operations are

safely implemented through encapsulated routines to minimize the impact of parallel write operations on the primary process and ensure compatibility with its exception handling.

- The configuration layer facilitates the configuration, loading, and deployment of raw data points sampling for each node and performance metrics through global and local switches, offering runtime flexibility.
- The processing layer hosts an indicator measurement library and a performance measurement thread pool, ensuring real-time calculation of raw data points obtained by the data storage layer to derive real-time estimates of performance metrics.

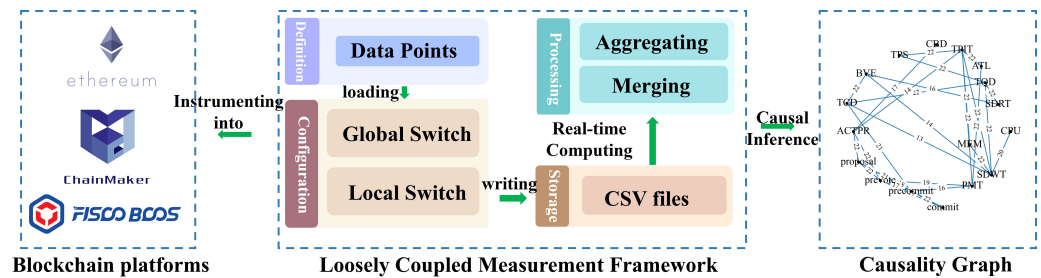


Figure 2. Workflow of loosely coupled measurement framework.

The workflow of the framework is as follows: firstly, the data points in the definition layer are added or subtracted according to the actual situation, and then the blockchain nodes are configured, created, and started, at which time the initialization of all the data points complete, generate the corresponding data files, and add the table headers. Next, the configuration layer configures the framework, and in the contract invocation process, it is possible to control whether the data are written or not by real-time adjustment of the global and local switches when the global switch is off, any performance indicator stops written, and when the global switch is on, and a performance indicator switch (local switch) is off, the performance indicator stops write. In addition, we can view the status of the global and local switches at any time. Then, when writing data, the data storage layer performs fast asynchronous writes for large amounts of data while handling exceptions. Finally, the processing layer computes the collected data, converts the data points into performance metrics, and generates mesoscopic performance structures in real time.

In addition, we applied the framework on ChainMaker, Ether, and FISCO BCOS for the application of micro-performance metrics; the framework implements all of the above micro-performance metrics in ChainMaker. For BCOS and Ether, the metrics are consistent, except for the contract conflict metric, which is not implemented on both platforms and the time spent in different phases by different consensus protocols.

3.3. Mesoscopic Performance Structure

The introduction of causal inference methods for performance metrics in this paper aims to address the current challenges in understanding the bottleneck of blockchain systems. Although existing research has focused on performance metrics, studies on the causality between performance metrics still need to be completed. Therefore, introducing causal inference methods can delve deeper into the relationships between performance metrics and reveal potential bottleneck issues. By analyzing causal graphs, we can accurately identify the causality between performance metrics and pinpoint crucial factors leading to system bottlenecks.

Blockchain performance bottlenecks are characterized by performance metrics peaking under stress, indicating saturation. Before saturation, modules interact to drive transaction processing, and performance metrics possess correlations with each other. After saturation, there is minimal change in the performance metrics, and the interdependencies weaken. This causal relationship can represent performance metrics graphically, reflecting the state of the blockchain's operation on a mesoscopic.

In this paper, we construct causal graphs of mesoscopic performance structures using a combination of Bayesian and graph structure searches. First, we use the A-star algorithm to initialize the relationships between performance metrics for all performance metrics data to create the initial structure graph. Then, we traverse the data using a sliding window, use A-star to obtain the causal graph for each window, add the set of causal graphs, aggregate all the graphs in the set to obtain the set of weights, and finally reconstruct the causal graph according to the set of weights. Please refer to Algorithms 1 and 2 for details.

Algorithm 1 A approach for constructing the causality graph of mesoscopic performance structure

Input:

Performance metrics Data D

Output:

Causality graph G_r

- 1: Initialize adjacency matrix A
 - 2: Create baseline graph using A* algorithm $G = A^* [41] (A, D)$
 - 3: Initialize set of causal graphs $\Gamma = \{\}$
 - 4: for $t = 1$ to Iterate through sliding window T do
 - 5: $G_t = A^*(G, A, D[t - 1, t])$
 - 6: $\Gamma = \Gamma \cup \{G_t\}$
 - 7: end for
 - 8: $W = \text{AggregateWeights}(\Gamma)$
 - 9: $G_r = \text{ReconstructCausalGraph}(W)$
 - 10: **return** G_r
-

Algorithm 2 AggregateWeights

Input:

Set of causal graphs Γ

Output:

Weight sets for causal graph W

- 1: Weight sets for causal graph $W = \{\}$
 - 2: **for** each edge (i, j) in $\bigcup_{G \in \Gamma} G$ do
 - 3: $W_{i,j} = \sum_{G \in \Gamma} \mathbf{1}[(i, j) \in G]$
 - 4: end for
 - 5: **return** W
-

4. Results

4.1. Experiment Setup

4.1.1. Experiment Settings

We used ChainMaker v2.3.1 and conducted experiments on the Huawei Cloud Kubernetes (k8s) cluster. Each node was constrained to 12 GB of memory and 3 CPU cores; the physical machine on which each node resides is configured with Huawei Cloud's EulerOS 2.0 operating system, using Intel Xeon Gold 6240 processors, with 32 vCPUs, 64 GB of RAM, and 350 GB of storage capacity allocated. Experiment 1 aimed to assess the impact of the loosely coupled measurement framework on the blockchain's main process performance. Stress tests were conducted on 4, 7, and 10 nodes, with transaction volumes ranging from 1000 to 50,000 transactions and a bandwidth limit of 30 MB/s.

The objective of Experiment 2 was to quantify the decay process of the blockchain performance structure and investigate its potential correlation with system bottlenecks. Performance variations under different transaction pressures were investigated using four nodes. Experiments with transaction rates ranging from 100 to 700 TPS at 30 MB/s bandwidth and experiments with transaction rates ranging from 500 to 700 TPS at 40 MB/s bandwidth were run for 10 h.

Algorithm 1 performs causal inference on all the performance metrics data collected. Specifically, a baseline performance structure was first established using all data. Subse-

quently, all data were traversed using a sliding window approach with a window size of 10,800 and a step size of 1200, resulting in 22 directed causal graphs. These graphs were then traversed and accumulated based on the number of edges appearing between every two nodes, and their accumulation was used as weights, with the magnitude of the weights representing the strength of the causal relationship between the performance metrics, resulting in a frequency-weighted directed causality graph.

4.1.2. Metrics

For Experiment 1, we choose four imperative indicators for evaluating the performance of the blockchain to illustrate the degree of influence of the framework, the TPS peak, the elapsed time for processing transactions, the average block confirmation latency, and the block confirmation latency peak, which can comprehensively reflect the throughput capacity of the blockchain system, the efficiency of the transaction processing, as well as the speed and stability of the final confirmation, and are the key indicators for measuring the performance of the blockchain.

For Experiment 2, we chose the median, upper quartile, and lower quartile to evaluate the change in TPS. For the performance metric causality, we decided the number of nodes and edges as a measure of the complexity of the causal graph and the disappearance of edges between nodes as the emergence of its bottleneck.

4.2. Framework Evaluation

In a 10-node environment, Figure 3 illustrates the impact of the loosely coupled measurement framework on blockchain main process performance. When the framework is disabled, block confirmation latency remains stable below 40 ms. Enabling the framework results in slightly higher confirmation latency, but it is still stable below 40 ms. As transaction pressure increases, block transaction confirmation latency remains stable in both scenarios. Regarding TPS, turning off the framework offers a more significant advantage at lower transaction volumes, with the TPS gap between enabling and disabling gradually stabilizing as transaction volume increases.

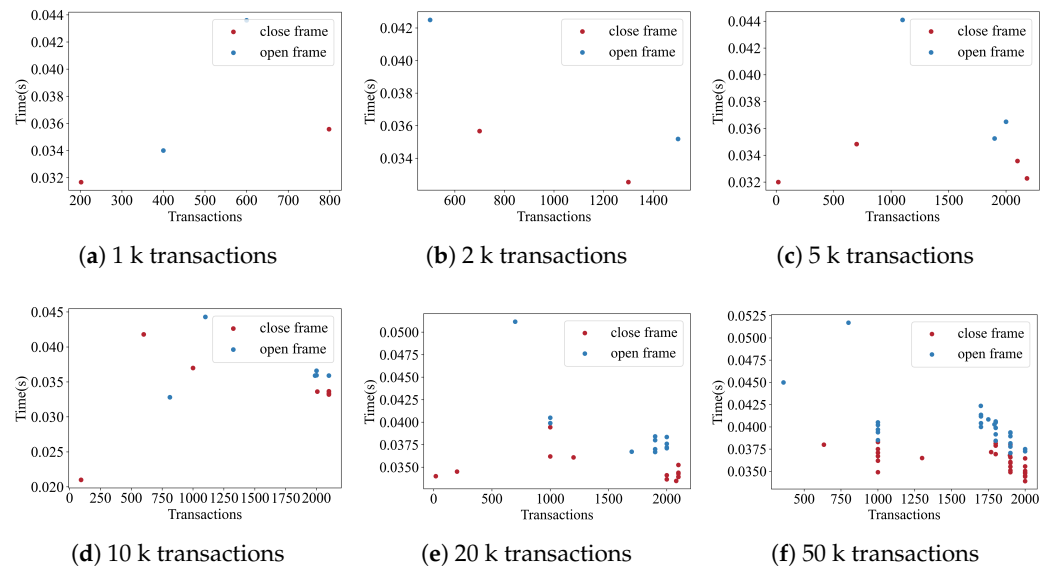


Figure 3. Impact of the framework on the system at different transactions.

Table 1 shows that, for the same transaction volume, the increase in the time taken to process these transactions is no more than 1s in the case of using the framework. While there is a significant difference in the TPS peaks between the two instances when the transaction volume is low, this gap gradually narrows as the transaction volume increases. It is worth noting that the data peaks for block confirmation latency are primarily below 40 ms when the framework is enabled. In addition, the impact of framing on the average

block confirmation latency decreases as transaction volume increases, with the effect of framing on the total elapsed time dropping to 3.3% at a transaction volume of 50,000.

Table 1. Impact of the framework on the blockchain under 10 nodes at different transactions.

Transactions		Tps Peak (tx/s)	Block Confirmation Latency Peak (ms)	Average Block Confirmation Latency (ms)	Time (s)
1 k	open	798	44	39 (+14.7%)	2 (+0)
	close	600	36	34	2
2 k	open	1500	42	39 (+14.7%)	2 (+0)
	close	1300	36	34	2
5 k	open	2182	44	39 (+18.2%)	4 (+33.3%)
	close	2000	35	33	3
10 k	open	2100	44	37 (+12.1%)	7 (+16.7%)
	close	2100	42	33	6
20 k	open	2100	51	39 (+11.4%)	13 (+8.3%)
	close	2000	39	35	12
50 k	open	2000	52	40 (+11.1%)	31 (+3.3%)
	close	2000	38	36	30

In summary, the loosely coupled performance measurement framework has a discernible impact on blockchain performance, albeit diminishing as transaction volume increases. While enabling the framework may result in slight performance declines in particular metrics, its overall impact remains minimal.

Given that ChainMaker defaults to a transaction pool size of 50,000, this study evaluated the framework’s impact under various node counts using this transaction volume. As shown in Figure 4, block confirmation latency tends to increase with the number of nodes when the performance measurement framework is enabled, peaking at 40 ms, 48 ms, and 52.5 ms. However, in most cases, block confirmation latency remains below 40 ms. The TPS gap between different configurations diminishes with increasing node count, indicating a minor impact of the performance framework on blockchain performance.

In conclusion, although the performance measurement framework may affect block confirmation latency and TPS to some extent, its overall impact on blockchain performance is minor, especially under conditions of a large number of nodes and high transaction volumes. Therefore, this measurement framework proves suitable for real-time performance measurement during blockchain operation.

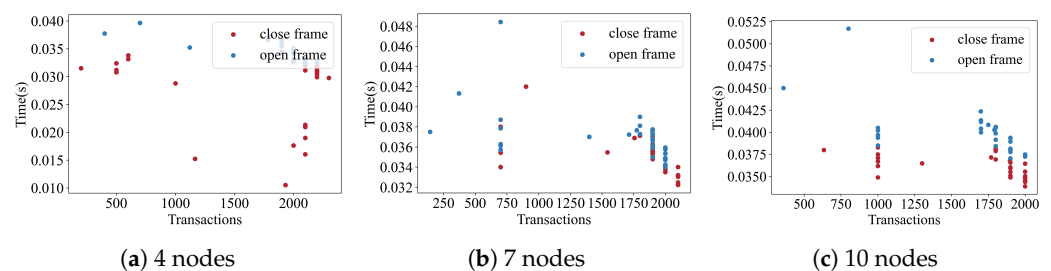


Figure 4. Impact of the framework on the system at different transaction rates.

4.3. Bottleneck Analysis

As can be seen in Figure 5, from the results of seven experiments with different transaction rates, at 30m bandwidth, when the transaction rate is lower than 500, the weights of the edges between TPS and the TPIT are 22 and have performed 22 sliding window operations on each batch of data to reason about the causal relationship between the performance metrics, i.e., we determine the existence of a causal relationship between the two metrics in each of the reasoning, which indicates that there is a very stable causality.

When the transaction rate is over 500, the causal relationship between the two indicators disappears. We speculate that there is a bottleneck in the system at this point. To verify the speculation, trends in transaction pressure, node count, and edge count were plotted from the frequency-weighted graph, as shown in Figure 6a. The graph illustrates that as transaction rates increase, the number of nodes and edges represent causal relationships between performance. The graph illustrates that as transaction pressure increases, the number of nodes and edges representing causal relationships between performance metrics rises, peaking at a transaction rate of 500.

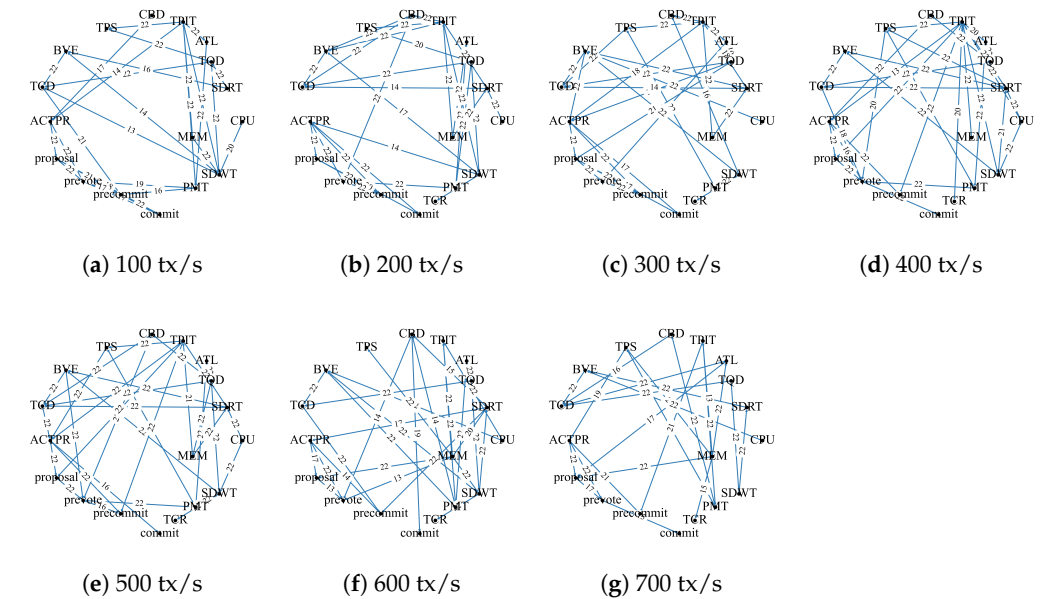


Figure 5. ChainMaker frequency-weighted causality graph comparison with transaction rates range from 100 tx/s to 700 tx/s with a bandwidth of 30 MB/s. The nodes represent performance metrics, the edges represent the existence of a causal relationship, and the greater the weight of an edge between two nodes, the stronger the causal relationship between these two performance metrics.

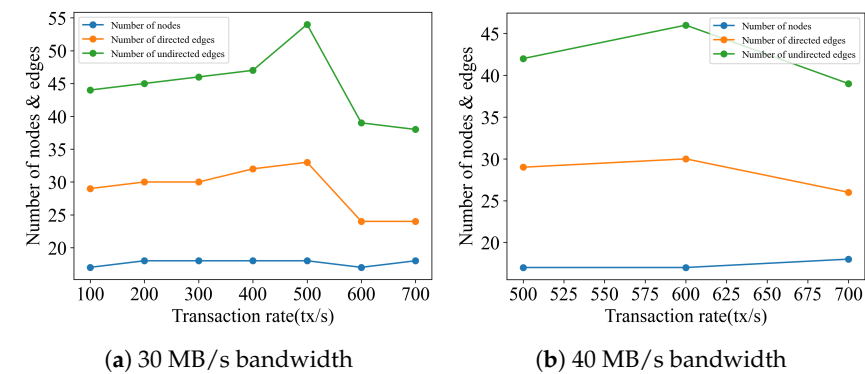


Figure 6. Number of nodes and edges at different transaction rates for different bandwidths.

In addition, we can see a visual comparison of the transaction pressure data across the seven experiments, as shown in Figure 1a. The TPS increases with the transaction rate, with an abrupt change at transaction rates greater than 500. Specifically, the median, upper quartile, and lower quartile of TPS are the same in the range of transaction rates from 100 to 500, indicating that the system can efficiently process all transactions at lower rates. At this point, TPS and transaction rate show a linear relationship. TPS increases as the transaction rate increases because the system resources still need to be maximally utilized. When the transaction tx rate further increases to 600, the system reaches a bottleneck. At this time, even

if the transaction rate continues to increase, the increase in TPS will slow down, which may indicate that the system is approaching its limit.

Based on the above conclusion, we can further speculate that when the network bandwidth increases, the transaction rate that causes the system to reach the bottleneck should also increase, i.e., the bottleneck threshold will be greater than 500, so we conducted experiments at transaction rates of 500, 600, and 700 at 40 MB/s bandwidth, and as can be seen in Figure 7, similarly to that of 30 MB/s bandwidth, the experimental results show that there is a stable causal relationship between TPS and TPIT for transaction rates of 500 and 600, and the causal relationship disappears when the transaction rate exceeds 600. Similarly, we can observe that edges and nodes peak at 600 from Figure 6b, and the change in TPS slows down the growth rate at greater than 600 from Figure 1b. This means that the system bottleneck will reach a transaction rate of 600.

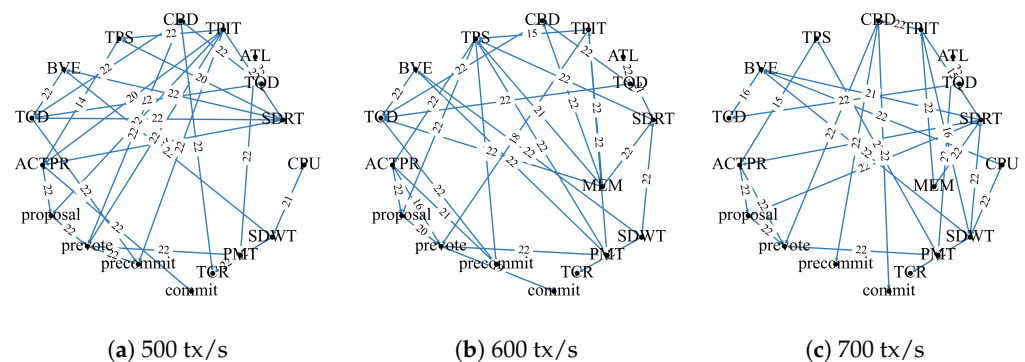


Figure 7. ChainMaker frequency-weighted causality graph comparison: transaction rates range from 500 tx/s to 700 tx/s, with a bandwidth of 40 MB/s. The nodes represent performance metrics, and the edges represent the existence of a causal relationship. The greater the weight of an edge between two nodes, the stronger the causal relationship between these two performance metrics.

From the experimental results, we can draw a clear conclusion: under varying network bandwidth and transaction rate conditions, the system exhibits a stable causal relationship between performance metrics and a distinct bottleneck phenomenon. Specifically, under 30 MB/s bandwidth, when the transaction rate is below 500, a stable causal relationship exists between TPS and TPIT. However, as the transaction rate exceeds 500, this causal relationship disappears, indicating that the system has reached a bottleneck. When the transaction rate further increases to 600, the growth of TPS noticeably slows down, confirming the system's resource constraints. Similarly, under 40 MB/s bandwidth, the system's bottleneck transaction rate shifts from 500 to 600, suggesting that an increase in bandwidth can delay the system from reaching its bottleneck. Therefore, we conclude that the system's performance bottleneck is closely related to the transaction rate and network bandwidth. While the system's processing capacity improves with increased bandwidth, a bottleneck will still emerge at higher transaction rates.

5. Conclusions

This study introduces a novel causality-based approach to analyzing performance bottlenecks in blockchain systems, addressing a critical gap where general performance metrics for comprehensive evaluation were lacking. We provide a more holistic view of blockchain performance by proposing 18 fine-grained performance metrics spanning multiple layers of blockchain systems. Developing a generalized loosely coupled measurement framework enables us to capture these metrics and construct a mesoscopic performance structure, uncovering causal relationships between performance indicators under different operational conditions. This approach facilitates the detection of bottlenecks and provides deeper insights into how performance deteriorates under stress.

One of the most significant results of this work is the discovery that as blockchain systems experience increased transaction pressure and approach performance bottlenecks, the causal relationships between crucial performance metrics begin to degrade and ultimately collapse. This finding highlights the intricate interdependencies within blockchain performance, offering a new lens for understanding system behavior in high-load scenarios. Additionally, our framework imposes a minimal overhead of less than 15% on the ChainMaker blockchain, demonstrating its practical feasibility for real-world implementation without severely impacting system throughput.

These results underscore the importance of considering causal structures in performance analysis, providing a new methodology that can be applied beyond blockchain systems to other distributed systems. The insights gained from this study pave the way for more robust blockchain optimization strategies, including proactive bottleneck detection and performance tuning before critical failures occur.

The current study has limitations, and future work could further extend this framework to other blockchain platforms to validate its generality. In addition, we will also explore the changes in the causal relationships between performance metrics in the event of other system anomalies (such as an increase or decrease in the number of nodes or a change in bandwidth), promoting the broader application of blockchain technology in high-performance scenarios.

Author Contributions: Conceptualization, W.S. and D.L.; methodology, W.S.; software, W.S. and C.Z.; validation, W.S.; formal analysis, W.S.; investigation, W.S.; resources, W.S.; data curation, W.S.; writing—original draft preparation, W.S.; writing—review and editing, J.Z., H.Z., M.Z., Y.S. and L.L.; visualization, W.S.; supervision, H.Z.; project administration, H.Z.; funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB2700300.

Data Availability Statement: <https://github.com/weihus/chainmaker-recorder> (accessed on 21 May 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Shalaby, S.; Abdellatif, A.A.; Al-Ali, A.; Mohamed, A.; Erbad, A.; Guizani, M. Performance evaluation of hyperledger fabric. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 608–613.
2. Dinh, T.T.A.; Wang, J.; Chen, G.; Liu, R.; Ooi, B.C.; Tan, K.L. Blockbench: A framework for analyzing private blockchains. In Proceedings of the 2017 ACM International Conference on Management of Data, Chicago, IL, USA, 14–19 May 2017; pp. 1085–1100.
3. Alom, I.; Ferdous, M.S.; Chowdhury, M.J.M. Blockmeter: An application agnostic performance measurement framework for private blockchain platforms. *IEEE Trans. Serv. Comput.* **2023**, *16*, 3879–3891. [[CrossRef](#)]
4. Fischl, W.; Gottlob, G.; Longo, D.M.; Pichler, R. Hyperbench: A benchmark and tool for hypergraphs and empirical findings. *J. Exp. Algorithmics* **2021**, *26*, 1–40. [[CrossRef](#)]
5. Cheng, Y.; Wei, K.; Zhang, Y.; Jiang, C.; Pang, W.; Zhang, Q.; Liu, B.; Zhang, L.; Liu, T.; Wu, Y. TrustedBench: An Efficient and User-friendly Distributed Performance Testing Tool for Blockchain System. In Proceedings of the 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Exeter, UK, 1–3 November 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 2141–2146.
6. Chainmaker. 2021. Available online: <https://chainmaker.org.cn/> (accessed on 27 January 2021).
7. Alaa, A.M.; van der Schaar, M. Bayesian nonparametric causal inference: Information rates and learning algorithms. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 1031–1046. [[CrossRef](#)]
8. Ahmed, S.S.; Roy, S.; Kalita, J. Assessing the effectiveness of causality inference methods for gene regulatory networks. *IEEE/Acm Trans. Comput. Biol. Bioinform.* **2018**, *17*, 56–70. [[CrossRef](#)]
9. Ebert-Uphoff, I.; Deng, Y. Causal discovery for climate research using graphical models. *J. Clim.* **2012**, *25*, 5648–5665. [[CrossRef](#)]
10. Ponnappalli, S.; Shah, A.; Banerjee, S.; Malkhi, D.; Tai, A.; Chidambaram, V.; Wei, M. {RainBlock}: Faster transaction processing in public blockchains. In Proceedings of the 2021 USENIX Annual Technical Conference (USENIX ATC 21), Online, 14–16 July 2021; pp. 333–347.
11. Schölkopf, B.; Locatello, F.; Bauer, S.; Ke, N.R.; Kalchbrenner, N.; Goyal, A.; Bengio, Y. Toward causal representation learning. *Proc. IEEE* **2021**, *109*, 612–634. [[CrossRef](#)]

12. Huang, B.; Zhang, K.; Zhang, J.; Ramsey, J.; Sanchez-Romero, R.; Glymour, C.; Schölkopf, B. Causal discovery from heterogeneous/nonstationary data. *J. Mach. Learn. Res.* **2020**, *21*, 1–53.
13. Cheng, Y.; Yang, R.; Xiao, T.; Li, Z.; Suo, J.; He, K.; Dai, Q. Cuts: Neural causal discovery from irregular time-series data. *arXiv* **2023**, arXiv:2302.07458.
14. Cheng, Y.; Li, L.; Xiao, T.; Li, Z.; Suo, J.; He, K.; Dai, Q. CUTS+: High-dimensional causal discovery from irregular time-series. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 11525–11533. [[CrossRef](#)]
15. Aslanidis, N.; Bariviera, A.F.; Martínez-Ibañez, O. An analysis of cryptocurrencies conditional cross correlations. *Financ. Res. Lett.* **2019**, *31*, 130–137. [[CrossRef](#)]
16. Kim, M.J.; Canh, N.P.; Park, S.Y. Causal relationship among cryptocurrencies: A conditional quantile approach. *Financ. Res. Lett.* **2021**, *42*, 101879.
17. Azqueta-Gavaldón, A. Causal inference between cryptocurrency narratives and prices: Evidence from a complex dynamic ecosystem. *Phys. Stat. Mech. Its Appl.* **2020**, *537*, 122574.
18. Yousaf, I.; Gubareva, M.; Teplova, T. Connectedness of non-fungible tokens and conventional cryptocurrencies with metals. *N. Am. J. Econ. Financ.* **2023**, *68*, 101995.
19. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
20. Li, H.; Chen, Y.; Shi, X.; Bai, X.; Mo, N.; Li, W.; Guo, R.; Wang, Z.; Sun, Y. FISCO-BCOS: An Enterprise-grade Permissioned Blockchain System with High-performance. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Denver, CO, USA, 12–17 November 2023; pp. 1–17.
21. Voronov, T.; Raz, D.; Rottenstreich, O. A framework for anomaly detection in blockchain networks with sketches. *IEEE/ACM Trans. Netw.* **2023**, *32*, 686–698.
22. Liu, C.; Xu, Y.; Sun, Z. Directed dynamic attribute graph anomaly detection based on evolved graph attention for blockchain. *Knowl. Inf. Syst.* **2024**, *66*, 989–1010.
23. Wen, Y.F.; Hsu, C.M. A performance evaluation of modular functions and state databases for Hyperledger Fabric blockchain systems. *J. Supercomput.* **2023**, *79*, 2654–2690.
24. Hao, Y.; Li, Y.; Dong, X.; Fang, L.; Chen, P. Performance analysis of consensus algorithm in private blockchain. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 280–285.
25. Alqahtani, S.; Demirbas, M. Bottlenecks in blockchain consensus protocols. In Proceedings of the 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), Barcelona, Spain, 23–25 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8.
26. Korkmaz, K.; Bruneau-Queyreix, J.; Mokhtar, S.B.; Réveillère, L. ALDER: Unlocking blockchain performance by multiplexing consensus protocols. In Proceedings of the 2022 IEEE 21st International Symposium on Network Computing and Applications (NCA), Boston, MA, USA, 14–16 December 2022; IEEE: Piscataway, NJ, USA, 2022; Volume 21, pp. 9–18.
27. Croman, K.; Decker, C.; Eyal, I.; Gencer, A.E.; Juels, A.; Kosba, A.; Miller, A.; Saxena, P.; Shi, E.; Gün Sirer, E.; et al. On Scaling Decentralized Blockchains: (A Position Paper). In Proceedings of the International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 22–26 February 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 106–125.
28. Wang, C.; Chu, X. Performance characterization and bottleneck analysis of hyperledger fabric. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November 2020–1 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1281–1286.
29. Thakkar, P.; Nathan, S.; Viswanathan, B. Performance benchmarking and optimizing hyperledger fabric blockchain platform. In Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Milwaukee, WI, USA, 25–28 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 264–276.
30. Xu, X.; Sun, G.; Luo, L.; Cao, H.; Yu, H.; Vasilakos, A.V. Latency performance modeling and analysis for hyperledger fabric blockchain network. *Inf. Process. Manag.* **2021**, *58*, 102436.
31. Toyoda, K.; Machi, K.; Ohtake, Y.; Zhang, A.N. Function-level bottleneck analysis of private proof-of-authority ethereum blockchain. *IEEE Access* **2020**, *8*, 141611–141621.
32. Schäffer, M.; Di Angelo, M.; Salzer, G. Performance and scalability of private Ethereum blockchains. In Proceedings of the Business Process Management: Blockchain and Central and Eastern Europe Forum: BPM 2019 Blockchain and CEE Forum, Vienna, Austria, 1–6 September 2019; Proceedings 17; Springer: Cham, Switzerland, 2019; pp. 103–118.
33. Hyperledger. Hyperledger Caliper. 2017. Available online: <https://github.com/hyperledger/caliper> (accessed on 20 March 2018).
34. Mazzoni, M.; Corradi, A.; Di Nicola, V. Performance evaluation of permissioned blockchains for financial applications: The ConsenSys Quorum case study. *Blockchain Res. Appl.* **2022**, *3*, 100026.
35. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; pp. 1–15.

36. Ampel, B.; Patton, M.; Chen, H. Performance modeling of hyperledger sawtooth blockchain. In Proceedings of the 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), Shenzhen, China, 1–3 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 59–61.
37. Alsahan, L.; Lasla, N.; Abdallah, M. Local bitcoin network simulator for performance evaluation using lightweight virtualization. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qata, 2–5 February 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 355–360.
38. Rubin, J. Btcspark: Scalable analysis of the bitcoin blockchain using spark. *Dec* **2015**, *16*, 1–14.
39. Saingre, D.; Ledoux, T.; Menaud, J.M. BCTMark: A framework for benchmarking blockchain technologies. In Proceedings of the 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA), Antalya, Turkey, 2–5 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
40. Zhang, J.; Gao, J.; Wang, K.; Wu, Z.; Lan, Y.; Guan, Z.; Chen, Z. TBFT: Understandable and efficient Byzantine fault tolerance using trusted execution environment. *arXiv* **2021**, arXiv:2102.01970.
41. Yuan, C.; Malone, B. Learning optimal Bayesian networks: A shortest path perspective. *J. Artif. Intell. Res.* **2013**, *48*, 23–65. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.