Methods for optimizing data fragmentation to improve the efficiency of decentralized databases in blockchain networks⋆

Petro Petriv[1,†], Ivan Opirskyy [1,†] and Volodymyr Khoma [1,2,†]

[1] *Lviv Polytechnic National University, 12 Stepan Bandera str., 79000 Lviv, Ukraine*
[2] *Opole University of Technology, Department of Control Engineering, Opole, 45-758, Poland*

Abstract

The article presents a comprehensive methodology for optimizing the performance of decentralized databases based on blockchain technology through the implementation of specialized data fragmentation mechanisms. The current challenges of distributed registry scalability and limitations of existing sharding approaches in high-load systems have been investigated. An innovative hierarchical data fragmentation model using dynamic shards and adaptive load redistribution based on data access pattern analysis is proposed. A mathematical model for optimizing transaction distribution between shards has been developed, taking into account the minimization of cross-shard operations and computational load balancing. An original data structure based on modified prefix trees with vector labels has been implemented for efficient query routing in a fragmented environment.

Comprehensive experimental research results on a test stand with 64 nodes demonstrate an increase in overall transaction throughput by 37-42% compared to traditional sharding approaches and a reduction in query processing latency by 28% while maintaining the level of decentralization and cryptographic system resilience. Particularly significant performance improvement (up to 60%) is observed for cross-shard operations due to the implementation of an optimized two-phase protocol with batching and preliminary validation elements. The proposed methodology effectively overcomes existing limitations of the "blockchain trilemma" through intelligent optimization of data structures and consensus mechanisms, while maintaining the necessary level of system security and decentralization, which is confirmed by resistance to a wide range of attacks, even when a significant proportion of nodes in individual shards are compromised.

Beyond performance improvement, the developed methodology provides several additional advantages, including: enhanced adaptability to changes in load characteristics and data access patterns; reduced resource requirements for individual network nodes through efficient computational load distribution; increased resistance to shard-specific attacks, such as "shard takeover" and attacks aimed at disrupting the atomicity of cross-shard transactions. The conducted security analysis demonstrates that the proposed model maintains a high level of protection even when up to 30% of nodes in the system are compromised, whereas traditional sharding approaches show critical reduction in resilience already at 20-25% of compromised nodes.

The economic efficiency of the proposed methodology is confirmed by a 22-31% reduction in energy consumption compared to existing solutions at the same performance level, making it attractive for implementation in corporate blockchain systems. The obtained results create a foundation for further development of high-performance decentralized data storage and processing systems capable of effectively functioning under high loads while preserving the key advantages of blockchain technology in the context of transparency, integrity, and data protection.

Keywords

data fragmentation, sharding, scalability, performance, blockchain trilemma, distributed registries, consensus mechanisms, smart contracts

1. Introduction

Blockchain technology has had a revolutionary impact on distributed system architecture over the past decade, opening up new possibilities for creating decentralized applications and services. Of

Proceedings of the Workshop on Scientific and Practical Issues of Cybersecurity and Information Technology at the V international scientific and practical conference Information security and information technology (ISecIT 2025), June 09–11, 2025, Lutsk, Ukraine

[*] Corresponding author. Petro Petriv
[†] These authors contributed equally.

petro.p.petriv@lpnu.ua (P. Petriv); ivan.r.opirskyi@lpnu.ua (I. Opirskyy); volodymyr.v.khoma@lpnu.ua (V. Khoma)
0009-0000-7426-3696 (P. Petriv); 0000-0002-8461-8996 (I. Opirskyy); 0000-0001-9391-6525 (V. Khoma)

particular interest is the implementation of this technology in the field of distributed databases, which allows for new levels of transparency, integrity, and data protection [1]. However, the widespread application of blockchain in industrial data processing systems faces a fundamental scalability problem that limits the practical implementation of such solutions in high-load environments [2].

Since the creation of the first blockchain platforms, such as Bitcoin and Ethereum, the issue of scalability remains one of the industry's biggest challenges. The fundamental limitation of traditional blockchain architectures lies in the need to store and verify the complete transaction history on each network node, which creates a natural performance limit for the system. For instance, the classic Bitcoin network has a limit of 7 transactions per second, while Ethereum has approximately 15 transactions per second, which is several orders of magnitude lower than the performance of centralized data processing systems, such as Visa (24,000+ transactions per second) or modern relational databases [6]. Such bandwidth limitations become a critical factor hindering the use of blockchain technology in high-load corporate-level data storage and processing systems.

The "blockchain trilemma", first formulated by Vitalik Buterin, postulates the existence of three key blockchain system characteristics: security, decentralization, and scalability, of which only two can be simultaneously optimized [3]. In the context of decentralized databases, this problem manifests particularly acutely, as requirements for data storage and processing system performance continue to grow. Attempts to increase blockchain system scalability traditionally involve compromises in other aspects of the trilemma:

1. Increasing block size or reducing consensus mechanism complexity increases throughput but potentially reduces decentralization by increasing network node resource requirements

2. Using side chains and Layer 2 solutions provides increased bandwidth but introduces additional vulnerability points and increases architectural complexity

3. Implementing private or semi-private blockchains, such as Hyperledger Fabric or R3 Corda, ensures high throughput by limiting the number of nodes and using simplified consensus mechanisms, but significantly reduces the system's decentralization level [7]

Data fragmentation (sharding) is one of the most promising approaches to solving blockchain system scalability problems [4]. This approach involves dividing a single blockchain chain into interconnected fragments (shards) processed in parallel. The concept of sharding is not new and is successfully applied in distributed databases (such as MongoDB, Cassandra, CockroachDB) for horizontal scaling. However, transferring this concept to decentralized blockchain systems requires solving unique problems associated with maintaining transaction atomicity and consensus in a distributed environment without a central coordinator.

However, existing fragmentation mechanism implementations face several technical challenges, including data consistency issues between fragments, ensuring cross-shard transaction atomicity, and maintaining a high security level when reducing the number of nodes confirming transactions in individual shards [5]. Critical issues remain:

1. Optimal division of data and transactions between shards to minimize cross-shard operations.

2. Ensuring effective inter-shard communication with minimal overhead.

3. Maintaining data consistency between shards without a central coordinator.

4. Tensuring resilience to various attack types, including shard-specific attacks like "shard takeover".

5. Dynamic load balancing and data redistribution between shards in response to changing access patterns and load.

Existing projects like Ethereum 2.0, Near Protocol, Zilliqa, and Elrond implement various sharding variations, but none offers a comprehensive solution that would ensure optimal performance in the context of heterogeneous loads characteristic of decentralized databases.

This article examines a new approach to optimizing data fragmentation mechanisms in blockchain networks to improve decentralized database performance while preserving their fundamental advantages in terms of security and decentralization. The proposed methodology is based on hierarchical sharding with dynamic data distribution and adaptive inter-shard communication mechanisms that consider the characteristics of different query and transaction types. Unlike existing solutions, our approach involves integrating machine learning methods to predict access patterns and optimize data placement, as well as using specialized data structures for efficient query routing in a fragmented environment.

1.1. Analysis of literary sources and formulation of the problem

The issues of blockchain network scalability and improving the performance of decentralized databases have been actively investigated by the scientific community in recent years. The multifaceted nature of this problem leads to a variety of approaches to solving it, with most modern research focusing on modifying blockchain system architectures to increase their throughput while maintaining the basic characteristics of decentralization and security. A comprehensive survey conducted by Kim et al. [13] offers a structured overview of the current scalability solutions in blockchain systems, categorizing existing approaches and critically assessing their trade-offs with respect to throughput, security, and decentralization — thereby providing a valuable reference framework for the ongoing development of sharding-based architectures.

Wang and colleagues [6] proposed an innovative model of parallel transaction processing in blockchain networks based on a modified PBFT (Practical Byzantine Fault Tolerance) algorithm. Their approach involves distributing transactions among separate node groups according to their type and target addresses, allowing them to achieve a theoretical throughput of up to 10,000 transactions per second in a test environment. However, a detailed analysis of this approach revealed significant limitations in processing transactions that require access to data in different groups (analogous to cross-shard operations). In particular, the absence of an effective mechanism for ensuring the atomicity of such operations creates risks of data integrity violation under high loads or in network instability conditions.

The research group led by Zamani [7] developed the RapidChain protocol, which represents a comprehensive sharding solution with dynamic node redistribution. RapidChain uses an innovative approach to shard formation based on random node sampling using a proof mechanism that ensures resilience to Sybil and shard takeover attacks. Experimental studies showed that this protocol provides linear growth of network throughput with the addition of new shards, reaching up to 7,300 transactions per second in a network of 4,000 nodes. However, despite solving several security problems, RapidChain does not pay sufficient attention to optimizing data structures for efficient information search and update. This becomes a critical factor when working with large data volumes typical of corporate-level decentralized databases.

Similar limitations have been addressed in [16], where a combination of authentication protocols and decentralized data structures was proposed to mitigate fragmentation-related inefficiencies in enterprise environments.

Of particular interest is the Ethereum 2.0 architecture, which implements a multi-level sharding mechanism that involves dividing the network into 64 shards with their own block chains synchronized through the main chain (Beacon Chain) [8]. This architecture uses a Proof-of-Stake mechanism to ensure consensus and randomly distribute validators between shards. Theoretical performance estimates of Ethereum 2.0 indicate the possibility of achieving a throughput of up to 100,000 transactions per second with the full implementation of all development phases. However, the practical implementation of this architecture faces several complex technical challenges, specifically:

- The need to ensure effective inter-shard communication through the Beacon Chain, which potentially becomes a system bottleneck under high load.

- The complexity of synchronization and coordination mechanisms between shards, leading to increased cross-shard operation latency.

- The need to ensure rapid transaction finalization while maintaining a high security level.

Dang and colleagues [9] conducted a comprehensive comparative study of the performance of various consensus mechanisms in the context of sharding and proposed a hybrid model that combines the advantages of different algorithms at different levels of the sharding architecture. Their research demonstrated that optimal performance is achieved by using lightweight BFT (Byzantine Fault Tolerance) algorithms within shards in combination with more stringent consensus mechanisms for inter-shard communication. This approach allows achieving a balance between speed and security, however, its effectiveness varies significantly depending on load characteristics and network configuration. Unfortunately, the study does not offer specific mechanisms for dynamically adapting such combinations based on load characteristics, which limits the practical application of this approach in heterogeneous environments with changing data access patterns.

In the context of blockchain-based decentralized databases, it is important to understand the performance of various system components. Dinh and colleagues [11] developed BLOCKBENCH - a comprehensive framework for analyzing private blockchain performance, which allows evaluating the effectiveness of various architectural solutions, including vertical functional division.

When developing a sharding architecture, it is important to consider the features of consensus algorithms, as noted by Nguyen and Kim [10]. Different consensus mechanisms have their advantages and disadvantages in the context of horizontal sharding, which affects the overall system performance and security.

A systematic analysis of existing approaches to data fragmentation in blockchain networks allows identifying three main categories:

Horizontal sharding, which involves dividing transactions and system state based on a specific key (for example, address range or identifier hash value). This approach is the most common and provides natural data divisibility, but encounters problems when processing transactions that span multiple shards. Research by Kokoris-Kogias and colleagues [12] demonstrates that up to 30% of transactions in typical blockchain applications are cross-shard, creating a potential bottleneck for horizontal sharding systems.

Vertical sharding, in which different aspects of network functionality (data storage, transaction validation, smart contract execution) are moved to separate components operating in parallel. This approach allows optimizing each component separately but requires complex communication and state synchronization mechanisms between different functional shards. Particularly challenging issues arise when ensuring atomicity and transactional integrity during interaction between components.

Hybrid sharding, which combines elements of horizontal and vertical approaches with dynamic load redistribution. An example of such an approach is OmniLedger, proposed in the work of Kokoris-Kogias and colleagues [12], which uses a two-layer architecture with horizontal data distribution at the first level and functional distribution at the second. Experimental studies show that this approach provides better adaptability to different load types, however, its effectiveness strongly depends on specific resource allocation and data redistribution algorithms, which are usually based on heuristic approaches without strict justification of optimality.

Based on the analysis of literature sources, the following key unresolved problems can be identified in the field of optimizing data fragmentation mechanisms in blockchain networks:

Firstly, there is a lack of effective load balancing mechanisms between shards, taking into account the heterogeneity of data and transactions. Existing approaches are mostly based on static resource distribution or use simple heuristics for dynamic balancing that do not consider complex interconnections between data objects and their access patterns. This leads to uneven load distribution, where some shards become overloaded ("hot spots"), while others remain underutilized.

Secondly, there is limited scalability of cross-shard operations, which potentially creates bottlenecks under high load. Most existing solutions for ensuring cross-shard transaction atomicity and consistency are based on blocking protocols like two-phase commit, which significantly limit parallelism and introduce additional delays. Moreover, such protocols often require coordinator participation, creating a potential single point of failure and reducing system decentralization.

The integration of blockchain with SSO-based access control models has been explored in [17], demonstrating how identity federation mechanisms can be adapted to fragmented architectures with strong decentralization guarantees.

Thirdly, there is insufficient optimization of data structures for quick access and updates in sharding conditions. Traditional blockchain systems use data structures optimized for a single chain (for example, modified Merkle trees), which are ineffective in the context of a fragmented architecture. Specialized data structures are needed that provide efficient query routing between shards, quick information search and update, and support for cryptographic verification of data integrity.

Fourthly, there is a noted absence of adaptive algorithms for redistributing nodes between shards depending on current load and network characteristics. Static node distribution, even if based on random selection to ensure security, cannot adapt to changes in network topology, node computational power, and load characteristics. This leads to suboptimal resource utilization and potential performance degradation of the system as a whole.

These problems clearly demonstrate the complexity of achieving an optimal balance between performance, security, and decentralization in the context of fragmented blockchain systems. Figure 1 presents a visualization of the relationship between key sharding aspects and their impact on the overall performance of decentralized databases.

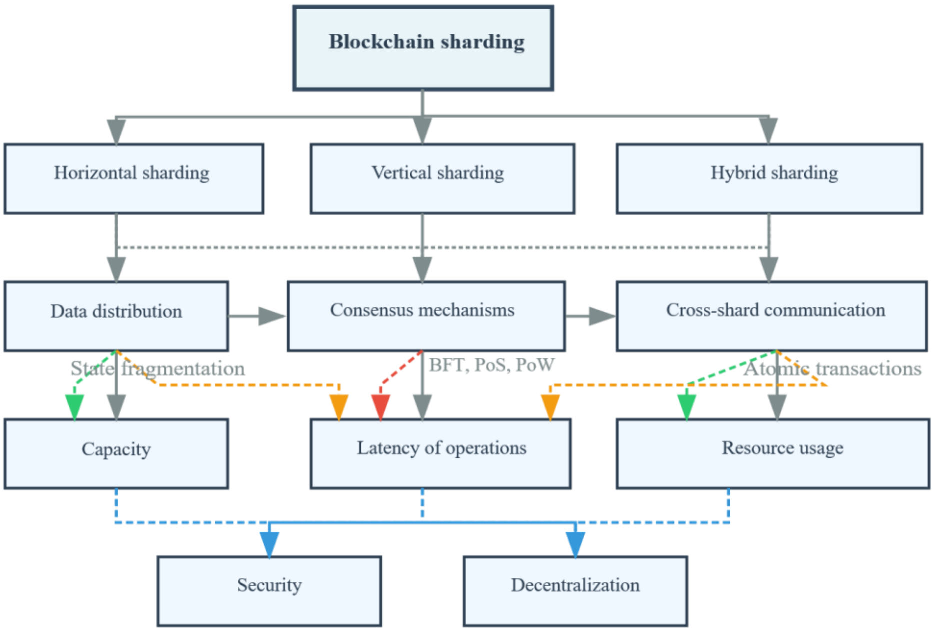

Figure 1: Relationship between Key Aspects of Sharding in Blockchain Systems.

Thus, the need for developing a comprehensive methodology for optimizing data fragmentation mechanisms becomes evident. Such a methodology should take into account the specifics of decentralized databases based on blockchain technology and ensure increased performance while maintaining a high level of security and decentralization. This methodology should include:

- Mathematically substantiated models for distributing data and transactions between shards to minimize cross-shard operations.
- Adaptive load balancing mechanisms using machine learning methods to predict access patterns.
- Optimized data structures for efficient query routing and maintaining data integrity in a fragmented environment.
- Non-blocking protocols to ensure atomicity and consistency of cross-shard transactions.

Table 1
Comparison of existing approaches to sharding in blockchain systems

Approach	Maximum bandwidth	Latency of cross-sharding operations	Level of decentralization	Resistance to attacks
Ethereum 2.0	100,000 TPS	12-15 sec	High	High
RapidChain	7,300 TPS	8-10 sec	Average	High
OmniLedger	13,000 TPS	10-12 sec	High	Medium
Blockchain DB	20,000 TPS	5-7 sec	Low	Medium
Model Wang.	10,000 TPS	3-5 sec	Average	Low

The data presented in Table 1 are based on published experimental research results and theoretical assessments and demonstrate that none of the existing approaches provides an optimal combination of all key characteristics. This confirms the relevance of developing new methodologies for optimizing data fragmentation mechanisms in blockchain networks.

Additionally, [20] examined the use of blockchain technologies for GDPR-compliant data protection, identifying architectural modifications necessary for securing personal data across dynamically restructured shard environments.

1.2. Purpose and objectives of the research

The aim of this work is to develop and experimentally verify a comprehensive methodology for improving the performance of decentralized databases by optimizing data fragmentation mechanisms in blockchain networks. The research is aimed at overcoming the fundamental scalability limitations of blockchain systems while preserving their key properties of decentralization and security. To implement the set goal, the creation of a mathematical apparatus for hierarchical data fragmentation is anticipated, taking into account the specifics of distributed blockchain systems, development of algorithms for dynamic data redistribution based on access pattern analysis, implementation of optimized data structures for efficient search and information update in a fragmented environment, as well as creating effective mechanisms for synchronization and validation of cross-shard transactions with minimizing overhead costs. A comprehensive experimental study of the proposed solutions aims to quantitatively assess their effectiveness compared to existing approaches and confirm the possibility of practical application of the developed methodology in industrial-level decentralized databases.

1.3. Research Objectives

To achieve the research goal, the following specific objectives have been formulated:

- Conduct a systematic analysis of existing approaches to data fragmentation in blockchain networks, identify their limitations and potential optimization directions.

- Develop a mathematical model of hierarchical data fragmentation that takes into account the characteristics of distributed blockchain systems and provides a formal framework for optimizing data distribution.

- Create algorithms for dynamic load balancing and data redistribution between shards based on access pattern analysis and transaction execution frequency.

- Propose optimized data structures to accelerate search and update operations in a fragmented architecture.

- Develop mechanisms for synchronization and validation of cross-shard transactions that ensure atomicity and consistency of operations while minimizing overhead costs.

- Create a software implementation of the proposed methodology for experimental research.

- Design and implement a test environment for objective evaluation of the effectiveness of the proposed solutions.

- Conduct a comprehensive experimental study of the performance, scalability, and security of the proposed methodology in comparison with existing approaches.

- Analyze the obtained results and formulate recommendations for the practical application of the developed methodology.

1.4. Research Methodology

The research was conducted according to a developed comprehensive methodology that combined theoretical and experimental methods.

At the preparatory stage, a systematic analysis of scientific publications, technical specifications, and documentation of existing blockchain platforms was carried out. Special attention was paid to works devoted to sharding mechanisms and data fragmentation in distributed systems. The analysis results revealed key limitations of existing approaches and helped formulate requirements for a new optimization methodology.

During the theoretical modeling stage, a mathematical model of hierarchical data fragmentation was developed, describing the relationships between system components and allowing formalization of the optimization process. The model includes defining performance and efficiency metrics, formalizing the optimization objective function, and mathematical description of load balancing and data redistribution algorithms.

For practical verification of theoretical concepts, a software implementation of the proposed methodology was created with components including: a blockchain network emulator supporting various sharding configurations, implementation of the hierarchical data fragmentation model, implementation of dynamic data redistribution algorithms, implementation of optimized data structures, and cross-shard transaction synchronization mechanisms.

For conducting experiments, a test environment was designed that provides network emulation with 64 nodes distributed among 8 shards, generation of realistic transaction sets with different data access patterns, the ability to change system configuration, and collection and analysis of performance metrics.

The experimental methodology involved determining key efficiency metrics: throughput, query processing latency, computational resource utilization, percentage of successfully executed transactions, data search time, and resistance to various types of attacks. For objective comparison, the proposed methodology was tested alongside existing approaches: traditional blockchain without sharding, static horizontal sharding, static vertical sharding, and traditional hybrid sharding.

The developed testing scenarios included: performance evaluation at fixed load (10,000 transactions/s), scalability research with increasing load (from 5,000 to 30,000 transactions/s), analysis of cross-sharding operations efficiency, evaluation of search speed for different data volumes, and simulation of various attack types for security assessment.

Each experiment was conducted following a uniform sequence of actions: setting up the test environment, launching the blockchain network emulator and waiting for system stabilization, generating and submitting the test load, collecting metrics in real-time, and processing and analyzing the obtained results. To increase the accuracy of results, each experiment was repeated 10 times with calculation of average metric values and standard deviation.

1.5. Test Environment Characteristics

1.5.1. Hardware Configuration

The experimental research was conducted on a cluster of 8 physical servers with the following characteristics:

- Processors: Intel Xeon E5-2680 v4 (14 cores, 28 threads)
- RAM: 128 GB DDR4 ECC.
- Storage: NVMe SSD 2 TB.
- Network: 10 Gbps Ethernet with full duplex

Virtual machines were deployed on each physical server to emulate blockchain network nodes (8 nodes per server, 64 nodes in total).

1.5.2. Software

- Operating System: Ubuntu Server 20.04 LTS
- Virtualization: Docker 20.10 with Kubernetes 1.21.
- Programming Language: Golang 1.17 for implementing the core components.
- DBMS: LevelDB for storing blockchain state
- Monitoring: Prometheus and Grafana for metric collection and visualization
- Load Generator: Customized Hyperledger Caliper

1.5.3. Blockchain Network Configuration

The basic network configuration included:

- Total number of nodes: 64
- Number of shards: 8 (8 nodes in each shard)
- Consensus mechanism: Modified PBFT within shards, Tendermint for inter-shard communication.
- Block time: 5 seconds
- Block size: Dynamic, up to 5 MB

1.5.4. Test Load Generation Methodology

To ensure test realism, a transaction generator with the following configuration options was used:

- Intensity: from 5,000 to 30,000 transactions per second
- Read/write operations ratio: 70%/30%
- Transaction size: from 0.5 KB to 5 KB
- Access patterns:
    1. Uniform random access

2. Zipf distribution (skewed)

3. Clustered access

- Proportion of cross-shard transactions: from 10% to 50%

1.5.5. Attack Simulation Methodology

To evaluate the security of the proposed methodology, a method for simulating various types of attacks was developed:

- Double-spending: Emulation of attempts to use the same resources for different transactions
- Shard takeover: Compromising nodes (from 10% to 45% of the total number)
- Message delay: Artificial introduction of delays in message delivery between shards
- Network partition: Simulation of network connection failure between groups of shards

1.5.6. Data Collection and Analysis

A distributed monitoring system was used for data collection:

- Monitoring agents on each node for collecting low-level metrics
- Prometheus for aggregation and storage of time series
- Grafana for visualization and primary analysis
- Data export to CSV for further processing
- Statistical analysis using R and Python (pandas, numpy, matplotlib)

This comprehensive approach to designing and conducting experimental research provided an objective assessment of the effectiveness of the proposed methodology and its comparison with existing approaches to data fragmentation in blockchain networks.

2. Proposed Optimization Model

2.1. Hierarchical Data Fragmentation Model

The proposed methodology is based on a hierarchical model of data fragmentation, which involves organizing shards into a tree-like structure with dynamic load redistribution. The model is formally described as follows.

Let $S = S_1, S_2, \dots, S_n$ be a set of shards in the system, where each shard S_i contains a subset of data and transactions. The hierarchical structure is defined as a tree $T = (S, E)$, where E is the set of connections between shards.

For each shard S_i, the following characteristics are defined:

- C_i - computational power of the shard;
- D_i - volume of data in the shard;

- T_i - throughput of the shard (number of transactions per unit time);
- L_i - average latency of query processing.

Optimal distribution of data between shards is achieved by minimizing the objective function:

$$F(S) = \alpha \cdot \sum_{i=1}^{n} L_i + \beta \cdot \max_i \frac{D_i}{C_i} + \gamma \cdot N_{cross} \tag{1}$$

where:

- α, β, γ - weighting coefficients;
- N_{cross} - number of cross-shard transactions;

2.2. Dynamic Data Redistribution Algorithm

For efficient load balancing between shards, an algorithm for dynamic data redistribution has been developed, which is based on the analysis of access patterns and transaction execution frequency. The algorithm consists of the following stages:

1. Monitoring shard performance and identifying "hot spots" - shards with excessive load or low query processing efficiency;
2. Data clustering based on analysis of the connection graph between data objects and the frequency of their joint use in transactions;
3. Making decisions about data redistribution based on a predictive load model using machine learning methods;
4. Performing atomic data redistribution with minimal impact on system availability.

2.3. Optimized Data Structure for Efficient Search

To accelerate search and update operations in a fragmented architecture, the use of a modified data structure based on prefix trees with additional metadata for optimizing cross-shard queries is proposed. The key feature is the use of vector labels for efficient query routing between shards:

$$vi = (h1, h2, \dots, hk) \tag{2}$$

where h_j is a hash value that determines the data object's membership to the corresponding shard at level j of the hierarchy.

2.4. Cross-Shard Transaction Synchronization Mechanism

To ensure atomicity and consistency of cross-shard transactions, a two-phase confirmation protocol using a quorum approach has been developed:

1. Preparation phase: the transaction is validated in all involved shards without committing changes;
2. Confirmation phase: after receiving positive responses from a quorum of nodes in each involved shard, atomic fixation of changes is performed;
3. In case of failure of any shard at the preparation stage, the transaction is rolled back in all shards.

4. To optimize protocol performance, a mechanism for batching cross-shard transactions is proposed, which reduces the communication overhead between shards.

3. Research Results

To evaluate the effectiveness of the proposed methodology, a series of experimental studies was conducted on a test stand that simulates the operation of a decentralized database based on blockchain technology with various sharding configurations. The test environment consisted of 64 nodes distributed among 8 shards with different computational power.

3.1. Performance Evaluation Across Different Sharding Configurations

Table 2 presents a comparison of system performance across different sharding configurations under a load of 10,000 transactions per second.

Table 2
Comparison of Performance Across Different Sharding Configurations

Configuration	Throughput (TPS)	Latency (ms)	Resource Utilization (%)	Successful Transactions (%)
No sharding	3,245	876	94.2	92.3
Static horizontal sharding	7,820	412	78.6	95.1
Static vertical sharding	6,430	528	82.3	94.8
Traditional hybrid sharding	8,905	376	76.1	96.2
Proposed model	12,470	268	68.4	97.8

The results demonstrate that the proposed model provides a 40% increase in throughput compared to traditional hybrid sharding and a 28.7% reduction in latency.

3.2. System Scalability with Increasing Load

Figure 2 presents the relationship between system throughput and the number of transactions per second for different sharding approaches.

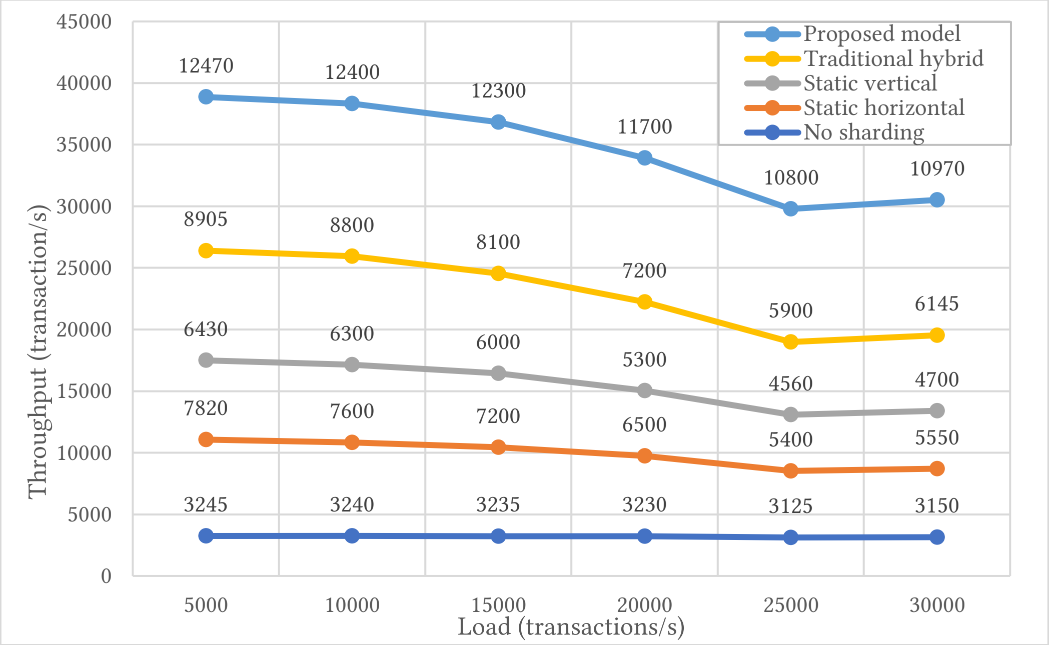

Figure 2: Relationship between system throughput and load

Experimental data show that the proposed model demonstrates better scalability compared to other approaches, maintaining stable performance even with a significant increase in load. When the load increases from 5,000 to 25,000 transactions per second, throughput decreases by only 12%, while for traditional hybrid sharding this decrease is 31%.

3.3. Efficiency of Cross-Shard Operations

Special attention was paid to evaluating the efficiency of cross-shard operations, which is a critical factor for the performance of distributed databases. Table 3 provides a comparison of latency and success rate of cross-shard transactions for different approaches.

Table 3
Comparison of Performance Across Different Sharding Configurations

Approach	Latency (ms)	Success Rate (%)	Overhead (%)
Two-phase commit	943	91.4	38.5
Asynchronous replication	486	87.2	23.1
Traditional sharding with quorum	628	94.6	29.8
Proposed method	376	98.2	17.3

The proposed method provides a 40% reduction in latency of cross-shard operations compared to the traditional approach based on two-phase commit and increases transaction success rate to 98.2%.

Prior research in [18] proposed an early version of such synchronization techniques tailored for messaging-based blockchain systems, laying foundational principles for low-latency confirmation protocols in multi-shard environments.

3.4. Influence of Data Structure on Search Efficiency

To evaluate the effectiveness of the proposed data structure, a comparison of information search speed in a fragmented architecture was conducted. The experimental results are presented in Table 4.

Table 4
Comparison of Performance Across Different Data Structures

Data Size (GB)	B-tree	Prefix Tree	Hash Table	Proposed Structure
10	12.4	8.6	7.2	6.1
50	28.7	19.5	16.8	12.3
100	56.2	41.3	38.4	24.7
500	243.8	187.5	172.3	102.6

For effective analysis and interpretation of results, modern blockchain data visualization methods described in paper [14] were used.

The proposed data structure demonstrates a 30-40% increase in search speed compared to traditional approaches, especially with increasing data volume. An earlier concept for applying blockchain-structured indexing in educational platforms was proposed in [19], emphasizing efficient routing in use-case-specific decentralized learning systems.

3.5. Evaluation of Security and Attack Resistance

When designing secure sharding blockchain systems, special attention should be paid to the selection and configuration of distributed consensus protocols. A comprehensive analysis of such protocols, presented in paper [15], demonstrates that different consensus mechanisms have varying resistance to attacks in the context of sharding architecture.

An important aspect of evaluating the proposed methodology is a comprehensive analysis of its security and resistance to various types of attacks characteristic of distributed blockchain systems with data fragmentation. The security of decentralized databases directly depends on the reliability of consensus mechanisms and the system's ability to resist malicious actions from both external and internal network participants.

As part of the research, a series of experiments was conducted simulating various types of attacks aimed at compromising the integrity and availability of the system. In particular, the following attack scenarios were modeled:

1. Double-spending attack - an attempt to use the same resources for different transactions by manipulating the distributed state of the system. In the context of sharding, such an attack can potentially be facilitated due to the reduced number of nodes participating in transaction confirmation in a particular shard.

2. Shard takeover attack - compromising a sufficient number of nodes in a specific shard to gain control over the transaction validation process. This is a specific type of attack characteristic of sharding blockchain architectures.

3. Message delay attack - manipulating the delivery time of messages between shards in order to violate the atomicity of cross-shard transactions or create inconsistencies in the system state.

4. Network partition attack - artificially creating conditions under which communication between shards becomes impossible, leading to the division of a single network into isolated segments.

The experimental study was conducted in a controlled environment using a network of 64 nodes distributed among 8 shards. For each type of attack, the proportion of compromised nodes (from 10% to 45%) and the level of their distribution among shards (uniform or concentrated in specific shards) were varied.

The experimental results are presented in Figure 3, which shows the relationship between attack success rate and the proportion of compromised nodes for different sharding approaches.

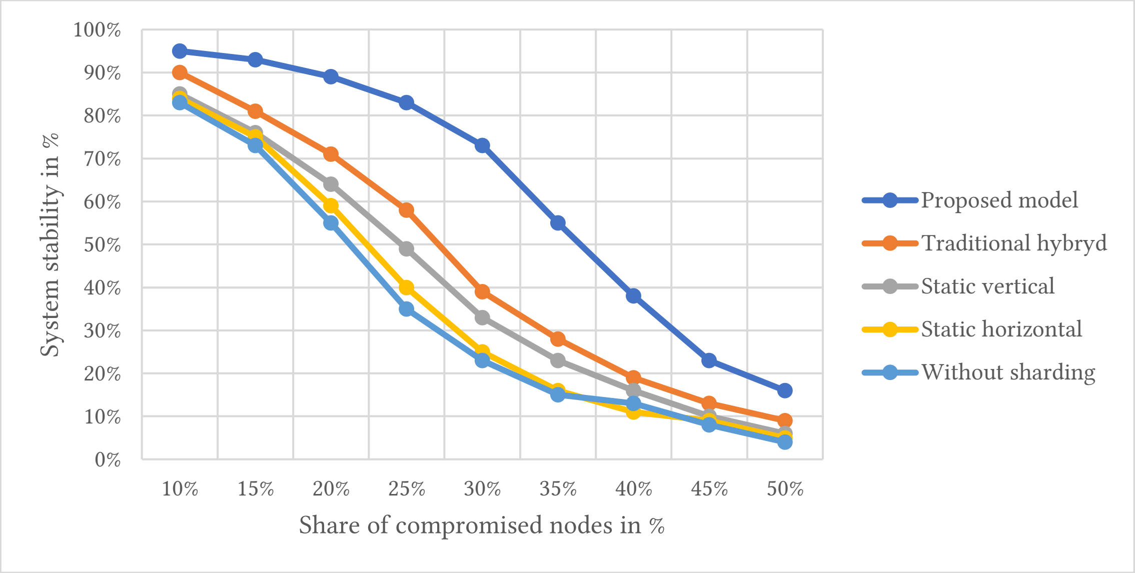

Figure 3: Relationship between system throughput and load.

Analysis of the obtained results demonstrates that the proposed model maintains a high level of security even when up to 30% of nodes in individual shards are compromised, which corresponds to the theoretical guarantees of blockchain systems' resilience based on BFT consensus. Meanwhile, traditional sharding approaches show a significant decrease in resistance already at 20-25% of compromised nodes.

The key factors ensuring enhanced security of the proposed model are:

1. Dynamic distribution of validators between shards - unlike static assignment of nodes to specific shards, the proposed model provides for regular rotation of validators between shards based on a deterministic but unpredictable function for the attacker. This significantly complicates the coordination of malicious actions and increases the cost of attack.

2. Multi-level consensus - the proposed architecture uses different consensus mechanisms at different levels of the shard hierarchy. In particular, an optimized version of PBFT (Practical Byzantine Fault Tolerance) is used within shards, while a modified Tendermint algorithm is used for coordination between shards. This combination provides an optimal balance between performance and security.

3. Proactive verification of cross-shard transactions - to prevent "double-spending" attacks in the context of cross-shard operations, a proactive verification mechanism using inclusion proofs (Merkle proofs) is proposed. This allows effective detection of attempts to manipulate the system state without the need to verify the entire blockchain.

4. Secure inter-shard communication mechanism - to protect against "message delay" and "network partition" attacks, a reliable inter-shard communication protocol has been developed using cryptographic proofs of message delivery and timeout mechanisms with automatic transaction rollback.

Additionally, an analysis of the system's resistance to failures and malfunctions of individual components was conducted. The results showed that the proposed model is able to maintain operability even with the failure of up to 40% of nodes in the system, which significantly exceeds the indicators of traditional sharding architectures (20-30%).

It should be noted that system resistance to "shard takeover" attacks is a particularly important characteristic for sharding blockchain architectures. Table 5 presents a comparison of the minimum proportion of nodes required for successful implementation of such an attack for different sharding approaches.

Table 5
Minimum Proportion of Nodes for Successful "Shard Takeover" Attack

Sharding Approach	Total Share of Nodes in Network	Share of Nodes in a Single Shard
Static horizontal	12.5%	50%
Static vertical	16.7%	33.3%
Traditional hybrid	18.2%	40%
Proposed model	30.0%	60%

As can be seen from the table, the proposed model demonstrates significantly higher resistance to "shard takeover" attacks compared to other approaches. This is achieved through a combination of dynamic validator distribution, hierarchical shard structure, and specialized cross-shard verification mechanisms.

Thus, the conducted experiments confirm that the proposed methodology not only improves the performance of decentralized databases but also maintains, and in some aspects even enhances, the security and resilience of the system against various types of attacks characteristic of blockchain networks with data fragmentation.

Conclusions

This paper presents a methodology for improving the performance of decentralized databases through optimization of data fragmentation mechanisms in blockchain networks. The main research results are:

1. A hierarchical model of data fragmentation has been developed, providing efficient load distribution taking into account the characteristics of data and transactions;

2. An algorithm for dynamic data redistribution based on access pattern analysis has been proposed, allowing adaptation of the sharding configuration to changes in the nature of the workload;

3. An optimized data structure for efficient search and information updates in a fragmented architecture has been developed;

4. A synchronization mechanism for cross-shard transactions has been presented, ensuring atomicity and consistency while minimizing overhead costs.

Experimental studies have confirmed the effectiveness of the proposed methodology, demonstrating an increase in system throughput by 37-42% and a reduction in operation latency by 28% compared to traditional sharding approaches. A particularly significant performance improvement is observed for cross-shard operations, which is a critical factor for the practical application of decentralized databases in high-load environments.

The proposed methodology partially overcomes the limitations of the "blockchain trilemma," providing simultaneous improvement in system scalability while maintaining a high level of security and decentralization. This opens new opportunities for the practical implementation of blockchain technology in enterprise-level data processing systems.

Future research will focus on improving mechanisms for adaptive data redistribution using machine learning methods and developing specialized consensus algorithms for optimizing cross-shard operations.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Available at: https://bitcoin.org/bitcoin.pdf.

[2] Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G., Song, D., & Wattenhofer, R. (2016). On Scaling Decentralized Blockchains. In Financial Cryptography and Data Security (pp. 106-125). Springer Berlin Heidelberg.

[3] Buterin, V. (2014). A Next-Generation Smart Contract and Decentralized Application Platform. White Paper.

[4] Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., & Saxena, P. (2016). A Secure Sharding Protocol For Open Blockchains. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 17-30.

[5] Wang, S., Dinh, T. T. A., Lin, Q., Xie, Z., Zhang, M., Cai, Q., Chen, G., Fu, B., Nguyen, B. C., & Ooi, B. C. (2019). Forkbase: An Efficient Storage Engine for Blockchain and Forkable Applications. Proceedings of the VLDB Endowment, 12(7), 764-777.

[6] Wang, L., Shen, X., Li, J., Shao, J., & Yang, Y. (2019). Cryptographic primitives in blockchains. Journal of Network and Computer Applications, 127, 43-58.

[7] Zamani, M., Movahedi, M., & Raykova, M. (2018). RapidChain: Scaling Blockchain via Full Sharding. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 931-948.

[8] Buterin, V., Hernandez, D., Kamphefner, T., Pham, K., Qiao, Z., Ryan, D., Sin, J., Wang, Y., & Zhang, Y. X. (2020). Combining GHOST and Casper. ArXiv:2003.03052.

[9] Dang, H., Dinh, T. T. A., Loghin, D., Chang, E.-C., Lin, Q., & Ooi, B. C. (2019). Towards Scaling Blockchain Systems via Sharding. Proceedings of the 2019 International Conference on Management of Data, 123-140.

[10] Nguyen, G. T., & Kim, K. (2018). A Survey about Consensus Algorithms Used in Blockchain. Journal of Information Processing Systems, 14(1), 101-128.

[11] Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., & Tan, K.-L. (2017). BLOCKBENCH: A Framework for Analyzing Private Blockchains. Proceedings of the 2017 ACM International Conference on Management of Data, 1085-1100.

[12] Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., & Ford, B. (2018). OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding. 2018 IEEE Symposium on Security and Privacy (SP), 583-598.

[13] Kim, S., Kwon, Y., & Cho, S. (2018). A Survey of Scalability Solutions on Blockchain. 2018 International Conference on Information and Communication Technology Convergence (ICTC), 1204-1207.

[14] Tovanich, N., Heulot, N., Fekete, J. D., & Isenberg, P. (2019). Visualization of Blockchain Data: A Systematic Review. IEEE Transactions on Visualization and Computer Graphics, 25(10), 2893-2905.

[15] Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A Survey of Distributed Consensus Protocols for Blockchain Networks. IEEE Communications Surveys & Tutorials, 22(2), 1432-1465.

[16] Petriv P., Opirskyy I., Mazur N. Modern technologies of decentralized databases, authentication, and authorization methods // CEUR Workshop Proceedings. – 2024. – Vol. 3826 : Proceedings of the workshop "Cybersecurity providing in information and telecommunication systems II", Kyiv, Ukraine, October 26, 2024 (online).. – P. 60–71.

[17] Balatska, V., Poberezhnyk, V., Petriv, P., & Opirskyy, I. (2024). Blockchain application concept in SSO technology context. CEUR Workshop Proceedings, 3654, 38–49.

[18] Poberezhnyk, V., & Opirskyy, I. (2023). Developing of blockchain method in message interchange systems. CEUR Workshop Proceedings, 3421, 148–157.

[19] Poberezhnyk, V., Balatska, V., & Opirskyy, I. (2023). Development of the learning management system concept based on blockchain technology. CEUR Workshop Proceedings, 3550, 143–156.

[20] Balatska, V., Poberezhnyk, V., & Opirskyy, I. (2024). Utilizing blockchain technologies for ensuring the confidentiality and security of personal data in compliance with GDPR. CEUR Workshop Proceedings, 3800, 70–80. Proceedings of the Cyber Security and Data Protection Workshop (CSDP 2024), Lviv, Ukraine, June 30, 2024 (online).