ACCEPTED FROM OPEN CALL

The State-of-the-Art and Promising Future of Blockchain Sharding

Qinglin Yang, Huawei Huang, Zhaokang Yin, Yue Lin, Qinde Chen, Xiaofei Luo, Taotao Li, Xiulong Liu, and Zibin Zheng

The authors research the state-of-the-art studies published in the past three years on the subject of blockchain sharding.

Abstract

Blockchain sharding is a significant technical area, improving the scalability of blockchain systems. It is regarded as one of the potential solutions that can achieve on-chain scaling, and significantly improve the scalability of blockchains without alleviating the decentralization feature of blockchain. To provide a reference and inspire participation from both the academic and industrial sectors in the area of blockchain sharding, we have researched the state-of-the-art studies published in the past three years. We have also conducted experiments to show the performance of representative sharding protocols such as Monoxide, LBF, Metis, and BrokerChain. We envision the potential challenges and promising future of sharding techniques in terms of the urgent demands of high throughput required by emerging applications such as Web3, Metaverse, and Decentralized Finance (DeFi). We hope that this article is helpful to researchers, engineers, and educators, and will inspire subsequent studies in the field of blockchain sharding.

Introduction

Blockchain technology is the foundation for decentralized networks, which guarantees transparency, immutability, distribution, and anonymity of the interoperability of networks. Despite the promising benefits of blockchain technologies, past implementations of applications on blockchain reveal critical dependencies between scalability, security, and decentralization. For example, the urgent requirements of high throughput incurred by emerging applications (e.g., Web3, Metaverse, and DeFi services) lead to blockchain scalability and consensus challenges because micro-payments dominate the high-volume, low-value, per-transaction mass-market services.

Blockchain systems require all nodes to participate in consensus, computation, and full storage to satisfy security and decentralization. In reality, the storage space of an individual node is limited. As ledger data continually expands, the storage overhead for a particular node keeps growing. For instance, the data volume of an Ethereum full node has approached 6 Terabytes (TB) under the condition of 25 transactions per second (TPS), which far exceeds the disk storage capacity of most home-use devices. When the blockchain system's TPS reaches 20,000 or even larger, the storage overhead is greatly magnified. Therefore, if the storage overhead of individual nodes is not addressed, the entry threshold for blockchain will restrict the participation of most general users, which is contrary to the decentralization property of blockchain.

Scalability is a fundamental aspect of blockchain technology that ensures its viability, efficiency, and the ability to support a broad range of applications. The scalability issues of blockchains have attracted substantial attention since the distributed ledger technology was proposed in 2008. To improve the scalability of blockchain, several representative solutions have emerged from the literature, including Layer1, and Layer2 solutions.

As a Layer1 solution, sharding is adopted for scaling out the blockchain, aiming to increase its throughput and capacity [1]. Although it was first proposed in the field of databases, sharding is becoming a promising future of blockchain scalability. Meanwhile, it is the key to enabling a high-level throughput measured by TPS and allowing developers and users to regularly use the mainstream Layer1 platform (e.g., Ethereum) at an affordable cost.

Nevertheless, it is also one of the mostly misunderstood concepts in the blockchain ecosystem. Sharding is different from several popular Layer2 solutions such as Bitcoin lightning network and Zero-Knowledge Rollups, in which the sharding technique does not attempt to move transactions off of the blockchain. Layer2 solutions are protocols built on top of a base blockchain (Layer1) to enhance transaction speeds and scalability while maintaining the security of the underlying blockchain. These solutions process transactions off-chain or more efficiently, periodically batch or consolidate them back to the main chain.

Sharding divides all consensus nodes into smaller groups by modifying the blockchain architecture. Each shard is capable of processing its transactions and smart contracts in parallel [2], as illustrated in Fig. 1 the overall TPS. With multiple blockchain shards processing transactions concurrently, the network can handle a large number of transactions in each round of global consensus.

Sharding can potentially allow a large-scale nodes to participate in the network, as the threshold to run a consensus node can be lowered in terms of storage and computational power. Although Layer2 rollups have become a buzzing technology routine and the Ethereum community prefers the scaling via rollups, the invention

Qinglin Yang is with Guangzhou University (Huangpu), China; Huawei Huang (corresponding author), Zhaokang Yin, Yue Lin, Qinde Chen, Xiaofei Luo, Taotao Li, and Zibin Zheng are with Sun Yat-Sen University, China; Xiulong Liu is with Tianjin University, China.

Digital Object Identifier: 10.1109/MCOM.004.2400026

of Danksharding is viewed as a giant upgrade in Ethereum's future roadmap. This is because Danksharding introduced a new structure named blobs such that Ethereum transactions can be confirmed more efficiently. Furthermore, the benefit of Danksharding can be added on top of rollup solutions. Conventional sharding focuses on dividing the blockchain into multiple independent shards, each processing its transactions and data. It faces challenges related to cross-shard communication and maintaining security. Danksharding aims to enhance Ethereum's scalability by optimizing data availability for rollups, treating the entire network as a unified shard. It simplifies some aspects of sharding while introducing new mechanisms to ensure data availability and integrity. Overall, Danksharding represents a specialized approach tailored for Ethereum's ecosystem and rollup-based scaling, while conventional sharding is a broader technique applicable to various blockchain architectures.

In addition to sharding, several other technical directions have been proposed to improve the performance of blockchains. Those directions include applying the structure of Directed Acyclic Graph (DAG), designing new consensus algorithms, increasing the size of blocks, and implementing Layer2 solutions like Ethereum's zero-knowledge rollup, and so on. As depicted in Fig. 2, the scaling solutions support the TPS requirements of top application layers such as decentralized finance (DeFi), digitalization, Web3, decentralized autonomous organization (DAO), and Metaverse. DAG and increasing the block size can improve the TPS from the perspective of changing the structure of blocks. DAG structures inherently facilitate parallel processing, which allows for the simultaneous generation of multiple blocks. This capability significantly boosts the efficiency of block generation compared to traditional linear blockchains, where blocks are produced sequentially [3] New consensus algorithms aim to improve the efficiency of consensus. Compared with the aforementioned solutions, the advantages of sharding are in leveraging transactions' parallel processing and reducing the redundancy spent on transaction execution or storage.

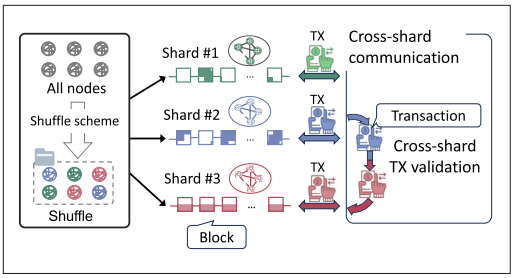

FIGURE 1. Architecture of blockchain sharding.

Although sharding benefits the blockchain architecture, its risks should not be ignored. For example, when a blockchain network is divided into smaller groups, the risk of suffering from the 51 percent attack increases. Some more vulnerability issues are also induced, including inter-shard, intra-shard, and system-level issues.

Until now, some related surveys have discussed sharding technologies of blockchain from a distinct perspective. For example, the existing survey on sharding [4] breaks down sharding blockchain systems into functional components, including node selection, epoch randomness, node assignment, intra-shard consensus, cross-shard transaction processing, shard reconfiguration, and motivation mechanism. They describe each component's interfaces, functionality, and properties and how they come together to form a sharding blockchain system. Furthermore, Liu *et al*. [5] discuss the scalability problem of blockchain and the potential solution of sharding technology. It highlights the limitations of existing consensus mechanisms and the need to enhance blockchain scalability. Although the above surveys [4, 5] provide a comprehensive review that properly classifies existing schemes and a thorough analysis using a uniform set of evaluation criteria, they all ignore the discussion on the TPS performance of sharding technologies.

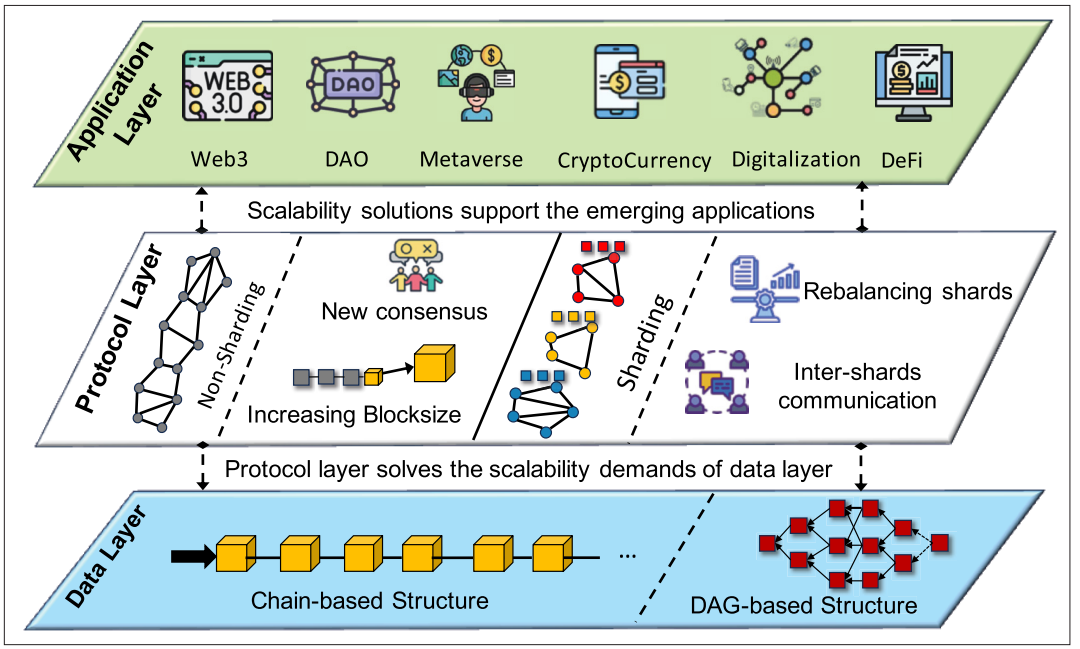

FIGURE 2. The position of sharding in the field of blockchain scalability technologies.

Theme	Descriptions	Methods	TPS	Confirmation Latency (seconds)	Security
Processing Cross-shard Transactions	Authenticated data structures	GriDB [6]	Improves by 1.40×for the low skewness and 1.37× for the high skewness	221 ms for clinet	●
	Account split, graph segmentation	BrokerChain [1]	2.9k with 64 shards	14.87 with 64 shards	◖
Balancing Shard Workloads	Machine learning detection	LB-Chain [7]	Increases 10 %	Reduces up 90 % confirmation latency	◖
	Cross-chain operations, Relay chain-based sharding	Sliver [8]	Improves 5×, compared to state-of-the-art sharding protocols	×	◖
Transaction Processing related to Smart Contract	Smart contract storage and execution decoupling	Jenga [2]	4.3k with 12 shards	≈ 10	◖
	Confirm the transaction order, reducing conflict	Prophet [9]	1,203	Decreases by 62.9 %, compared with Monoxide	○
	Cross-shard contract execution engine	ShardCon [10]	10× increase	2× decrease	●
Shard Reorganization	Consistent hashing, State trie	S-Store [11]	≈ 28.5k in concurrent addition	≈ 125 / 1 million transactions	○
	Reorganization of state trie	tMPT [12]	198 % higher than Ethereum's full sync method	≈ 100 with 150 millions data volume	◖
	Node selection and epoch reconfiguration	FS [13]	20k with 128 nodes, 2 shards and 1MB block size	0.4 with 128 nodes, 2 shards and 1MB block size	●
Multi-shard Security	Multi-shard supervision	CoChain [14]	Compared with harmony, 35× with 6,000+ nodes	≈ 38	●
Network Communications	Overlapping network, virtual accounts	Overshard [15]	128× the throughput at one block interval	≈ 12	○
Directed Acyclic Graph (DAG)	Adaptive sharding, avatar account caching	SharDAG [3]	14,666 TPS under 16 shards	< 5	◖

Note: ● denotes high security ◖ denotes medium security ○ denotes low security

TABLE 1. A summary of blockchain sharding techniques.

Thus, we are motivated to conduct this overview to help readers realize both the benefits and challenges of sharding. We wish to contribute insights into this promising technique for future blockchain systems. Firstly, we explain why sharding is a critical scalability solution for blockchain architecture. We also review the state-of-the-art studies on blockchain sharding. Secondly, we demonstrate the different performance of several popular sharding protocols, including Monoxide, Load Balancing Framework (LBF), Metis, and BrokerChain [1]. Finally, to inspire successive studies, we envision the open issues of blockchain sharding.

Classification of Blockchain Sharding Techniques

As illustrated in Table 1, we summarized cutting-edge studies on sharding techniques regarding different performance, including TPS, confirmation latency, network communication overhead, and security. These metrics mainly cover all critical aspects of a sharding blockchain's performance, from technical efficiency and security to user satisfaction. By monitoring and optimizing these metrics, developers can fine-tune the system for better performance, ensuring it meets its design goals of scalability, efficiency, and reliability.

Preliminaries of Cross-Shard Transactions

Figure 3 demonstrates multiple types of transactions, including a regular transaction, an original cross-shard transaction, an intra-shard relay transaction, and an inter-shard relay transaction. For example, Monoxide splits an original cross-shard transaction into an *intra-shard* relay transaction and an *inter-shard* relay transaction, aiming to achieve the so-called *eventual atomicity* of a cross-shard transaction. The intra-shard relay transaction deducts the funds from the payer's account in the source shard, while the inter-shard relay transaction deposits the funds into the payee's account in the destination shard. These two relay transactions are connected through transaction-relaying messages.

Miners select transactions from the local transaction pool (TXpool) to generate a new block while handling all transactions in each shard. These miners can determine if an original transaction in the block is a cross-shard transaction (CTX) or an intra-shard regular transaction based on the locations of the payer and payee accounts. That is, the inter-shard relay transaction of the original CTX will be relayed to the payee's shard (i.e., the destination shard) after the block is committed on the payer's shard local chain.

Once a consensus node receives an inter-shard relay transaction, it checks whether the related intra-shard relay transaction has been successfully included in the blockchain ledger. If the corresponding intra-shard relay transaction is on the chain, this consensus node adds the inter-shard relay transactions to its TXpool for future packaging.

Transaction Per Second (TPS)

All the current sharding blockchains use system-specific specialized methods for implementation, but these methods can't support the flexibility of traditional sharding data management. Meanwhile, sharding blockchain solutions face challenges with many cross-shard transactions, lengthy processing times, and unbalanced shard loads.

Huang *et al*. [1] propose segmenting the network's account structure, allowing special Broker accounts to exist in multiple shards simultaneously. By leveraging the turnover capability of Broker accounts, the number of cross-shard transactions can be reduced. Moreover, cross-shard transactions are facilitated through these Broker accounts acting as intermediaries. Additionally, to balance transaction loads across shards, the authors apply the Metis tool and a graph partitioning approach to distribute each account into the most suitable shard. Meanwhile, Li *et al*. [2] propose a solution, named Jenga, to accelerate smart contract execution by reducing both cross-shard consensus and cross-shard communication. Jenga decouples the execution of smart contracts from state storage. Different shards are stored in diverse states. Execution channels are established based on state sharding. Each node simultaneously belongs to one shard and one execution channel. Each channel overlaps with all shards, with different channels executing different contracts. Through the overlapping nodes, contract states can be broadcast directly between state shards and execution channels without additional cross-shard communication. Therefore, transactions involving smart contracts can be executed in one go without requiring multi-round cross-shard consensus.

To address the challenges of storing and ensuring the processing consistency of cross-shard transactions in DAG-based blockchains, Cheng *et al*. [3] provide an adaptive scalable and efficient sharding mechanism, named SharDAG. It is composed of a cross-shard avatar account caching scheme, a Byzantine cross-shard verification mechanism, and a two-tier state storage model. The authors believe that SharDAG outperforms the state-of-the-art (e.g., Monoxide) regarding latency and throughput, providing storage scalability.

The aforementioned research focuses on the improvement of TPS, but they weaken the following two aspects in sharding blockchains, such as confirmation latency, network communication overhead, and security.

Confirmation Latency

The confirmation of cross-chain transactions is typically facilitated through intermediaries (e.g., relay chains), which could potentially become performance bottlenecks. Although sharding technology is promising for relay chains to accelerate the confirmation of cross-chain transactions, randomly distributing the transactions related to cross-chain across different shards doesn't boost the confirmation [7]. The two main factors that influence the performance of sharding blockchains are the proportion of cross-shard transactions and the balance of shard loads. Considering these factors, Li *et al*. propose a machine learning-based account migration scheme, LB-Chain, to balance the shard loads by predicting the future transaction volume of each account. Based on these predictions, an account partitioning algorithm is executed to ensure load balance across shards.

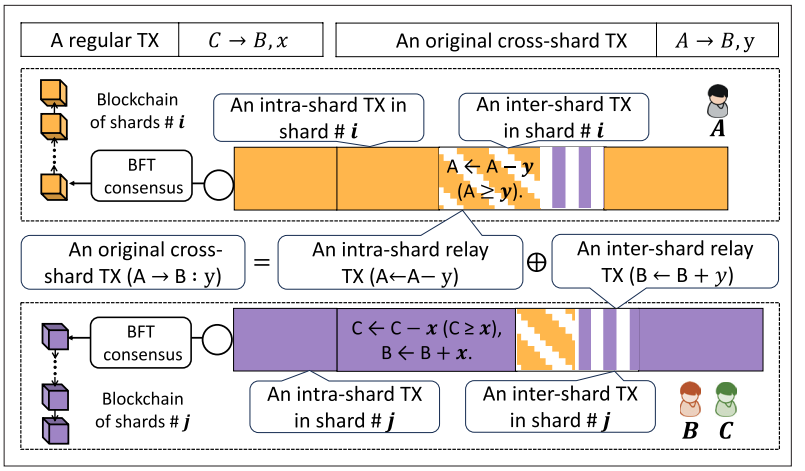

FIGURE 3. The illustration of multiple types of transactions, including a regular transaction, an original cross-shard transaction, an intra-shard relay transaction, and an inter-shard relay transaction.

In sharding blockchains, the cross-shard transactions involving smart contracts often lead to significant transaction conflicts and transaction aborts. Because different shards are orchestrated independently and randomly. Existing blockchain sharding protocols typically require complex multi-round cross-shard consensus protocols and extensive communication to execute smart contracts during state transfer.

Hong *et al*. [9] propose the *Prophet* solution to reduce transaction conflicts. *Prophet* relies on the collaboration from nodes of different shards to pre-execute cross-shard transactions, determining the call relations of various contracts. Through the cooperation of the shards, *Prophet* implements stateless transaction ordering to establish a global sequence. Following this order, shards execute and commit transactions in a coordinated manner, thereby avoiding conflicts.

Furthermore, the implementation of shard reconfiguration faces several challenges, including how to synchronize a large amount of state data and how to reduce the latency of large-scale blockchain shard reconfigurations. Huang *et al*. [12] devise a new MPT structure, named tMPT, to make state data synchronization more efficient. Meanwhile, a new shard reconfiguration protocol is proposed to minimize the impact of shard reconfiguration on transaction processing.

Multi-Shard Security

The consensus execution of blocks is confined within individual independent shards, which diminishes the fault tolerance of the blockchain. Li *et al*. [14] exploit increasing the number of blockchain shards to improve throughput while ensuring system security. The proposed *CoChain* scheme supports multiple small shards in the system. However, each shard is overseen by several other shards. The blocks produced by a shard can achieve cross-shard consensus among these overseeing shards. If a shard is detected to be overtaken by malicious nodes, other shards take over its subsequent consensus, thus restoring the system. In this way, shards can tolerate the presence

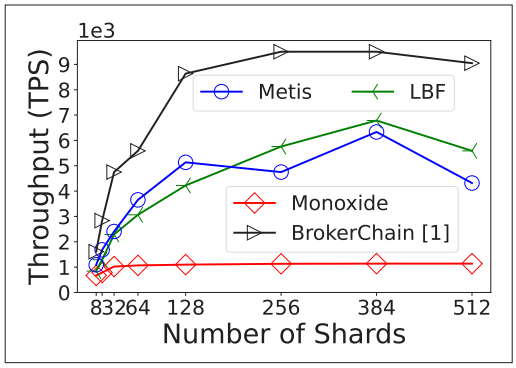

FIGURE 4. Multi-Sharding vs Single-Sharding, with Number of Shards 32.

of more malicious nodes while ensuring system security. Therefore, it's possible to safely reduce the size of shards, increase the number of shards, and raise concurrency.

In a transaction-based sharding blockchain, the heterogeneity of the shards leads to significant consensus latency variations among them. Accordingly, the shard with the longest latency constrains the system's throughput.

Flexible Sharding (FS) [13] includes node selection and epoch reconfiguration. It utilizes the Intra-Shard BFT protocol for processing intra-shard transactions and confirming shard member lists for each new epoch, and the Cross-shard BFT protocol for handling cross-shard transactions. The threshold-vote rule is employed to defend against node censorship attacks. Each shard reconfigures itself, with nodes submitting PoW solutions and employing a threshold voting rule to confirm the member list for the next epoch. The secure FS protocol is defined to satisfy properties related to consistency, common prefixes in the same blockchain, and no conflict among different intra-shard blockchains.

GriDB [6] delegates massive cross-shard data exchange to randomly selected nodes from different shards. Untrusted delegates collaborate to generate succinct proofs for cross-shard data exchanges. Subsequently, consensus handles low-cost proof verification. Hence, new authenticated data structures (ADS) are introduced to address the database services' verification requirements. The method extends the threat model and simplifies traditional accumulator-based ADS for verifiable cross-shard queries. An off-chain and live approach for inter-shard load balancing is proposed to ensure efficiency and availability during balancing.

Network Communication Overhead

In a sharding blockchain, irrational transaction distribution strategies can lead to instability in transaction queues across shards. If the transaction queue of a shard cannot remain stable, transactions directed to that shard might not be promptly included in the consensus and added to the blockchain. The significant overhead of shard reconfiguration is derived from constructing the Merkle Patricia Tree (MPT) and migrating extensive data. Therefore, Qi [11] proposes S-Store, a scalable data storage technique for permissioned blockchain sharding based on the Aggregate Merkle B+ tree (AMB-tree). S-Store addresses the challenges of data migration and Merkle tree reconstruction in existing Merkle tree-based state storage solutions. The proposed technique utilizes consistent hashing to reduce data migration and employs split and merge operations on AMB-tree to decrease Merkle tree reconstruction overheads.

The OverShard [15] protocol is a full-sharding approach for scaling blockchain that incorporates an overlapping network and virtual accounts. Virtual accounts are created in other shards for each externally owned account (EOA) to improve transaction processing efficiency. The consensus protocol in OverShard involves simultaneous Proof of Work (PoW) mining for multiple shards. Each new block includes parent block hashes from other shards for confirmation. The consensus process includes transaction collection, block generation, and block chaining. The performance of OverShard [15] was evaluated in terms of TPS, confirmation latency, network communication overhead, storage overhead, and security. The system achieved nearly 128× the throughput with confirmation latency remaining at one block interval when 1200 nodes were randomly and evenly allocated to 128 shards, and each node joined 12 shards.

To implement a cross-shard contract execution engine in each off-chain executor, Zhang *et al*. [10] introduce the ShardCon off-chain model. The model includes a contract-driven deployment rule that selects specific TEE providers as the executing set based on the dependency of cross-shard contracts, reducing the synchronizations of contract states. Meanwhile, an off-chain state atomic commit protocol is introduced to adapt to the multi-chain property of a sharding system, ensuring atomicity and consistency. The experimental results show that ShardCon can achieve more than a 10× increase in throughput and a 2× decrease in confirmation latency compared to state-of-the-art sharding systems. It achieves 1,000+ transaction throughput and less than 1s confirmation latency for complex cross-shard contracts with state synchronization requirements.

Performance of Blockchain Sharding

To prove the promising performance of blockchain sharding, we present several experimental results of a blockchain sharding system. The results are obtained from an open-sourced testbed, namely BlockEmulator, which is an experimental tool enabling blockchain sharding.

Basic Settings

Baselines: We implement several sharding blockchain systems for performance evaluation, including Monoxide, Metis, LBF, and BrokerChain. Metis is indeed a powerful software package used for various computational tasks involving graphs and matrices. LBF periodically updates the distribution of accounts to achieve a balanced TX distribution.

Dataset: We crawled 1.67 million real historical transactions from Ethereum. These transaction data include hash, timestamp, tokens transferred, value, transaction fee, status, and so on.

Experimental Settings: We implement transaction-driven simulation tests using Python. In the experiments, each block is set to 2,000 transactions, with a block interval of 8 seconds. We assume that the bandwidth between two nodes has no limitation since the experiments are conducted at a local machine. For the experiments of Brokerchain, we set the partition shard as 1 and all the shards also act as mining shards.

Experimental Environments: The experiment is conducted on a machine equipped with an Intel® i9-13900K 24-core CPU and 32GB of RAM.

Results and Discussion

We compare the performance of sharding with a single-shard system in terms of TPS. To figure out the effect of the number of shards on workload performance, we continuously increase the transaction arrival rate under a fixed number of shards. Figure 4 shows the performance of sharding systems exhibits an initial increase trend followed by convergence. The yellow curve indicates the theoretical maximum TPS limit of 250 for the traditional single-shard system. It is notably lower than other sharding systems' performance. This is because sharding systems can parallelize transaction processing. Furthermore, BrokerChain can reduce the number of cross-shard TXs to a certain low degree. The average cross-shard TX ratios of Monoxide, LBF, Metis, and BrokerChain are 98.6 percent, 98.6 percent, 83.5 percent, and 7.4 percent, respectively [1] The sharding mechanism significantly enhances the performance of throughput. While sharding introduces cross-shard transactions, the performance benefits outweigh the increased system workload.

Next, we explore the performance potential of sharding. In Fig. 5, transaction confirmation latency decreases with the increase in shards. It keeps a low latency after the number of shards exceeds 128. For sharding blockchain systems, the performance improvements made by the increase of shards are limited to the blockchain system workload. The uneven distribution of workloads results in the convergence of throughput, such as some shards being underutilized while others are overloaded.

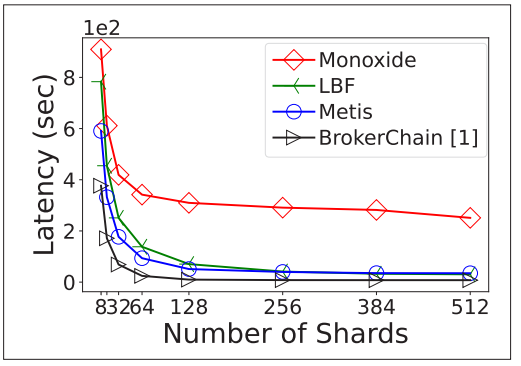

FIGURE 5. Latency with TX arrival rate 16,000 TXs/Sec.

Open Issues

Although blockchain sharding has received notable research attention, it faces multiple challenges in practical deployment. Hence, some further challenges and open issues should be addressed, such as *cross-shard communication*, *rebalancing shards*, *security concerns*, *complexity*, and *compatibility of smart contracts*.

Cross-Shard Communication: Cross-shard transactions are always sophisticated since they span multiple shards in a sharding blockchain. Ensuring data consistency and atomicity across shards is one of the most significant challenges in cross-shard transactions. Each shard operates as a semi-independent unit with its state and ledger. When a transaction involves multiple shards, it is crucial to maintain consistency, meaning all involved shards must agree on the transaction outcome. This necessitates sophisticated protocols to ensure that a transaction either fully commits across all shards or does not commit at all, avoiding partial updates that could lead to inconsistent states. Furthermore, the need for additional communication between shards introduces latency, as each shard must exchange messages to coordinate the transaction. This inter-shard communication can become a bottleneck, especially as the number of shards increases, potentially leading to scalability issues.

Rebalancing Shards: In the sharding blockchain, each node only processes a portion of the network's transactions, which should improve the overall performance of the system. Nevertheless, in reality, the bandwidth of the network and the computing capacity of nodes often become bottlenecks that limit overall performance. Especially during peak transaction periods, the limitations of individual node processing capabilities and network bandwidth can lead to delays in transaction confirmations, thereby affecting user experience. Therefore, as the network grows or shrinks, or as transaction volume changes, there may be a need to rebalance the distribution of data and transaction processing across shards. This rebalancing can be technically challenging.

Security Concerns: Although sharding technology can accelerate transaction processing, it might decrease the overall security of the blockchain network. The existence of security issues poses a significant challenge to the healthy and stable operation of public blockchains. For example, attackers may focus their efforts on specific shards with fewer nodes, as compromising a smaller shard may require fewer resources compared to attacking the entire network. This is sometimes referred to as the "1 percent attack" where an attacker might only need to compromise 1 percent of the network (a single shard) to disrupt it. Additionally, cross-shard transactions involve coordination between shards, which can introduce vulnerabilities. Attackers may attempt to exploit these vulnerabilities to double-spend tokens, especially in scenarios where there is a delay or inconsistency in confirming cross-shard transactions.

Complexity: The complexity of sharding in blockchain refers to the multifaceted technical and conceptual challenges it introduces compared to traditional, non-sharding blockchains. It involves not only the design of intra-shard consensus but also the organization policy of shard committees, inter-shard consensus algorithms, cross-shard communication mechanisms, the security guarantee for blockchain shards, and the atomicity of executing cross-shard transactions, and so on. The complexity issues require meticulous research, design, and validation. In addition, due to the increased complexity, testing a shared protocol or blockchain system to ensure its security and robustness is more challenging compared to non-sharding blockchains.

Compatibility of Smart Contracts with Sharding: Smart contracts, as a significant application on public blockchains, also present a hot topic in terms of their compatibility with sharding mechanisms. Smart contracts need to manage their state

Currently, how to design smart contracts that can execute effectively across shards while ensuring their security and execution efficiency is a technical challenge that needs to be addressed because sharding architecture faces attacks that target specific shards or attempt to manipulate cross-shard communication.

across these shards efficiently since it typically involves dividing the state (data) of the blockchain into different shards. Ensuring that state transitions are atomic and consistent across shards is challenging, especially when contracts need to interact with each other. The introduction of sharding technology requires that smart contracts be able to execute across different shards without affecting their functionality and performance. Currently, how to design smart contracts that can execute effectively across shards while ensuring their security and execution efficiency is a technical challenge that needs to be addressed because sharding architecture faces attacks that target specific shards or attempt to manipulate cross-shard communication.

Even though these challenges are substantial, it's worth noting that significant research and development efforts are ongoing to address them.

Conclusion

We first reviewed cutting-edge studies on sharding techniques to offer a reference for researchers and developers interested in this area. Then, we compared the different performances of several popular sharding protocols such as Monoxide, LBF, Metis, and BrokerChain. Finally, we discussed the challenges and open issues in light of the urgent demands for the massive adoption of blockchain sharding. We hope that this article will encourage further studies in the field of sharding techniques and protocols.

Acknowledgment

This work was partially supported by the National Key R&D Program of China (No. 2022YFB2702304), NSFC (No. 62272496), General Universities Key Field Project of Guangdong Province (No. 2023ZDZX1001, 2024A1515012360), and Fundamental Research Funds for the Central Universities (Sun Yat-sen University, No. 23lgbj019).

References

[1] H. Huang *et al.*, "Brokerchain: A Cross-Shard Blockchain Protocol for Account/Balance-Based State Sharding," *Proc. IEEE Conf. Computer Commun.*, 2022, pp. 1968–77.
[2] M. Li *et al.*, "Jenga: Orchestrating Smart Contracts in Sharding-Based Blockchain for Efficient Processing," *Proc. 42nd IEEE Int'l. Conf. Distributed Computing Systems*, 2022, pp. 133–43.
[3] F. Cheng *et al.*, "Shardag: Scaling Dag-Based Blockchains via Adaptive Sharding," *Proc. 2024 IEEE 40th Int'l. Conf. Data Engineering*, IEEE, 2024, pp. 2068–81.
[4] Y. Liu *et al.*, "Building Blocks of Sharding Blockchain Systems: Concepts, Approaches, and Open Problems," *Computer Science Review*, vol. 46, 2022, p. 100513.
[5] X. Liu *et al.*, "A Survey on Blockchain Sharding," *ISA Trans.*, vol. 141, 2023, pp. 30–43.
[6] Z. Hong *et al.*, "Gridb: Scaling Blockchain Database via Sharding and Off-Chain Cross-Shard Mechanism," *Proc. VLDB Endowment*, vol. 16, no. 7, 2023, pp. 1685–98.
[7] M. Li, W. Wang, and J. Zhang, "LB-Chain: Load-Balanced and Low-Latency Blockchain Sharding via Account Migration," *IEEE Trans. Parallel and Distributed Systems*, 2023.
[8] Y. Tao, B. Li, and B. Li, "On Sharding Across Heterogeneous Blockchains," *Proc. 39th IEEE Int'l. Conf. Data Engineering*, 2023, pp. 477–89.
[9] Z. Hong *et al.*, "Prophet: Conflict-Free Sharding Blockchain via Byzantine-Tolerant Deterministic Ordering," *Proc. IEEE Conf. Computer Commun.*, 2023, pp. 1–10.
[10] J. Zhang *et al.*, "Efficient Execution of Arbitrarily Complex Cross-Shard Contracts for Blockchain Sharding," *IEEE Trans. Computers*, no. 01, 2024, pp. 1–14.
[11] X. Qi, "S-store: A Scalable Data Store Towards Permissioned Blockchain Sharding," *Proc. IEEE Conf. Computer Commun.*, 2022, pp. 1978–87.
[12] H. Huang, Y. Zhao, and Z. Zheng, "tmpt: Reconfiguration Across Blockchain Shards via Trimmed Merkle Patricia Trie," *Proc. IEEE/ACM 31st Int'l. Symposium Quality of Service*, 2023, pp. 1–10.
[13] Y. Liu *et al.*, "A Flexible Sharding Blockchain Protocol Based on Cross-Shard Byzantine Fault Tolerance," *IEEE Trans. Information Forensics and Security*, 2023.
[14] M. Li *et al.*, "Cochain: High Concurrency Blockchain Sharding via Consensus on Consensus," *Proc. IEEE Conf. Computer Commun.*, 2023, pp. 1–10.
[15] B. Yu *et al.*, "Overshard: Scaling Blockchain by Full Sharding With Overlapping Network and Virtual Accounts," *J. Network and Computer Applications*, vol. 220, 2023, p. 103748.

Biographies

Qinglin Yang [M'21] received his Ph.D. degree in Computer Science and Engineering from the University of Aizu, Japan, 2021. He is an assistant professor at the Cyberspace Institute of Advanced Technology/Huangpu Research School of Guangzhou University (Huangpu), China. His current research interests include blockchain, Web3 protocol, and federated learning.

Huawei Huang [SM'22] is an Associate Professor at Sun Yat-Sen University. He received his Ph.D. degree from the University of Aizu (Japan) in 2016. He has served as a research fellow of JSPS, and a program-specific Assistant Professor at Kyoto University, Japan. His research interests include blockchain and distributed computing. He has served as a lead guest editor for multiple blockchain special issues at IEEE JSAC, and IEEE OJCS. He also served as a TPC chair for multiple blockchain conferences and workshops.

Zhaokang Yin is currently pursuing his MS degree at the School of Software Engineering, Sun Yat-Sen University. His research interests mainly include Blockchain.

Yue Lin is currently pursuing his Master's degree at the School of Computer Science and Engineering, at Sun Yat-Sen University. He received his Bachelor's degree from Sun Yat-Sen University in 2020. His research interests mainly include Blockchain.

Qinde Chen is currently pursuing his Ph.D. degree in School of Software Engineering of Sun Yat-sen University. His current research interests mainly include blockchain technology.

Xiaofei Luo received his Ph.D. degree from the Department of Computer Science and Engineering, the University of Aizu, Japan, in 2023. He is currently a postdoctoral researcher at the School of Software Engineering, Sun Yat-sen University, China. His research interests include blockchain, computer networks, and payment channel networks.

Taotao Li received the Ph.D. degree in cyber security from the Institute of Information Engineering, University of Chinese Academy of Sciences, China, in 2022. He is currently a postdoc with the School of Software Engineering, Sun Yat-Sen University, Zhuhai, China. His main research interests include blockchain, Web3, and applied cryptography.

Xiulong Liu is a full-time professor in College of Intelligence and Computing, Tianjin University, China. He received Ph.D. degree from Dalian University of Technology (China) in 2016. He also worked as a visiting researcher in Aizu University, Japan; a postdoctoral fellow in The Hong Kong Polytechnic University, Hong Kong, China; and a postdoctoral fellow in Simon Fraser University, Canada. His research interests include wireless human activity recognition, indoor localization and IoT security&privacy, and so on.

Zibin Zheng [F] is a Professor and Deputy Dean of the School of Software Engineering, Sun Yat-sen University, China. His research interests include blockchain, software engineering, and services computing. He was a recipient of several awards, including the IEEE TCSVC Rising Star Award, IEEE Open Software Award, Top 50 Influential Papers in Blockchain, the ACM SIGSOFT Distinguished Paper Award of ICSE, and the Best Student Paper Award at ICWS.