ShardingSim: A Modular Committee-Based Sharding Blockchain Simulator

***Abstract*—Blockchain performance has always been a central focus in blockchain research. Among various approaches, sharding blockchain technology stands out as a promising technique. This method involves splitting the blockchain network into smaller, more manageable segments called shards. This structure facilitates faster transaction processing and more rapid block generation, greatly improving the blockchain's performance and scalability. However, this approach introduces specific challenges, particularly in managing cross-shard transaction verification and ensuring inter-shard load balancing. The key in cross-shard transaction verification is to maintain atomicity without significantly compromising performance. Additionally, effective inter-shard load balancing is essential to avoid issues like hot-shards, where an uneven distribution of transaction loads can lead to considerable performance bottlenecks. Successfully addressing these aspects is crucial for the practical implementation and wider adoption of sharding in blockchain systems.**

To provide insights into these challenges and enable accurate assessment of sharding blockchain systems, we developed ShardingSim, a modular, committee-based sharding blockchain simulator. Unlike traditional performance measurement techniques, which often fall short in sharding environments due to their resource-intensive nature and inability to capture dynamic shard interactions, ShardingSim is designed to simulate various sharding structures. It allows for the configuration of different network parameters to measure performance and the utilization of diverse transaction datasets to analyze hot-shard issues. With ShardingSim, our aim is to contribute to the enhancement of scalability and efficiency in sharding blockchain networks, providing valuable insights into their optimization and the complexities of sharding.

***Index Terms*—sharding blockchain, simulation, performance measurement, cross-shard verification, hot-shard issue**

I. Introduction

Blockchain technology has emerged as a groundbreaking innovation, providing a decentralized and secure framework for recording transactions.However, as blockchain networks grow in size and usage, they face significant scalability challenges. Traditional blockchains like Bitcoin and Ethereum can only process a limited number of transactions per second, leading to network congestion and increased transaction fees [1], [2]. To address these limitations, sharding has emerged as a promising solution [3].

Sharding is a database partitioning technique adapted for blockchain networks to enhance scalability and performance. By dividing the network into smaller, more manageable segments known as shards, each capable of processing transactions independently, sharding aims to increase the overall throughput of the network. This approach not only accelerates transaction processing but also maintains the core principles of decentralization and security inherent to blockchain technology [3], [4].

Despite its potential, sharding introduces new complexities, particularly in the realms of cross-shard transaction verification and inter-shard load balancing. These complexities necessitate innovative methods for performance measurement, as traditional approaches often fall short [5]. Traditional performance measurement techniques for blockchain networks are typically resource-intensive and may not accurately reflect the dynamic nature of sharding systems.

Recognizing the need for a more efficient and adaptable method to assess the performance of sharding blockchain protocols, we introduce ShardingSim, a versatile and modular simulator specifically designed for committee-based sharding blockchain protocols. ShardingSim possesses the capability to replicate various sharding mechanisms. It enables thorough investigations of blockchain performance under different network conditions and transactional loads. A key feature of our simulator is its focus on the hot-shard problem, where certain shards bear disproportionately high transaction volumes, potentially leading to significant bottlenecks. Utilizing ShardingSim, we model and analyze RapidChain [6], exploring its behavior in extreme hot-shard scenarios. Our experiments reveal a substantial decline in shard throughput, by up to 90%, under such stress conditions, highlighting critical challenges and limitations inherent in current sharding mechanisms. This insight into the scalability and robustness of sharding protocols is pivotal, as it underscores the need for further research and optimization in sharding blockchain systems.

II. Background And Related Work

A. Sharding in Blockchain Systems

Sharding technology, initially a database concept, has been adeptly adapted into blockchain systems as a crucial solution to enhance performance and scalability. By dividing the network into smaller, manageable segments known as 'shards,' each capable of processing transactions independently, sharding significantly boosts transaction throughput and reduces latency. This structural adaptation not only improves efficiency but also maintains the decentralization and security principles fundamental to blockchain technology.

In the evolution of sharding in blockchain, [6]–[10]have embraced a committee-based sharding architecture. This approach, characterized by components such as node selection, node assignment, epoch randomness, shard reconfiguration, intra-shard consensus, cross-shard transaction processing, and mempool management, plays a critical role in the effective functioning of sharding networks [4]. The node selection involves choosing qualified members via Proof of Work or Proof of Stake methods. Epoch randomness, an essential and unbiased decentralized random number, facilitates fair and secure node assignment to various shards. Intra-shard consensus upholds transaction integrity within each shard. Cross-shard transaction processing is key for managing multi-shard transactions and ensuring consistent state across the network. Mempool management aids in efficient transaction organization and processing, enhancing network throughput and reducing latency. Finally, shard reconfiguration, through periodic node rotation, ensures shard security by preventing collusion. When simulating sharding blockchain performance within an epoch, the emphasis is particularly on the latter aspects — intra-shard consensus, cross-shard transaction processing, and mempool management — as they are central to assessing operational efficiency in active phases.

Our ShardingSim simulator is designed to modular these key components, allowing for the assembly of various sharding blockchain architectures. By simulating these architectures, ShardingSim facilitates the performance evaluation, providing insights into the effectiveness of different sharding approaches and their impact on overall blockchain system performance.

B. Simulation of Blockchain Systems

In the evolving landscape of blockchain technology, the assessment of performance is vital due to its growing complexity and application range. Blockchain's decentralized nature offers notable advantages, but also presents challenges in response time, throughput, scalability, and security. Simulation has become a key solution, providing a way to replicate and analyze complex blockchain operations in a controlled setting. This approach significantly cuts real-world testing costs and complexities. Through simulation, various aspects of blockchain architecture, including network, consensus, data, execution, and application layers, can be rigorously tested under a range of conditions and settings [11].

Shadow Bitcoin [12], Blocksim:Ffaria [13],BlockSim:Pandey [14],BlockSim:Alharby [15] and SIMBA [16], which offer comprehensive simulations of consensus, data, and network layers, providing an in-depth analysis of blockchain operations. However, they are specifically tailored for non-sharding blockchain systems. These tools lack the necessary features to address the intricacies and distinctive challenges posed by sharding blockchains, thus limiting their applicability in simulating and analyzing the performance of sharding blockchain environments.

Our work, ShardingSim, addresses a significant gap in the field of blockchain simulation, particularly in the area of sharding blockchain systems. ShardingSim is a modular simulator that facilitates the assembly of various sharding blockchain architectures. Unlike many existing simulators that primarily focus on proof-of-work (PoW) blockchains, ShardingSim is tailored to simulate the dynamics of sharding in blockchain networks. This includes the simulation of key components like node assignment, intra-shard consensus and cross-shard transaction processing. By simulating these components, ShardingSim allows for a comprehensive performance evaluation of different sharding approaches, thereby contributing significantly to the development and optimization of sharding blockchain systems.

III. ShardingSim Architecture

ShardingSim is designed to model various aspects of committee-based sharding in blockchain systems. ShardingSim excels in simulating transaction handling within an epoch, allowing for the quantification of transaction processing speed (TPS) and confirmation time under various parameters such as block size, committee size, and peer bandwidth. This simulator provides a comprehensive analysis of transaction processing efficiency in a sharding blockchain environment.

ShardingSim is structured as depicted in Fig. 1. In ShardingSim, the simulation process begins with modeling node behaviors to create peers, using input parameters provided to the sharding config. This configuration guides the creation of peers by the peer factory, which, in conjunction with the bootstrap module, establishes a simulated sharding blockchain network with nodes assigned to specific shards. The transaction generator(Tx Generator) generates transaction data, which the transaction distributor(Tx Distributor) allocates to nodes across different shards for propagation, verification, and block generation. Finally, the outcomes of these operations are collected and analyzed to evaluate the performance of the simulated sharding blockchain system.

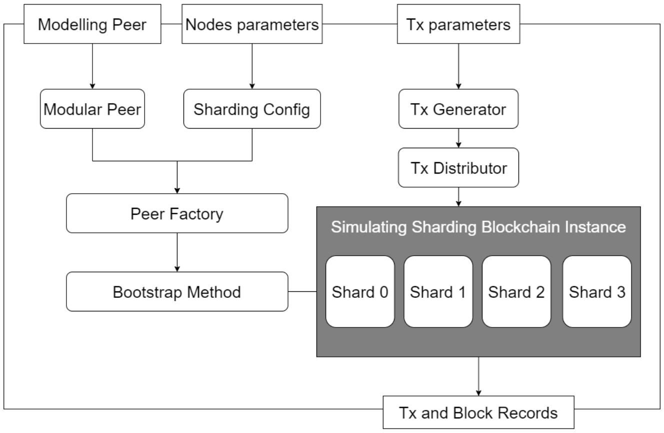

Fig. 1. Workflow of ShardingSim

A. Transaction Generator and Distributor

The transaction generator within ShardingSim features two specialized modes: the historical transaction data mode and the hot-shard issue detection mode.

In the historical transaction data mode, the transaction generator leverages data from the Bitcoin network to provide authentic transaction records for the simulation. This approach ensures that the simulated blockchain activity mirrors real-world scenarios, allowing for a comprehensive analysis of the sharding blockchain's performance under standard operating conditions. The hot-shard issue detection mode is designed to generate specific transaction scenarios that target the analysis of hot-shard challenges. This mode strategically creates and distributes transactions to simulate concentrated load conditions on certain shards. This targeted approach is essential for examining the resilience and efficiency of sharding mechanisms when faced with uneven transaction distribution, offering critical insights into the scalability and adaptability of sharding protocols under stress conditions.

Alongside this, the transaction distributor plays an essential role in allocating these transactions to nodes across the simulated sharding blockchain network, facilitating realistic transaction propagation and processing scenarios.

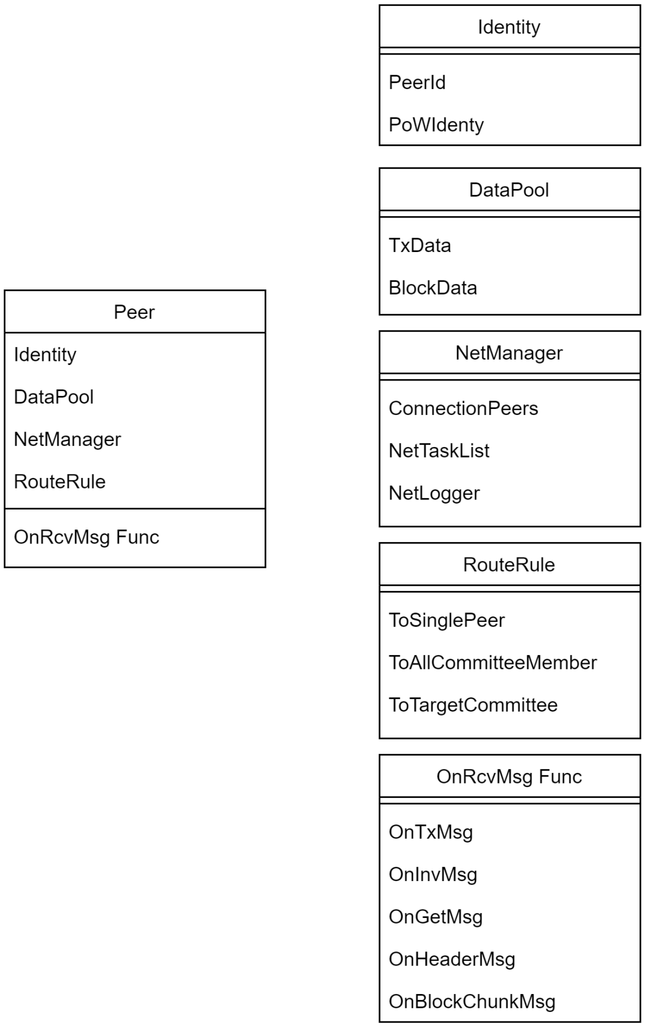

Fig. 2. Structure of Modular Peer

B. Modular Peer

The modularity of ShardingSim's peers is pivotal for emulating the diverse behaviors of sharding blockchains. This is achieved through a configuration that delineates specific responsibilities to different components of a peer:

- **Identity**: Manages the peer's unique identifiers, facilitating node recognition and verification in a decentralized network.
- **DataPool**: Acts as a storage for the peer's transactions and block data, enabling the peer's active engagement in the blockchain processes.
- **NetManager**: Oversees the peer's network settings, managing connections, bandwidth, latency, and network tasks to ensure effective communication within the network.
- **RouteRule**: Defines the peer's data routing methods, including the broadcasting of data within a committee and across shards, crucial for data propagation simulation.
- **OnRcvMsg Functions**: Specifies the peer's responses to received messages, with functions like `OnClientTxMsg` and `OnInvMsg` detailing the processing methods for various data packets, aligning with the peer's consensus protocol.

Through its modular architecture, ShardingSim allows for the flexible simulation of assorted sharding blockchain protocols by configuring or extending each module independently, thus offering a versatile simulation environment.

C. Peer Factory and Bootstrap Method

The Peer Factory within ShardingSim is responsible for generating nodes as defined by the user's sharding config. Once nodes are generated, the Bootstrap Method is applied to assign nodes to their respective committees and set up the network connections.

In its present iteration, ShardingSim utilizes a simplified Bootstrap Method that randomly distributes nodes across different committees and establishes their network connections. This method is sufficient for the simulator's core goal of TPS measurement, ensuring that the simplicity of the bootstrap process does not impact the accuracy of transaction processing speed assessments.

D. Simulating Sharding Blockchain Instance

After the initial setup, nodes in ShardingSim are allocated to their respective committees. The transaction distributor then begins the task of allocating transactions to the nodes. These transactions are processed by the nodes according to consensus rules outlined in the OnRcvMsg functions. For instance, the OnClientTxMsg function is utilized for handling client-submitted transactions. The process encompasses transaction forwarding, validation, entry into blocks, and the broadcasting of these blocks within the network. Blocks produced during the simulation are recorded, providing data for later analysis.

In summary, ShardingSim offers a detailed framework for simulating transaction processing within sharding blockchain systems. Its strength lies in its modular design, particularly through the modular peer component, which allows for the customization of various sharding blockchain parameters. This modularity enables the framework to replicate a wide range of sharding mechanisms, providing a versatile environment for testing and measuring transaction processing speed (TPS) and transaction confirmation times across different network configurations.

IV. Use Case: Modelling RapidChain

A. Overview of RapidChain

In the simulation of RapidChain [6], emphasis is placed on modeling key components, specifically: cross-shard transaction verification, intra-committee consensus, and inter-shard routing methods. These components are vital for understanding the protocol's performance and scalability.

Cross-Shard Transaction Verification and Inter-Shard Routing: When a transaction occurs, it often involves inputs from multiple shards. Each shard has a committee responsible for verifying the part of the transaction within its shard. Consider a transaction, **tx**, with two inputs. If both inputs are valid, the committees for these inputs verify them and then forward the confirmation to the committee responsible for the output. This successful verification leads to **tx** being processed. However, if any input is invalid, either because it's already been spent or doesn't exist, the committees will not approve them. As a result, **tx** won't be processed and will be withdrawn, ensuring robust and accurate transactions across the blockchain's shards.

RapidChain also utilizes a Kademlia-inspired routing mechanism [17] for efficient inter-shard communication. This mechanism is crucial for directing transaction data to the correct output committee, determined by hashing the transaction ID. This routing process ensures that transactions are accurately directed and processed within the blockchain's sharding structure.

Intra-Committee Consensus with Block Broadcasting and Agreement:The intra-committee consensus in RapidChain's shard involves two critical components. Firstly, it utilizes a rapid data block propagation among committee members. This is specifically achieved through an IDA-gossip protocol, which is a fast gossip protocol enhanced by Information Dispersal Algorithms (IDA) [18]. Secondly, the process includes reaching a consensus on the block header, initiated by electing a local leader. The leader compiles and distributes the block, followed by a Byzantine fault-tolerant consensus mechanism centered on the block header. This dual approach of block propagation and header consensus is crucial for validating and adding new blocks to the shard's blockchain, significantly affecting the network's transaction processing speed and reliability.

B. Modelling RapidChain in ShardingSim

The simulation of RapidChain within ShardingSim necessitates the incorporation of two critical components, reflecting the protocol's unique transaction routing and message handling characteristics:

- **RapidRouteRule for Network Routing**: The RapidRouteRule component in the simulation is designed to closely imitate RapidChain's Kademlia-like routing, crucial for message routing in inter-shard communication. Its implementation in the simulation primarily aims to accurately reflect RapidChain's transaction routing techniques, focusing on the correct delivery of transactions to their intended shards.
- **OnRcvMsgFunc for Message Processing**: This set of functions captures the essence of RapidChain's message reception logic. It dictates how nodes process incoming messages and is integral to the functioning of the network.
  - *Transaction Verification*: Employs the OnRcvMsgFunc to scrutinize incoming transactions, validating them by checking against the ledger, thus ensuring the integrity of the blockchain.
  - *Block Consensus*: Employs the OnHeaderMsg and OnBlockchunkMsg Func for the IDA-gossip of blocks within a committee and achieving Byzantine consensus on the block header,

By accurately implementing these components, ShardingSim can effectively model the behavior of RapidChain's network during an epoch, providing insights into the efficiency and scalability of the sharding approach.

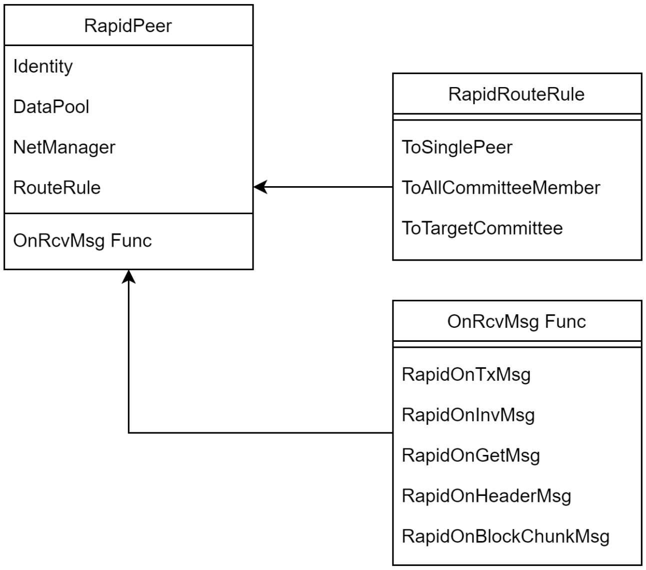

Fig. 3. Constructing the Peer Structure of RapidChain

V. Experiment

This section shows the experimental evaluation aimed at validating the simulation's accuracy to the RapidChain protocol and assessing its performance under hot shard stress scenarios. The validation experiments confirm the accuracy of the simulation, comparing key performance metrics against documented behaviors of RapidChain [6]. The focus then shifts to hot-shard experiments, where the transaction generator is specifically designed to induce hot shard conditions by clustering transaction hashes onto a select number of shards. The experiments are conducted in an environment equipped with an AMD Ryzen 7 CPU and 128GB of memory. The network is set up to simulate nodes each with a bandwidth of 20Mbps and a consistent communication latency of 100 milliseconds. And the block size is configured to be 2MB,

A. Validation

In the validation phase of ShardingSim, experiments were conducted to assess transaction per second performance(TPS) in networks ranging from 500 to 4000 nodes. The experimental outcomes, as depicted in Fig.4, demonstrate that ShardingSim accurately mirrors the TPS trends of RapidChain when scaling node count. The alignment between ShardingSim's TPS results and those of RapidChain underscores the simulator's effectiveness in replicating real-world performance. Despite some variations in TPS data, these remain within the acceptable margin for simulation.

Furthermore, ShardingSim was used to track the blocks produced by each shard, reflecting the load distribution among shards, as illustrated in Fig.5. Analysis of this data indicates that under historical transaction datasets, the load across shards is evenly distributed.

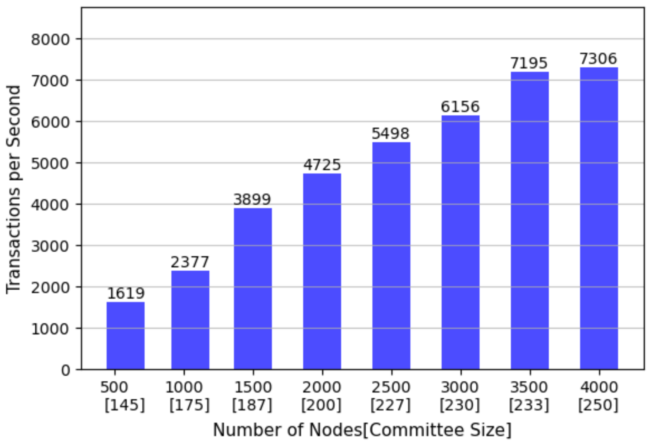

Fig. 4. Simulating Throughput scalability of RapidChain

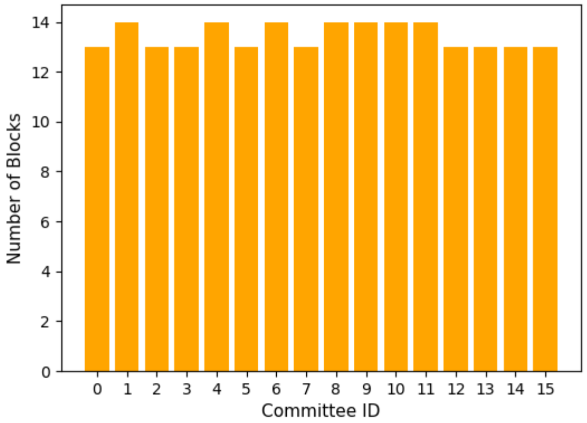

Fig. 5. Number of Blocks between Committees at Node 4000 with Committee Number 16

B. Hot-Shard Experiments

The hot-shard experiments are designed to test the system's performance when subjected to concentrated transaction loads. In these experiments, the transaction generator is configured to create transactions with hash suffixes that map them to the same committee, simulating a scenario where one shard experiences a significantly higher volume of transactions than others. And all other parameters, including network configurations and block properties, remain unchanged. The results of the TPS experiments are depicted in Fig.6, demonstrating that under conditions of extreme hot shards, the system's performance does not improve with an increase in the number of nodes. Additionally, the distribution of load across the different shards is illustrated in Fig.7, it can be observed that solely using transaction hashes for task distribution, under specific circumstances, can lead to transactions being concentrated in a single committee, resulting in a decline in performance.

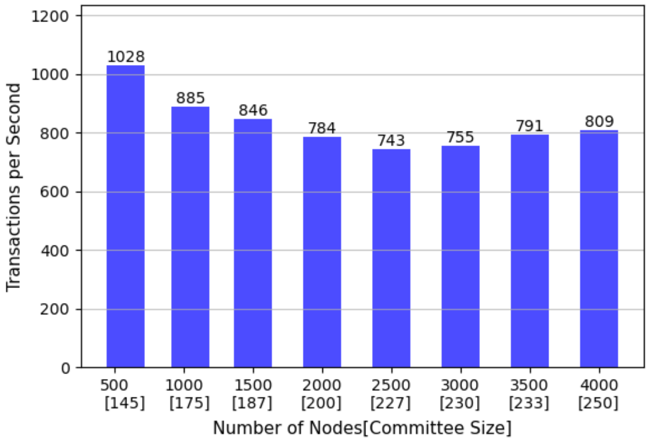

Fig. 6. Simulating Throughput scalability of RapidChain under Hot Shard Issue

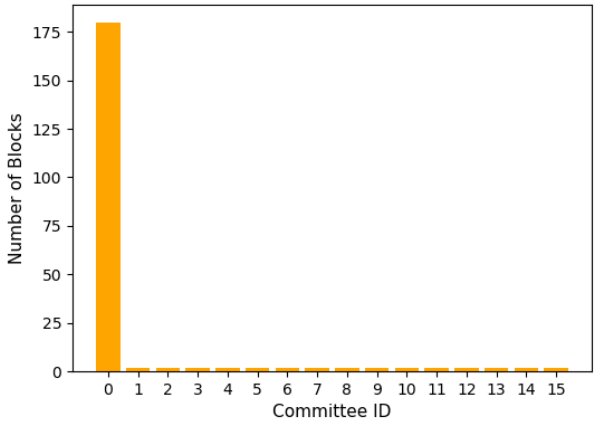

Fig. 7. Number of Blocks between Committees at Node 4000 with Committee Number 16 under Hot Shard Issue

VI. Conclusion

In this paper, we presented a modular, committee-based sharding blockchain simulator designed to simulate the complexities of committee-based sharding blockchain. Through validation, we demonstrated the simulator's ability to accurately replicate the behavior of established committee-based sharding protocols such as RapidChain. Our validation process confirmed the reliability and effectiveness of our simulation tool. Furthermore, we conducted a series of experiments to investigate hot-shard issues, leveraging our simulator's capability to handle different transaction datasets and assess shard load balancing.

In conclusion, our work ShardingSim contributes a simulator specifically tailored for committee-based sharding blockchain. This simulator effectively simulates the operational dynamics of such networks and provides a platform for evaluating shard load balancing and hot shard issues. This simulator offers assistance in the development and analysis of sharding blockchain architectures, aiding in better understanding and optimization of these systems.

In future work, the simulation will expand to cover more sharding blockchain protocols, integrating diverse transaction patterns and deepening the understanding of shard load distribution. This effort is aimed at enhancing the scalability and efficiency of committee-based sharding blockchain networks.

References

[1] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. On scaling decentralized blockchains: (a position paper). In *International conference on financial cryptography and data security*, pages 106–125. Springer, 2016.

[2] Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert Van Renesse, and Emin Gün Sirer. Decentralization in bitcoin and ethereum networks. In *Financial Cryptography and Data Security: 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, February 26–March 2, 2018, Revised Selected Papers 22*, pages 439–457. Springer, 2018.

[3] Hung Dang, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin, and Beng Chin Ooi. Towards scaling blockchain systems via sharding. In *Proceedings of the 2019 international conference on management of data*, pages 123–140, 2019.

[4] Yizhong Liu, Jianwei Liu, Marcos Antonio Vaz Salles, Zongyang Zhang, Tong Li, Bin Hu, Fritz Henglein, and Rongxing Lu. Building blocks of sharding blockchain systems: Concepts, approaches, and open problems. *Computer Science Review*, 46:100513, 2022.

[5] Caixiang Fan, Sara Ghaemi, Hamzeh Khazaei, and Petr Musilek. Performance evaluation of blockchain systems: A systematic survey. *IEEE Access*, 8:126927–126950, 2020.

[6] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. Rapidchain: Scaling blockchain via full sharding. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 931–948, 2018.

[7] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 17–30, 2016.

[8] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta, and Bryan Ford. Omniledger: A secure, scale-out, decentralized ledger via sharding. In *2018 IEEE symposium on security and privacy (SP)*, pages 583–598. IEEE, 2018.

[9] Zicong Hong, Song Guo, Peng Li, and Wuhui Chen. Pyramid: A layered sharding blockchain system. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.

[10] Huawei Huang, Xiaowen Peng, Jianzhou Zhan, Shenyang Zhang, Yue Lin, Zibin Zheng, and Song Guo. Brokerchain: A cross-shard blockchain protocol for account/balance-based state sharding. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pages 1968–1977. IEEE, 2022.

[11] Remigijus Paulavičius, Saulius Grigaitis, and Ernestas Filatovas. A systematic review and empirical analysis of blockchain simulators. *IEEE access*, 9:38010–38028, 2021.

[12] Andrew Miller and Rob Jansen. {Shadow-Bitcoin}: Scalable simulation via direct execution of {Multi-Threaded} applications. In *8th Workshop on Cyber Security Experimentation and Test (CSET 15)*, 2015.

[13] Carlos Faria and Miguel Correia. Blocksim: blockchain simulator. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 439–446. IEEE, 2019.

[14] Santosh Pandey, Gopal Ojha, Bikesh Shrestha, and Rohit Kumar. Blocksim: A practical simulation tool for optimal network design, stability and planning. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 133–137. IEEE, 2019.

[15] Maher Alharby and Aad Van Moorsel. Blocksim: a simulation framework for blockchain systems. *ACM SIGMETRICS Performance Evaluation Review*, 46(3):135–138, 2019.

[16] Seyed Mehdi Fattahi, Adetokunbo Makanju, and Amin Milani Fard. Simba: An efficient simulator for blockchain applications. In *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, pages 51–52. IEEE, 2020.

[17] Petar Maymounkov and David Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer, 2002.

[18] Noga Alon, Haim Kaplan, Michael Krivelevich, Dahlia Malkhi, and JP Stern. Addendum to scalable secure storage when half the system is faulty. *Information and Computation*, 2004.