Scalable Blockchain Architecture: Leveraging Hybrid Shard Generation and Data Partitioning

Praveen M Dhulavvagol[1], Prasad M R[2], Niranjan C Kundur[3], Jagadisha N[4], S G Totad[5]
School of Computer Science and Engineering, KLE Technological University, Hubli, India[1, 5]
Department of Computer Science and Engineering,Vidyavardhaka College of Engineering, Mysuru,India[2]
Department of Computer Science and Engineering, JSS Academy of Technical Education, Bengaluru, India[3]
Department of Information Science and Engineering, Canara Engineering College Bantwal, India[4]

***Abstract*—Blockchain technology has gained widespread recognition and adoption in various domains, but its implementation beyond crypto currencies faces a significant challenge - poor scalability. The serial execution of transactions in existing blockchain systems hampers transaction throughput and increases network latency, limiting overall system performance. In response to this limitation, this paper proposes a static analysis-driven data partitioning approach to enhance blockchain system scalability. By enabling parallel and distributed transaction execution through a simultaneous block-level transaction approach, the proposed technique substantially improves transaction throughput and reduces network latency. The study employs a hybrid shard generation algorithm within the Geth node of the blockchain network to create multiple shards or partitions. Experimental results indicate promising outcomes, with miners experiencing a remarkable speedup of 1.91x and validators achieving 1.90x, along with a substantial 35.34% reduction in network latency. These findings provide valuable insights and offer scalable solutions, empowering researchers and practitioners to address scalability concerns and promoting broader adoption of blockchain technology across various industries.**

***Keywords*—Ethereum; shard generation; data partitioning; proof of work**

I. INTRODUCTION

Blockchain technology has emerged as a groundbreaking innovation with transformative potential across various industries, revolutionizing the way we conduct transactions and manage data. The decentralized and tamper-resistant nature of blockchain networks, first introduced with Bit coin in 2008 by the enigmatic Satoshi Nakamoto, has paved the way for new applications beyond crypto currencies, such as supply chain management, healthcare, finance, and more. However, as blockchain networks continue to gain traction and see widespread adoption, a critical challenge looms large: scalability. Scalability is a pivotal concern in the broader implementation of blockchain systems [1]. As the number of participants (network nodes) and the volume of transactions grow exponentially, traditional blockchain architectures face limitations in accommodating the increasing demands on their resources and processing capabilities. This has led to performance bottlenecks, increased network latency, and limited transaction throughput, hindering the seamless scalability of blockchain networks. To address this challenge, this paper presents a novel and comprehensive solution that leverages a combination of hybrid shard generation and data partitioning techniques. The primary objective of this approach is to enhance the scalability of blockchain architectures while preserving the core principles of decentralization, security, transparency, and immutability [2].

Sharding techniques and data partitioning have emerged as promising solutions to address scalability challenges in blockchain networks. By effectively distributing the workload and data across multiple shards or partitions, these techniques aim to enhance transaction throughput and overall network performance [3]. However, implementing sharding and data partitioning in blockchain networks comes with its own set of challenges. Ensuring consensus across multiple shards, facilitating secure cross-shard communication, and maintaining data synchronization are critical considerations.

The proposed solution begins with the introduction of a groundbreaking hybrid shard generation algorithm, inspired by the concept of sharding in database technology. This algorithm intelligently creates multiple shards within the blockchain network, each functioning as an independent blockchain with its own set of verified users and data. By strategically distributing the transaction load among these shards, we aim to optimize resource utilization and improve overall system performance, facilitating seamless scalability. Moreover, to further enhance the efficiency of the blockchain network, we employ a static analysis-driven data partitioning technique. This innovative approach enables the execution of transactions in parallel and distributed fashion across different shards, reducing contention and enhancing coordination among network nodes. By effectively partitioning data and workloads, we mitigate performance bottlenecks, promoting smoother transaction processing and improved scalability [4].

The combination of hybrid shard generation and data partitioning forms a powerful and synergistic approach to tackle the scalability challenge in blockchain networks. By enhancing transaction throughput, reducing network latency, and efficiently utilizing network resources, our solution aims to pave the way for the broader adoption of blockchain technology across diverse industries. The primary goal of this study is to implement the sharding technique in a blockchain network to improve transaction throughput and reduce network latency. By distributing the workload across multiple shards, the aim is to achieve higher transaction processing capacity and faster transaction confirmation times.

The two significant contributions to enhance the scalability of blockchain networks are:

- A novel hybrid shard generation algorithm is introduced, creating multiple shards within the network, each functioning as an independent blockchain with its set of verified users and data. The algorithm strategically distributes transaction load among these shards, optimizing resource utilization and improving system performance.
- A static analysis-driven data partitioning technique is proposed, enabling parallel transaction execution across shards, reducing contention and enhancing efficiency.

The paper is organized with discussions on related work in Section II, followed by the detailed design of the hybrid shard generation algorithm in Section III, explanation of the static analysis-based data partitioning technique in Section IV, and an experimental evaluation in Section V, showcasing impressive speedups for miners and validators and reduced network latency. A comparison with existing solutions highlights the advantages and effectiveness of the proposed approach. Section VI concludes the study.

II. Related Work

This section provides a comprehensive review of the existing literature and research on blockchain scalability, shedding light on the current state of the field and the various approaches proposed to tackle scalability challenges. The related work encompasses studies on sharding techniques, data partitioning, and other scalability solutions tailored for blockchain networks. Satoshi Nakamoto's introduction of blockchain in 2008 [5] numerous industries such as finance, healthcare, supply chain, and real estate have experienced significant benefits from its innovative capabilities. However, both traditional distributed databases and blockchain systems exhibit distinct failure modes, as highlighted in [6]. To address this, a shard formation mechanism has been developed to facilitate the creation of efficient shards for parallel processing. The process of mining, being laborious and computationally intensive, has led to the emergence of distributed mining pools in well-known blockchain systems like Bitcoin and Ethereum. These pools harness distributed computational power to collectively identify the Proof of Work (PoW) and distribute rewards based on pre-established protocols [7]. Notably, the concept of dynamic blockchain sharding has been proposed, enabling the blockchain to alter its sharding configuration in real-time without requiring hard forks. An exemplary implementation of dynamic blockchain sharding is found in RapidChain.

Yizhong Liu et al. [8] present a thorough investigation of sharding in blockchain systems, with a focus on fundamental concepts, diverse approaches, and the challenges related to blockchain scalability. Their study offers a comprehensive overview of the essential building blocks of sharding solutions, with a particular emphasis on partitioning strategies, consensus mechanisms, and data management. Hung Dang et al. [9] have made significant progress in extending sharding to permissioned blockchain systems, emphasizing its potential as a solution to address scalability challenges in blockchain networks. Their primary goal is to develop a blockchain system capable of accommodating large network size comparable to major crypto currencies like Bitcoin and Ethereum. By doing so, they aim to overcome the limitations of existing blockchain networks, which are often confined to crypto currency applications and struggle to scale consensus protocols for handling average workloads comparable to centralized processing systems[10].

Mahdi Zamani et al. [11] introduce Rapid Chain as a robust and Byzantine-resilient public blockchain protocol. By employing full sharding, Rapid Chain efficiently partitions network nodes into multiple committees, enabling parallel processing of disjoint blocks of transactions and maintaining separate ledgers. This sharding approach brings significant advancements in scalability for blockchain systems. The authors further conduct a comprehensive performance comparison, pitting Rapid Chain against other state-of-the-art sharding-based protocols like Elastico and Omniledger [12]. The evaluation showcases RapidChain's smooth scalability, effectively supporting network sizes of up to 4000 nodes. Deepal Tennakoon et al. [13] introduce a novel dynamic blockchain sharding protocol, which offers advanced capabilities such as creating new shards, adjusting existing shards, and rotating shard participants. This dynamic approach addresses the limitations of traditional blockchain sharding protocols that lacked the flexibility to modify the number of shards used. The authors also emphasize the importance of security during shard creation to prevent malicious nodes from taking control of shards [14][16]. To counter bribery attempts, the paper proposes a shard committee rotation approach through transaction mapping. The performance evaluation of the solution on the CollaChain blockchain demonstrates quasi-linear scalability, highlighting its effectiveness in handling an increasing number of shards [18][19]. A comprehensive and systematic study of sharding techniques in blockchain systems. They extensively examine the key components of sharding schemes and the major challenges associated with each component. The paper thoroughly discusses various methods to generate epoch randomness and techniques for handling cross-sharding transactions [21]. This study provides valuable insights into the state-of-the-art in sharding on blockchain, contributing to a deeper understanding of this important scalability solution [15][17].

III. Design of Hybrid Shard Generation Algorithm

The design of hybrid shard generation algorithm is a novel approach aimed at creating efficient and scalable blockchain networks through the intelligent generation of multiple shards. Hybrid shard generation algorithm is derived through a combination of concepts from traditional sharding techniques and data partitioning strategies, with the aim of optimizing blockchain network scalability. The algorithm follows a systematic process to intelligently create multiple shards within the network, ensuring efficient transaction processing and resource utilization. It is derived through a combination of concepts from traditional sharding techniques and data partitioning strategies, with the aim of optimizing blockchain network scalability [20]. Here is a detailed description of the key steps involved in the hybrid shard generation algorithm:

A. Shard Size Determination

The first step in the algorithm is to determine the ideal size of each shard. This is based on various factors, such as network capacity, computational power, and desired transaction throughput. The goal is to find a shard size that allows for efficient processing of transactions within each shard without causing performance bottlenecks.

B. High-Volume Transaction Identification

The technique analyzes the transaction data to identify high-volume transactions and frequently accessed smart contracts. These high-volume transactions are crucial for shard creation, as they form the basis for creating initial shards.

C. Shard Creation

Shard creation is formed using the identified high-volume transactions and relevant smart contracts, the technique forms initial shards. Each initial shard is designed to handle specific types of transactions efficiently. The technique assigns a unique identifier to each shard for easy reference.

D. Load Balancing

Once the shards are created the load balancing technique analyzes the transaction load on each shard. The goal is to evenly distribute the workload among the shards to optimize resource usage and avoid overloading any specific shard. If there is an imbalance in the transaction distribution, load balancing mechanisms are employed to address it.

E. Dynamic Sharding

One of the key features of the Hybrid Shard Generation Technique is its dynamic sharding capability. This means that the technique can adjust the number of shards in real-time based on changing network conditions and transaction demands. This adaptability allows the blockchain network to scale efficiently as the transaction load fluctuates.

F. Consensus Mechanism Selection

Once the shards are formed the consensus mechanism is applied on each shard, this mechanism defines an appropriate consensus mechanism that suits the specific requirements and characteristics of the transactions processed within that shard. Different shards may use different consensus protocols, depending on their unique needs.

G. Data Partitioning

Data partitioning strategies are implemented to ensure that relevant data is stored within each shard. The aim is to minimize the need for frequent cross-shard communication, as this can impact the overall performance of the blockchain network. Data is partitioned in a way that reduces data access across different shards.

H. Communication Protocol Establishment

Blockchain transactions may require interactions between different shards, the technique establishes a communication protocol to facilitate cross-shard transactions when necessary. This communication protocol ensures secure and efficient communication between the shards.

I. Security Measures

To protect the security and integrity of the blockchain network, the technique incorporates robust security measures. These measures are designed to prevent shard takeovers by malicious nodes and safeguard the overall security and decentralization of the network.

The hybrid shard generation algorithm continuously optimizes the shard configuration, load balancing, and consensus mechanisms based on the dynamic nature of the network. It adapts to changing transaction demands, ensuring that the blockchain network can efficiently scale while maintaining security and performance [16]. The result of the algorithm is a set of optimized shards that work collaboratively to achieve enhanced scalability, increased transaction throughput, and reduced network latency. Hybrid Shard Generation Algorithm is a technique used to intelligently create multiple shards within a blockchain network, optimizing transaction distribution and load balancing to enhance scalability.

The Parameters considered for the design of hybrid shard generation algorithm are as follows [24]:

- Total Number of Shards: This parameter defines how many shards will be created in the blockchain network. The number of shards affects the overall network capacity and scalability.
- Shard Size: Each shard's size determines the number of transactions it can accommodate. Smaller shard sizes might lead to more efficient processing, but could also introduce overhead due to the increased number of shards.
- Hybrid Approach Ratio: If the algorithm combines different shard generation approaches (e.g., static and dynamic), this parameter might define the proportion of each approach to use.
- Dynamic Thresholds: If dynamic shard generation is used, parameters related to the thresholds for triggering the creation or merging of shards might be defined.
- Data Partitioning Criteria: Parameters related to how transactions are partitioned into different shards, such as transaction attributes, geographical location, or other relevant factors.
- Security and Consensus Parameters: Depending on the consensus mechanism used in the blockchain network (e.g., Proof of Work, Proof of Stake), there might be parameters related to security, validation, and consensus that impact shard generation.
- Load Balancing Criteria: Parameters related to load balancing across shards to ensure even distribution of transactions and computational resources.
- Adaptability Parameters: Parameters that determine how the system adapts to changing conditions, such as variations in transaction volume or network size.

A high-level description of the Hybrid Shard Generation Algorithm and its pseudocode:

Algorithm: Hybrid shard generation algorithm

Input:
Total number of nodes in the blockchain network (N)
Desired number of shards (S)
Sharding criteria and parameters (e.g., transaction load, user verification)

Output:
List of shards with their assigned nodes

Procedure:

a. Calculate the number of nodes per shard (Nodes_Per_Shard = N / S).
b. Initialize an empty list to store shards and their assigned nodes (Shard_Assignment).
c. For each shard (i = 1 to S), perform the following steps:
   i. Create a new shard (Shard_i).
   ii. Select Nodes_Per_Shard nodes randomly from the blockchain network and assign them to Shard_i.
   iii. Add Shard_i with its assigned nodes to the Shard_Assignment list.
d. If there are any remaining nodes after all shards have been created:
   i. Assign the remaining nodes to existing shards to balance the load (e.g., round-robin fashion).
   ii. Update the Shard_Assignment list accordingly.
e. Return the Shard_Assignment list as the final result.

```
def hybrid_shard_generation(N, S):
 # Calculate the number of nodes per shard
    nodes_per_shard = N // S
# Initialize an empty list to store shards and their assigned
   nodes
      Shard_Assignment = [ ]
# Create shards and assign nodes
       for i in range(S):
       shard_i = "Shard_" + str(i+1)
 assigned_nodes=randomly_select_nodes(N,nodes_per_shard)
 shard_assignment.append ((shard_i, assigned_nodes))
# Handle any remaining nodes
       remaining nodes = N % S
       for i in range(remaining_nodes):
       Shard_Assignment[i][1].append(Nodes[i])
       return Shard_Assignment
```

IV. Static Analysis-based Data Partioning

Static Analysis-Based Data Partitioning is an innovative technique designed to enhance the efficiency and scalability of blockchain networks by optimizing data distribution and transaction execution across different shards. Unlike traditional data partitioning methods that rely on runtime analysis, this approach utilizes static analysis to pre-determine data partitioning strategies, enabling more effective workload distribution and minimizing contention among network nodes. Here is a detailed description of the Static Analysis-Based Data Partitioning technique

A. Data Analysis

The first step in this technique involves a comprehensive analysis of the blockchain data. The goal is to identify data patterns, dependencies, and access frequency for various transactions and smart contracts. Static analysis tools are employed to analyze the code and data structures within the blockchain network.

B. Dependency Graph Generation

Based on the data analysis, a dependency graph is generated. This graph represents the relationships between different data elements and their dependencies on each other. The dependency graph provides insights into how data should be logically grouped and distributed across different shards.

C. Workload Estimation

This technique estimates the workload for each shard based on the transactions and smart contracts that are likely to be executed within each shard. This workload estimation helps in ensuring that each shard is appropriately sized to handle its share of the transactions.

D. Transaction Grouping

Transaction grouping technique groups related transactions together based on their dependencies and access patterns. Transactions that frequently interact with the same data elements are grouped together to reduce the need for cross-shard communication.

E. Shard Allocation

Once transaction grouping operation is performed, the data is partitioned and allocated to specific shards based on the dependencies and workload estimation. The goal is to minimize data access across different shards, thereby reducing contention and enhancing transaction processing speed.

F. Parallel Transaction Execution

Once the data is partitioned and allocated to the shards, parallel transaction execution process is carried out within each shard. Transactions within the same shard can be executed concurrently, optimizing resource utilization and reducing transaction processing time.

G. Cross-Shard Communication Optimization

Although the focus is on minimizing cross-shard communication, some transactions may still require interactions with data in other shards. To optimize such communication, the technique employs efficient communication protocols and algorithms, reducing latency and ensuring secure and timely cross-shard interactions.

H. Scalability Evaluation

The performance of the blockchain network with Static Analysis-Based Data Partitioning is evaluated through experiments and simulations. The goal is to assess the scalability improvements achieved by this technique and compare it with other data partitioning approaches.

Static Analysis-Based Data Partitioning is a technique that optimizes the distribution of data and workloads across different shards in a blockchain network. The goal is to

improve the efficiency of transaction processing and reduce contention among network nodes. Static Analysis-based Data Partitioning offers a proactive and efficient approach to data distribution and transaction execution in blockchain networks. By leveraging static analysis tools and pre-determined partitioning strategies, it minimizes performance bottlenecks, reduces contention, and enhances overall system performance, contributing to the seamless scalability and improved efficiency of the blockchain network [25].

Here is the Pseudocode and algorithm for Static Analysis-Based Data Partitioning:

Algorithm: Static Analysis-Based Data Partitioning

Input:
Transaction pool: List of pending transactions in the blockchain network.
Shards: List of shards in the blockchain network.
Static analysis data: Information about the workload and data distribution of each shard.

Output:
Partitioned transactions: Transactions distributed across different shards based on static analysis.

Procedure:
a. Initialize an empty dictionary to store the workload estimation for each shard.
b. For each shard in the Shards list, perform the following steps:
   i. Calculate the estimated workload for the shard using static analysis data, such as the number of pending transactions and their complexity.
   ii. Store the estimated workload in the dictionary.
c. Sort the shards in ascending order based on their estimated workload.
d. For each transaction in the Transaction pool, perform the following steps:
   i. Assign the transaction to the shard with the lowest estimated workload.
   ii. Update the estimated workload for that shard in the dictionary.
e. Return the partitioned transactions, which are now distributed across different shards based on static analysis.

```
# Initialize an empty dictionary to store the workload
  estimation for each shard
  workload_estimation = { }

# Calculate the estimated workload for each shard using static
  analysis data for shard in Shards:
  workload = calculate_workload_estimate(shard)
  workload_estimation[shard] = workload

# Sort the shards in ascending order based on their estimated
  workload
  sorted_shards=sort_shards_by_workload
  (workload_estimation)
# Initialize an empty dictionary to store the partitioned
  transactions
  partitioned_transactions = { }
# Assign transactions to shards based on workload
  for transaction in Transaction_pool:
  shard_with_lowest_workload = sorted_shards[ ]
  partitioned_transactions[transaction]=shard_with_lowest_w
  orkload
  workload_estimation[shard_with_lowest_workload] + = 1
sorted_shards=sort_shards_by_workload(workload_estimatio
n)
# Re-sort the shards after workload update
# Return the partitioned transactions
    return partitioned_transactions
```

V. Results and Analysis

The experiment is carried out using historical transaction data from the Ethereum blockchain, obtained from Google's Bigquery Engine's public-data archive. It's important to note that the proposed hybrid shard generation algorithm might indeed be better suited to some types of data. Blockchain networks often deal with diverse use cases, such as financial transactions, supply chain data, IoT data, and more. A hybrid approach might be optimized for specific types of transactions or use cases, making it perform better in scenarios that align with its design goals. The dataset comprised approximately 5 million transactions distributed across 80,000 blocks. Within the dataset, two types of transactions were considered: monetary transactions, involving simple and low-latency transfers of value, and contractual transactions, which required the execution of smart contracts and took longer to process. Hyperledger Caliper is a benchmarking tool specifically designed to measure the performance of blockchain systems, including transaction scalability. It supports various blockchain platforms, including Ethereum and Hyperledger Fabric, which makes it suitable for evaluating the proposed technique in different blockchain environments.

To evaluate the proposed scalable blockchain architecture, a workload was created by varying the ratios of smart contract and monetary transactions within each block. Different ratios, such as {1/1, 1/2, 1/4, 1/8, 1/16}, were used to represent different transaction scenarios. Each block in the workload contained 140 to 560 transactions based on the specified ratios. The execution setups involved both serial execution and parallel execution with multiple slave setups.

A. Performance Evaluation Metrics

The key metrics considered for assessing the effectiveness of the proposed Hybrid Shard Generation Algorithm and Static Analysis-Based Data Partitioning technique are:

- Transaction Throughput: This metric quantified the number of transactions processed per second by the blockchain network. A higher transaction throughput indicates better efficiency in handling a larger volume of transactions within a given time frame. Improving transaction throughput is a crucial aspect of enhancing the scalability of blockchain networks [22].

- Network Latency: Network latency refers to the time taken for a transaction to be propagated and confirmed across the blockchain network. It was measured as the end-to-end block generation time, which indicates how quickly new blocks are created and validated. Lower network latency signifies faster confirmation and validation of transactions, contributing to a more responsive and efficient blockchain system [23].

These performance metrics are fundamental in evaluating the proposed technique's impact on the scalability of the blockchain network. By analyzing transaction throughput and network latency, The aim is to demonstrate improvements in transaction processing speed and reduced network latency, leading to a more scalable and responsive blockchain architecture.

Fig. 1 show the average transaction execution time by the miner in relation to the number of shards generated, ranging from 1 to 5. As the number of shards increases, the transaction execution time decreases, measured in milliseconds (ms). This reduction in transaction execution time is attributed to the increased transaction processing efficiency within a block, resulting in enhanced transaction throughput. A larger number of shards in a block enable higher levels of parallelism, facilitating faster transaction processing and contributing to improved overall system performance.

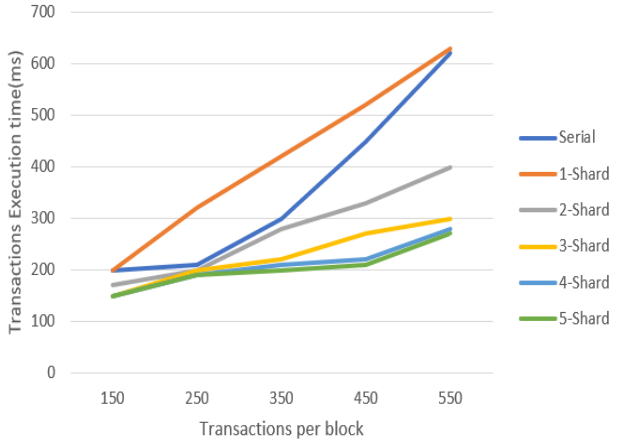

Fig. 1. Average transaction execution time by miner.

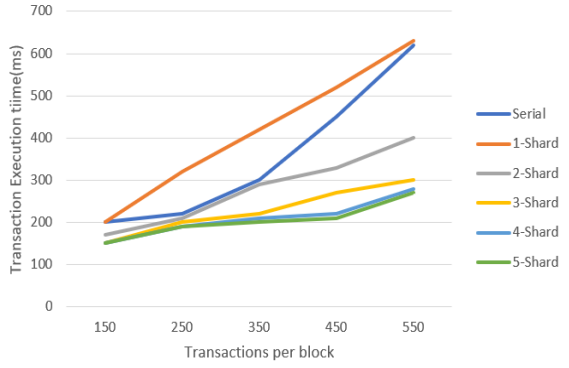

Fig. 2. Average transaction execution time by validator.

Fig. 2 illustrates the average transaction execution time by the validator as the number of shards generated increases from 1 to 5. The transaction execution time, measured in milliseconds (ms), decreases with a higher number of shards. This decrease in transaction execution time is attributed to the increased transaction processing efficiency within a block, resulting in improved transaction throughput. As the number of shards in a block increases, the transaction processing of the transactions within that block becomes more efficient, enabling greater parallelism. This enhancement in parallel processing contributes to the overall improvement in transaction throughput.

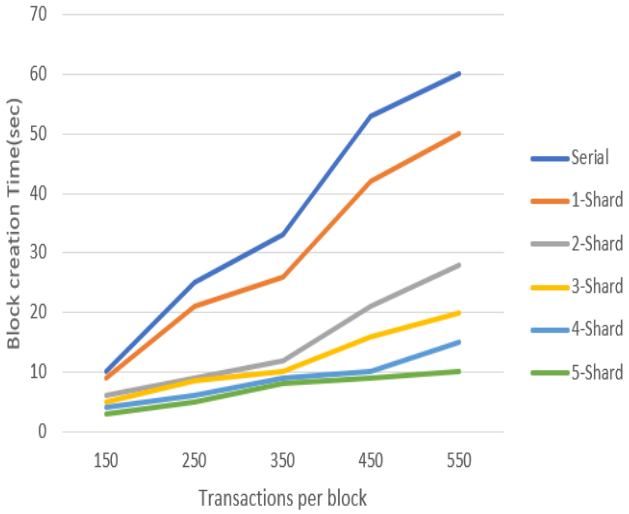

Fig. 3. Average end-to-end block creation time by miner (with mining).

Fig. 3 displays the average end-to-end block creation time by the miner (with mining) as the number of shards increases. As the number of shards rises, the block creation time, which refers to the time taken for a block to be accepted and added to the ledger, decreases. This reduction in block creation time is attributed to the efficient processing of more transactions at a faster rate when more shards are utilized. Consequently, the adoption of multiple shards leads to shorter block creation times, as the transactions are processed more rapidly and added to the blockchain in a more efficient manner.

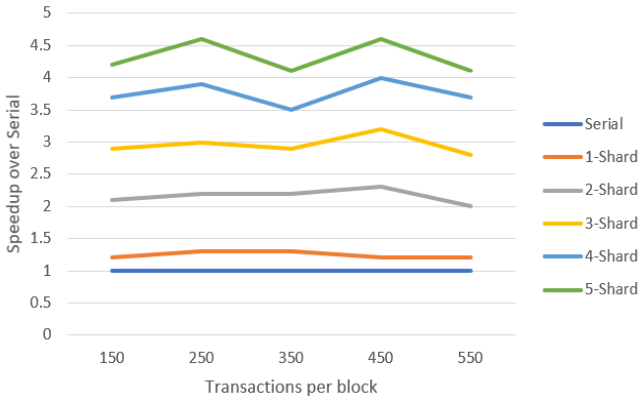

Fig. 4. Speedup of parallel mining over serial mining for average block creation.

Fig. 4 illustrates the speedup of parallel mining over serial mining for average block creation. Speedup represents the increase in transaction throughput compared to the baseline of serial execution, which is assigned a value of 1x. As the number of shards increase, the speedup observed becomes more significant, indicating a higher enhancement in transaction throughput. This increase in speedup is attributed to the efficient parallel processing of transactions achieved with the greater number of shards. Consequently, the adoption of more shards leads to a substantial boost in transaction throughput, demonstrating the effectiveness of the proposed approach in improving the overall performance of the blockchain system.

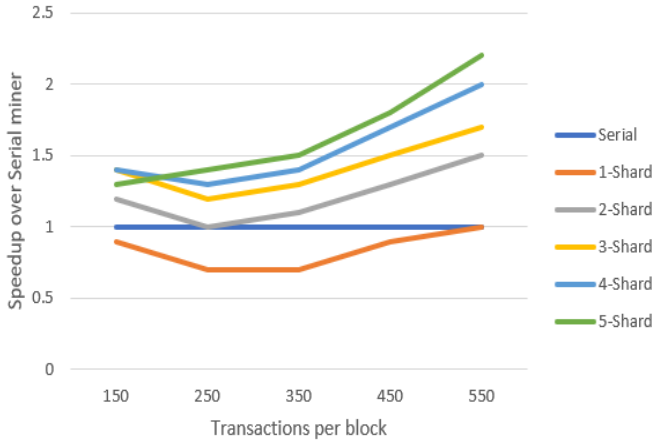

Fig. 5. Miner average speed increase (without mining) for transaction execution.

Fig. 5 displays the miner's average speed increase (without mining) for transaction execution. Speedup represents the rise in transaction throughput when compared to the baseline of serial execution, where the serial miner execution is assigned a value of 1x. As the number of shards increases, the observed speedup becomes more significant, indicating a higher enhancement in transaction throughput. Notably, the 5-shard architecture offers the highest speedup over the serial miner, suggesting that the adoption of a 5-shard configuration provides the most substantial improvement in transaction processing efficiency. This finding underscores the effectiveness of the proposed technique in optimizing the performance of the blockchain system and achieving higher transaction throughput.

Fig. 6 illustrates the validator's average speed increase for transaction execution. Speedup represents the rise in transaction throughput when compared to the baseline of serial execution, where the serial validator execution is assigned a value of 1x. As the number of shards increases, the observed speedup becomes more significant, indicating a higher enhancement in transaction throughput. Remarkably, the 5-shard architecture offers the highest speedup over the serial validator, signifying that the adoption of a 5-shard configuration provides the most substantial improvement in transaction processing efficiency for validators. This result further reinforces the efficacy of the proposed technique in optimizing the performance of the blockchain system and achieving higher transaction throughput for validators.

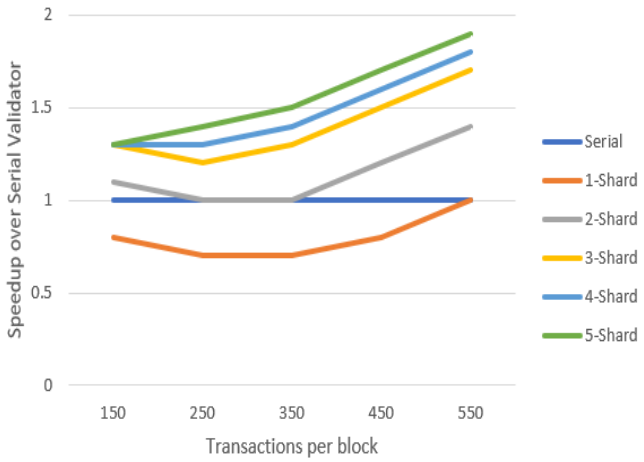

Fig. 6. Validator average speed increase for transaction execution.

B. Performance Analysis and Discussion

Table I presents the estimated workload speedup achieved with validators and miners (without mining) for different numbers of slaves in the system. The speedup values indicate how much faster the workload is executed when both validators and miners work concurrently compared to sequential execution (baseline). The table also provides the total number of transactions executed per block for each workload scenario. In the baseline scenario of serial execution, both miners and validators achieve a speedup of 1.00, which means no improvement in transaction throughput compared to sequential execution.

TABLE I. ESTIMATED WORKLOAD SPEEDUP WITH VALIDATOR AND MINER (WITHOUT MINING)

Workload (Average Speedup)		Total number of transactions executed per block				
		100	200	300	400	500
Serial Execution	Miner	1.00	1.00	1.00	1.00	1.00
	Validator	1.00	1.00	1.00	1.00	1.00
1 Slave	Miner	0.82	0.75	0.76	0.86	0.91
	Validator	0.78	0.71	0.69	0.82	0.89
2 Slaves	Miner	1.12	1.01	1.26	1.28	1.32
	Validator	1.03	1.05	1.22	1.24	1.45
3 Slaves	Miner	1.32	1.38	1.40	1.45	1.54
	Validator	1.05	1.34	1.56	1.65	1.72
4 Slaves	Miner	1.42	1.48	1.51	1.58	1.62
	Validator	1.21	1.28	1.31	1.52	1.72
5 Slaves	Miner	1.47	1.58	1.62	1.70	1.82
	Validator	1.28	1.29	1.36	1.79	1.89

The average speedup of miner and validator nodes for varying number of slave nodes as follows:

- 1 Slave: When using a single slave, both miners and validators experience a speedup of less than 1.00. This indicates that concurrent execution with only one slave is slower than sequential execution. The workload is not efficiently distributed among the participants.

- 2 Slaves: With two slaves, both miners and validators show a speedup greater than 1.00. This demonstrates that concurrent execution with two slaves is more efficient than sequential execution. The workload is better distributed, resulting in improved transaction throughput.
- 3 Slaves: As the number of slave increases to three, the speedup for both miners and validators further improves. This indicates that concurrent execution with three slaves is even more efficient in processing transactions compared to sequential execution.
- 4 Slaves: With four slaves, the speedup values continue to increase for both miners and validators, indicating a higher level of efficiency in parallel processing.
- 5 Slaves: The highest speedup values are observed when using five slaves for both miners and validators. This indicates that concurrent execution with five slaves offers the best transaction throughput improvement compared to sequential execution.

The results of the evaluation showcased the speedups achieved by miners and validators using the proposed scalable blockchain architecture. The architecture demonstrated significant improvements in transaction throughput compared to traditional serial execution. Additionally, network latency was reduced, indicating enhanced efficiency in block generation and transaction processing. To further emphasize the advantages and effectiveness of the proposed approach, a comparison with existing solutions was provided. The evaluation demonstrated that the Hybrid Shard Generation Algorithm and Static Analysis-based Data Partitioning technique outperformed conventional blockchain approaches, highlighting its potential to address scalability challenges effectively.

The results of the evaluation showcased the speedups achieved by miners and validators using the proposed scalable blockchain architecture. The architecture demonstrated significant improvements in transaction throughput compared to traditional serial execution. Additionally, network latency was reduced, indicating enhanced efficiency in block generation and transaction processing. To further emphasize the advantages and effectiveness of the proposed approach, a comparison with existing solutions was provided. The evaluation demonstrated that the Hybrid Shard Generation Algorithm and Static Analysis-Based Data Partitioning technique outperformed conventional blockchain approaches, highlighting its potential to address scalability challenges effectively. Overall, the results suggest that as the number of slaves increases in the system, the workload speedup also improves significantly. This demonstrates the benefits of parallel processing and how it enhances transaction throughput when validators and miners work concurrently in the blockchain network. The use of more slaves allows for better workload distribution and improved performance, contributing to the overall scalability and efficiency of the blockchain system.

VI. Conclusion

The proposed scalable blockchain architecture presents a significant step forward in addressing the critical challenge of scalability in blockchain networks. By leveraging the Hybrid Shard Generation Algorithm and Static Analysis-based Data Partitioning technique, the architecture effectively enhances transaction throughput and reduces network latency. The experimental evaluation using a blockchain simulation tool validates the effectiveness of these solutions, demonstrating notable improvements in performance. The results of the evaluation reveal that as the number of shards generated increases, transaction execution times by both miners and validators decrease, leading to improved transaction throughput. The speedup achieved by parallel mining over serial mining also increases with the rise in the number of shards, highlighting the advantages of parallel processing and efficient workload distribution. Moreover, the comparative study analysis emphasizes the significance of concurrent execution with multiple slaves, which substantially improves workload speedup for both miners and validators. This underscores the importance of parallel processing in achieving higher transaction throughput and overall network efficiency. The findings of the paper provide valuable insights into the potential of Hybrid Shard Generation and Static Analysis-Based Data Partitioning techniques in addressing scalability limitations in traditional blockchain platforms. These techniques offer practical solutions that make blockchain networks more efficient and adaptable for diverse applications in various industries. As research in this area continues to evolve, it is expected that these innovative techniques will play a pivotal role in shaping the future of blockchain networks, enabling them to handle increasing transaction volumes and meet the demands of an ever-evolving digital landscape.

References

[1] Dumitrel Loghin Ee-Chien Chang Hung Dang, Tien Tuan Anh Dinh. Towards scaling blockchain systems via sharding. In Proceedings of the 2019 International Conference on Management of Data. SIGMOD ’19.

[2] Deepal Tennakoon and Vincent Gramoli. Dynamic blockchain sharding. In 5th International Symposium on Foundations and Applications of Blockchain, 2022.

[3] Marcos Antonio Vaz Salles Yizhong Liua, Jianwei Liua. Building blocks of sharding blockchain systems: Concepts, approaches, and open problems. In Sciencedirect, 2021.

[4] Mark Nixon Song Ha Gang Wang, Zhijie Jerry Shi. Sok: Sharding on blockchain. In Proceedings of the 1st ACM Conference, Zurich, Switzerland, 2019. Advancement in financial technologies.

[5] Mahnush Movahedi Mahdi Zamani 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.

[6] Praveen M Dhulavvagol, S G Totad, Performance Enhancement of Distributed System Using HDFS Federation and Sharding, Procedia Computer Science, Volume 218, 2023, Pages 2830-2841, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2023.01.254.

[7] G. Wood. Ethereum: a secure decentralised generalised transaction ledger. In Ethereum Project YELLOW paper, page 1–32, 2014.

[8] S. Peri S. Rathor P. S. Anjana, S. Kumari and A. Somani. An efficient framework for optimistic concurrent execution of smart contracts. In 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pages 83–92, Feb,2019.

[9] O. Novo. Blockchain meets iot: an architecture for scalable access management in iot. In IEEE Internet of Things Journal, page 1184–1195, 2018.

[10] S. Faust S. Dziembowski and K. Hostáková. General state channel networks. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pages 949–966, Toronto, Canada, October,2018.

[11] Liuyang Ren and A. S. Ward. Transaction placement in sharded blockchains. Waterloo,Canada, 9 Jun,2022. arXiv:2109.07670v3.

[12] Jingjie Jiang Yuechen Tao, Bo Li. On sharding open blockchains with smart contracts. In 2020 IEEE 36th International Conference on Data Engineering (ICDE), page 1357–1368, Dallas, TX, USA, 2020. IEEE.

[13] M. Staples et al. X. Xu, I. Weber. A taxonomy of blockchain-based systems for architecture design. In Proceedings of the 2017 IEEE International Conference on Software Architecture (ICSA), page 243–252, Gothenburg, Sweden, April 2017. IEEE.

[14] Praveen M Dhulavvagol, Vijayakumar H Bhajantri, S G Totad, Blockchain Ethereum Clients Performance Analysis Considering E-Voting Application, Procedia Computer Science, Volume 167, 2020, Pages 2506-2515, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.303.

[15] Chuang Peng Runyu Chen, Lunwen Wang and Rangang Zhu. An effective sharding consensus algorithm for blockchain systems. In Electronics, Heifei,China, 2022.

[16] Primicerio Lin, Jian-Hong and Kevin. Lightning network: a second path towards centralisation of the bitcoin economy. In New Journal of Physics, 2020.

[17] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels,A. Kosba, A. Miller, P. Saxena, E. Shi, and E. G un. On scaling decentralized blockchains. In Proc. 3rd Workshop on Bitcoin and Blockchain Research, 2016.

[18] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A.Kroll, and E. W. Felten. Sok: Research perspectives and challenges for bitcoin an crypto-currencies. In 2015 IEEE Symposium on Security and Privacy, pages 104{121. IEEE, 2015.

[19] PM Dhulavvagol, SG Totad, P Pratheek, Enhancing Transaction Scalability of Blockchain Network Using Sharding. In Soft Computing for Security Applications: Proceedings of ICSCS 2023 1449, 253

[20] G. Zyskind, O. Nathan, and A. S. Pentland, ``Decentralizing privacy: Using blockchain to protect personal data," in Proc. Secur. Privacy Work- shops (SPW), May 2015, pp. 180_184.

[21] L. Luu, V. Narayanan, C. Zhang, K. Baweija, S. Gilbert, and P. Saxena. A secure sharding protocol for open blockchains. In CCS, 2016. [10] Koc et al., "Towards Secure E-Voting Using Ethereum Blockchain", 2018 IEEE 6th International Symposium on Digital Forensic and Security.

[22] Suporn Pongnumkul, Chaiyaphum Siripanpornchana, Suttipong Thajchayapong, "Performance Analysis of Private Blockchain", 26th International Conference on Computer Communication and Networks (ICCCN), 2017.

[23] M. M. Arer, P. M. Dhulavvagol and S. G. Totad, "Efficient Big Data Storage and Retrieval in Distributed Architecture using Blockchain and IPFS," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-6, doi: 10.1109/I2CT54291.2022.9824566.

[24] Yue Hao, Yi Li, Xinghua Dong, Li Fang, Ping Chen, "Performance Analysis of Consensus Algorithm in Private Blockchain", Intelligent Vehicles Symposium (IV) 2018 IEEE, pp. 280-285, 2018.

[25] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, K.-L. Tan, Blockbench: A framework for analyzing private blockchains.