SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V11I7P110

Volume 11 Issue 7, 124-133, July 2024

Original Article

Enhancing Split-Join Blockchain Performance through Load Balancing

Vemula Harish[1], R. Sridevi[2]

[1]*Department of CSE, Jawaharlal Nehru Technological University, Hyderabad, India.*
[2] *Department of CSE, JNTUH College of Engineering, Hyderabad, India.*

[1]*Corresponding Author : vemula.harish31@gmail.com*

Received: 05 May 2024 Revised: 07 June 2024 Accepted: 05 July 2024 Published: 26 July 2024

Abstract - *Blockchain technology is secure, tamper-proof and transparent in nature, but the scalability of blockchain remains a significant challenge due to its decentralized nature. Many researchers and enterprises are actively working on blockchain technology to resolve these issues and optimize performance. As a result, several approaches have been proposed, such as Directed Acyclic Graphs and Sharding mechanisms, to achieve parallelism and high scalability, but still there are many open issues to be addressed. This study explores the importance of load balancing in the Split-Join blockchain framework, which is designed to enhance scalability through parallel block processing. This work analyzes the throughput with and without implementing the load balancing in split-join blockchain, conducting an empirical study of increasing transaction volumes and their impact on blockchain performance. The results show that implementing a load balancer within the blockchain technology framework significantly enhances throughput and reduces processing times, thus proving the capability to enhance the overall performance of blockchain.*

Keywords *- Blockchain technology, Scalability, Performance, Load balancing, Split-join framework.*

1. Introduction

Blockchain is a decentralized technology that offers security through cryptographic hashes, and the transactions that are included in blockchain are immutable. Blockchain was first introduced by Satoshi Nakamoto in Bitcoin [1]. It is the first technology that solves distributed consensus problems through the introduction of Proof of Work (PoW) [2]; later, due to its high computational requirements, several other consensus methods were proposed by the researchers such as Proof of Stake (PoS) [3], Delegated Proof of Stake (DPoS) [4], Proof of Authority (PoA) [5], Proof of Capacity (PoC) [6], Proof of Burn (PoB) [7], Proof of Elapsed Time (PoET) [8], Proof of History (PoH) [9]. Apart from financial applications such as cryptocurrencies blockchain technology can be applicable to vide variety of applications like Supply Chain Management [10], Electronic Voting Systems [11], Patient Health Record Management Systems [12] and Decentralized Finance (DeFi) for verifying digital identities [13].

1.1. Scalability in Blockchain Technology

Blockchain technology is a decentralized peer-to-peer network where each node maintains its ledger that cannot be modified, i.e., Once a transaction is committed to the ledger, it is permanent it is secure enough to make digital transactions. Blockchain is struggling with scalability issues [14, 15] due to its decentralized nature. Scaling a blockchain to a great number of nodes requires processing a huge number of transactions so that it allows many nodes to transact simultaneously across the network. Many businesses are starting to use blockchain technology for their operations. As a result, they need blockchain platforms that can handle a large amount of data and transactions.

1.2. Load Balancing in Blockchain

Load balancing in the blockchain is very crucial, especially for blockchain platforms that support parallel processing of blocks or transactions [16]. To manage the load among the components of the platform, depending on its architecture, there is a need to employ the appropriate load balancing algorithm to handle high volumes of transactions; split-join blockchain supports parallel processing of blocks, which allows blockchain to scale, making it suitable to be adopted by many enterprises. As blockchain networks grow, the volume of transactions also rises, which could result in congestion and less efficiency.

Load Balancing is essential for handling the transaction overload between the mining pools Higher Mining Pool (HMP) and Lower Mining Pool (LMP) of a split-join blockchain framework. The Split-Join blockchain framework utilizes two mining pools for handling parallel blocks of transactions, HMP and LMP. Here, the load balancer is responsible for evenly distributing the workload between the mining pools, ensuring that neither pool becomes overloaded with transactions. This enhances operating efficiency and guarantees transaction verification. Efficient load balancing enhances network processing speed. An efficient blockchain network promotes user trust and lays the foundation for further technological improvements.

2. Literature Review

2.1. Blockchain Structure

Blockchain is also known as Distributed Ledger Technology [17]. It has a sequential chain of blocks that embed transactions in it; blockchain offers security, immutability and transparency and allows business operations in trust less environments through smart contracts [18]. The 'Block' is a basic building block of the blockchain, and it comprises several parameters such as previous blocks hash, Merkle root, nonce, etc. All transactions in the blockchain are protected from tampering through cryptographic hashes which are linked to one another, i.e., in order to modify the transaction details recorded in the blockchain, one should be able to modify all subsequent blocks and have to compute their cryptographic hashes also which is not feasible in real time where original blockchain is operated with the contribution of millions of nodes or network participants over the globe. Thus, it offers high security and data integrity.

The consensus mechanism plays a crucial role in blockchain networks to bring agreement about transaction information by all network participants. Due to this, one cannot deny the transactions made by them at a later point in time. These consensus mechanisms maintain the integrity and security of the blockchain, ensuring that all copies of the distributed ledger are consistent over the globe.

2.2. Scalability Challenges in Blockchain Technology

Blockchain is facing various performance and scalability challenges, primarily causes like consensus mechanisms, decentralized architecture, lack of trust, and governance etc. Popular blockchain platforms such as Bitcoin and Ethereum [19] are very slow in transaction processing, resulting in low performance. A block is a fundamental element of a blockchain. The performance of the blockchain depends on the transaction latency, underlying consensus algorithm, etc. Transaction validation duration is a crucial factor that affects blockchain scalability.

In public blockchains such as Bitcoin, the transactions of a block can be considered as confirmed only if a certain number of following blocks have been appended to the chain. The scalability concerns of blockchain technology have notable consequences, including unreliability during high demand and limitations for applications expecting fast confirmations. It is crucial to tackle these difficulties for the continuing growth and adoption of blockchain technology.

Addressing the scalability issues of blockchain requires re-designing the architectural components such as ledger architecture, consensus methods, etc. The researchers proposed some solutions, but still there are open issues to be addressed efficiently. Zhou, Q et al. [20] surveyed the existing scalability solutions such as Sharding [21] is a technique to divide the network nodes into small independent units and operate. Resulting in high scalability but the complexity grows as chain growth.

Omni Ledger [22] uses the public-randomness protocol to guarantee the atomicity of cross-shard transactions. Monoxide [23] scales the blockchain using Asynchronous Consensus Zones and Directed Acyclic Graphs [24]. These attempts to modify the traditional structure of the blockchain that allows parallel processing of transactions are adopted by blockchains such as Tangle [25].

2.3. Load Balancing

In traditional blockchain platforms, the load balancing does not have much impact as they process the blocks sequentially, but highly required in blockchains that use a DAG kind of consensus algorithm, which allows the processing of the transactions in parallel. Effective load balancing improves the overall performance of the platform; it optimizes resource utilization by assessing each node's capacity and allocating tasks based on this capacity. Modern load-balancing solutions are dynamic, allowing them to adapt to changing conditions in real time. Algorithms like Round Robin [26] and Least Connections [27] play a crucial role in determining load distribution.

3. Methodology

3.1. Structure of Split-Join Blockchain

The Split-Join (SPJ) blockchain has blocks that are linked systematically. Each block in the blockchain contains transaction data, a unique hash, a timestamp, and references to the hash of the previous block. Figure 1 shows the split-join blockchain framework architecture; when a client sends a request, the load balancer handles it, and it forwards transaction requests to both the mining pools Higher Mining Pool (HMP) and Lower Mining Pool (LMP). The split-join ledger is initiated with Genesis block G0. It contains the default configuration of the split-join network. Both mining pools process the blocks simultaneously. i.e., after the genesis block G0 blockchain state is changed to split-state, and two blocks will be mined (hb1, lb1), later is switched to join state again in which a single block only created, this block is known as join block, and it approves the transactions included in parallel blocks.

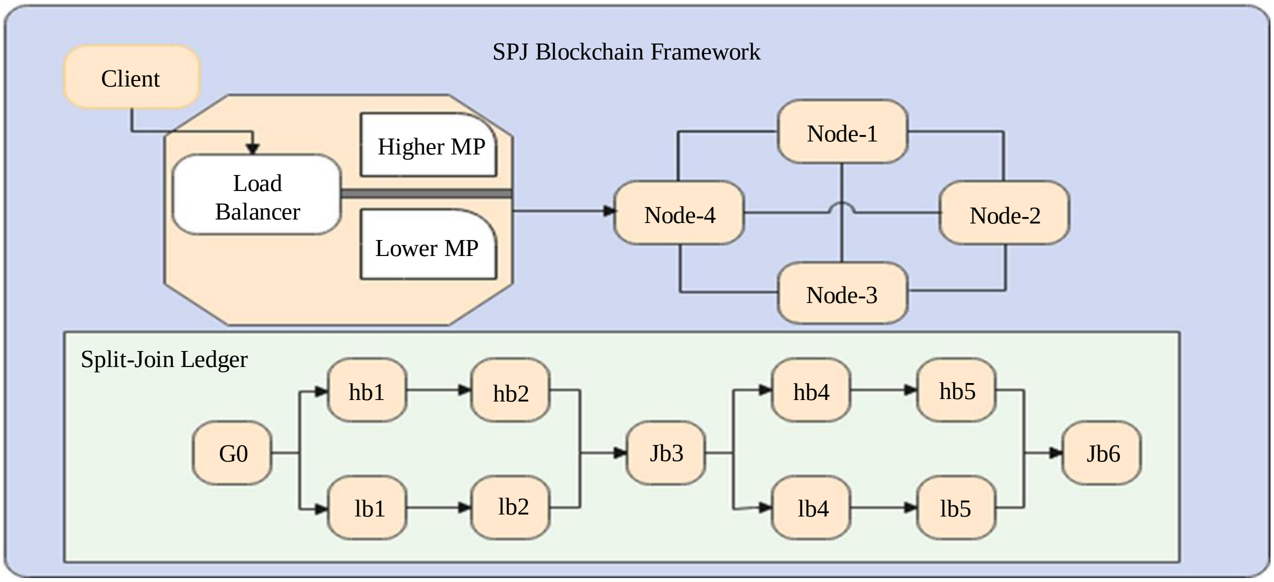

Fig. 1 Split-join blockchain system architecture

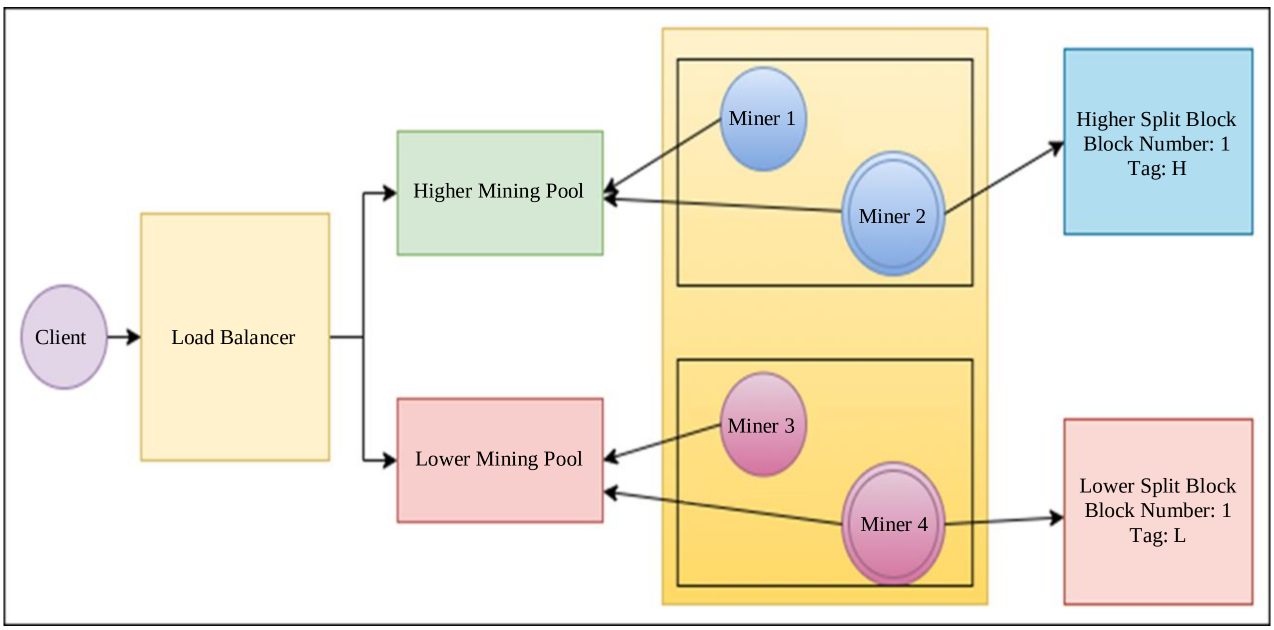

Fig. 2 Load balancing in split-join blockchain architecture

3.2. Mining

Mining in the split-join blockchain is similar to traditional blockchain frameworks. It depends on the underlying consensus algorithm. Miners will be registered with any one of the mining pools and create blocks by collecting the transactions from them.

This framework has two mining pools which operate in parallel to process the blocks simultaneously. All miners registered with HMP will solve the cryptographic puzzles, and one of them will emerge as a winner and publish the block. Similarly, one winner emerged from LMP as well [28].

3.3. Dynamic State Switching in Split-Join Blockchain

Dynamic state switching is introduced in the split-join blockchain framework, which allows the blockchain to handle the block of transactions in parallel. It helps reduce the overall complexity of the blockchain ledger.

Split-join blockchain switches the state between split-state and join-state. Initially, the network is bootstrapped in join-state and switches to split-state after creating the genesis block. While blockchain is in a split state, blocks are created simultaneously, and the number of parallel blocks created depends on the split-chain-length configuration property.

3.3.1. Join-State

In join-state, the split-join blockchain creates a single block called join-block. The join block approves the transactions that are included in the parallel blocks that are created after the previous join block is committed. It prevents the chain from spreading, reduces the complexity and helps in maintaining the simple sequence of blocks.

3.3.2. Split-State

During the split state of the split-join blockchain, blocks are created in parallel. Dynamic state switching enhances the efficiency of the blockchain and helps in the creation of parallel blocks. This enables the blockchain network to process a huge number of transactions within a given time. In addition, it incorporates a dynamic switching mechanism.

3.4. Load Balancer

This study focuses on the significance of the load balancer in a split-join blockchain. The load balancer distributes mining work across the available mining pools. The load balancer optimizes the efficiency of distributing transactions to mining pools and prevents the double spending issue by avoiding sending duplicate transactions across the mining pools.

Figure 2 shows that the client sends transactions to the split-join blockchain, load balancer of the SPJ blockchain directs transactions to either a higher mining pool or a lower mining pool based on the load balancing algorithm incorporated. Each mining pool comprises a set of miners. In the higher mining pool, Miner1 and Miner2 etc., these miners are registered HMP.

Among these miners, whoever first solves the puzzle will be declared as a Winner and as a result new block is created and marked as “Higher Split Block” with “Block Number: 1” and “Tag: H”. Similarly, the lower mining pool includes Miner3 and Miner4, etc., and Miner4 emerged as a ‘Winner’. Resulting in the creation of a new block, “Lower Split Block’’ with “Block Number: 1” and tagged as “Tag: L”. These blocks (L, H”) are created in parallel and added to the distributed ledger.

3.5. Performance Metrics

3.5.1. Total Experiment Duration

This metric measures the total time taken in milliseconds for the single round of experiments from start to finish. It includes the time taken to process all transactions and mine all blocks within the experiment’s scope.

Let T_{total} represent the Total Experiment Duration, measured in milliseconds (ms).

Total Experiment Duration T_{total}, T_{end}, T_{start}, where “T_{start}” is the starting time and “T_{end}” is the ending time, “T_i” represents i^{th} round duration.

Average Duration, $$T_{avg} = \frac{1}{n}\sum_{i=1}^{n} T_i$$

Where,
n = number of rounds, ith round
Ti = Tend, i – T_{start}, I.

3.5.2. Transaction Throughput

Blockchain network performance is measured based on the capacity of the blockchain platform that can be able to process how many transactions per second. It is also called Transactions Per Second (TPS).

$$\text{TPS} = \frac{\text{Total Number of Transactions}}{\text{Total Experiment Duration}}$$

3.5.3. Average Mining Time of HMP (AMTH)

The AMTH represents the simple average time required by the Higher Mining Pool. It is used to understand the transaction load on the mining pool.

$$\text{AMTH} = \frac{\text{Total Mining Time of HMP (TMTH)}}{\text{Number of Blocks Mined by HMP (NBH)}}$$

3.5.4. Average Mining Time of LMP (AMTL)

The AMTL represents the simple average time required by the Lower Mining Pool. It is used to understand the transaction load on the mining pool.

$$\text{AMTL} = \frac{\text{Total Mining Time of LMP (TMTL)}}{\text{Number of Blocks Mined by LMP (NBL)}}$$

4. Results and Discussions

This study is conducted to understand and analyze the impact of the load balancer component on the overall performance of the split-join blockchain framework. The experiment is conducted incrementally, i.e., by increasing the number of transactions and observing its impact. Every experiment is repeated five times to confirm the performance behavior.

4.1. Enhancing Scalability

Load balancing will avoid the performance bottlenecks of overloading a single mining pool such as exhausting one mining pool where the other one is idle. The load balancer component of the split-join framework efficiently handles the heavy transaction arrival rates and maintains an even workload among the mining pools that process the blocks simultaneously.

4.2. Impact of Load Balancing in Split-Join blockchain

The experiment is conducted in an incremental fashion increasing the number of transactions from 10, 50, 100, 500, 1000. and observed the performance metrics to analyze the impact over five rounds. Two key performance indicators were considered: the total duration of the experiment, measured in seconds (s), and the transaction throughput, measured in Transactions Per Second (TPS).

4.2.1. Performance Evaluation with Arrival Rate 10 Transactions for Each Round

Table 1. Performance evaluation with an arrival rate of 10

Rounds	Duration Without LB (s)	Duration with LB (s)	Throughput Without LB (txns/s)	Throughput with LB (txns/s)
Round 1	0.289	0.081	34.542	123.172
Round 2	0.159	0.113	62.878	88.638
Round 3	0.219	0.091	45.565	109.683
Round 4	0.273	0.105	36.579	94.909
Round 5	0.197	0.092	50.648	108.778

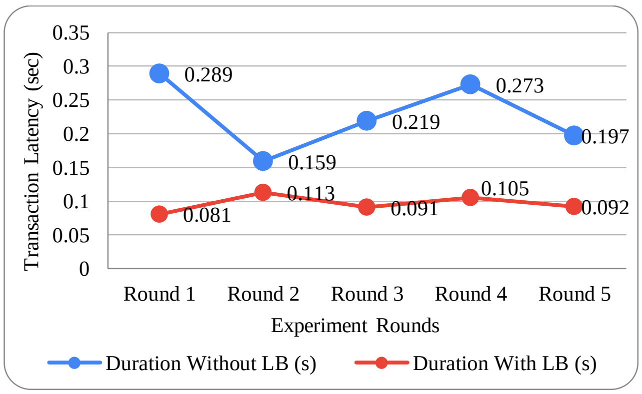

Fig. 3 Transaction latency in sec (10 txns)

Based on the results, including a load balancer into a split-join blockchain system significantly improved the transaction processing times. Ensuring a constant and decreased time of transaction duration is useful for preserving the performance and reliability of the system. Implementing a load balancer appears to resolve the issues, offering a more resilient means of managing transactions in the blockchain.

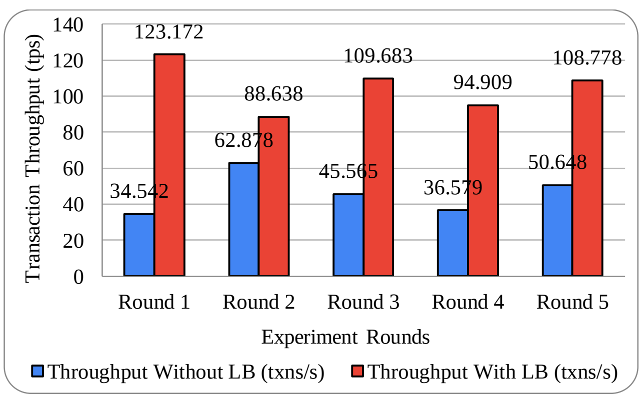

Fig. 4 Transaction throughput in tps (10 txns)

Figures 3 and 4 demonstrate that during all five rounds, the presence of the Load Balancer (LB) consistently results in higher throughput compared to the absence of a Load Balancer. This suggests that the LB has a positive impact on the blockchain's capacity to handle transactions, efficiently controlling the workload of the network.

4.2.2. Performance Evaluation with Arrival Rate 50 Transactions for Each Round

Table 2. Performance evaluation with arrival rate 50 txns

Rounds	Duration Without LB (s)	Duration with LB (s)	Throughput without LB (txns/s)	Throughput with LB (txns/s)
Round 1	0.74	0.24	67.531	208.207
Round 2	1.024	0.417	48.852	119.884
Round 3	0.833	0.307	60.006	163.086
Round 4	0.505	0.272	99.042	183.559
Round 5	0.517	0.457	96.656	109.402

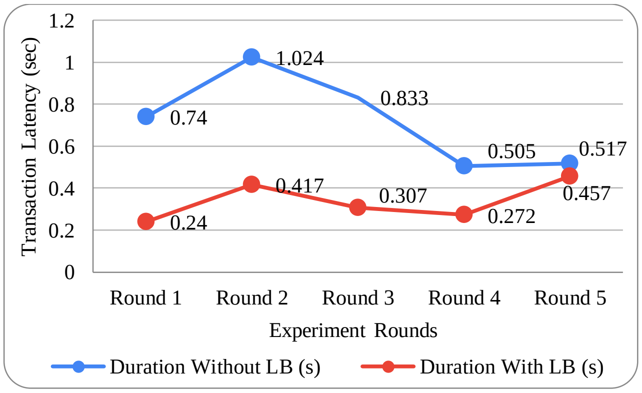

Fig. 5 Transaction latency in sec (50 txns)

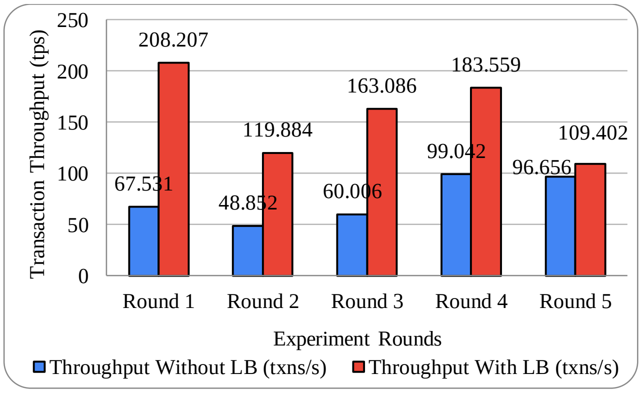

Fig. 6 Transaction throughput in tps (50 txns)

Figures 5 and 6 depict the influence of a load balancer on a transaction arrival rate of 50. Integrating a load balancer significantly improves the effectiveness of the blockchain system. These findings demonstrate that the load balancer successfully manages network traffic, enhances transaction processing, and optimizes workload distribution throughout the network in comparison to a system without a load balancer.

4.2.3. Performance Evaluation with Arrival Rate 100 Transactions for Each Round

Table 3. Performance evaluation with arrival rate 100 txns

Rounds	Duration Without LB (s)	Duration with LB (s)	Throughput without LB (txns/s)	Throughput with LB (txns/s)
Round 1	1.364	0.607	73.283	164.532
Round 2	1.404	0.501	71.184	199.552
Round 3	1.219	0.408	82.007	244.586
Round 4	1.342	0.429	74.509	232.753
Round 5	1.209	0.572	82.645	174.769

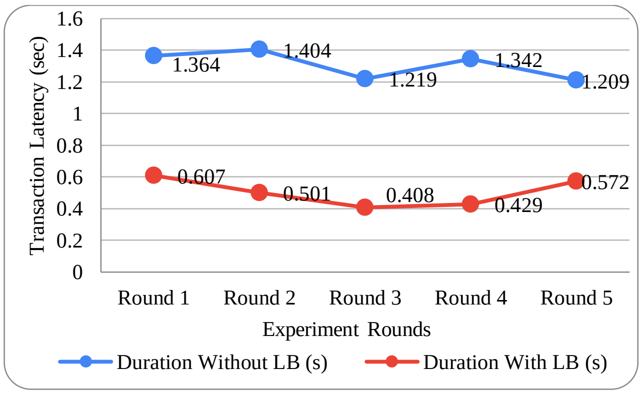

Fig. 7 Transaction latency in sec (100 txns)

Figure 7 demonstrates how the load balancer consistently reduces the overall experiment duration in all rounds, demonstrating enhanced effectiveness and performance for arrival rate 100.

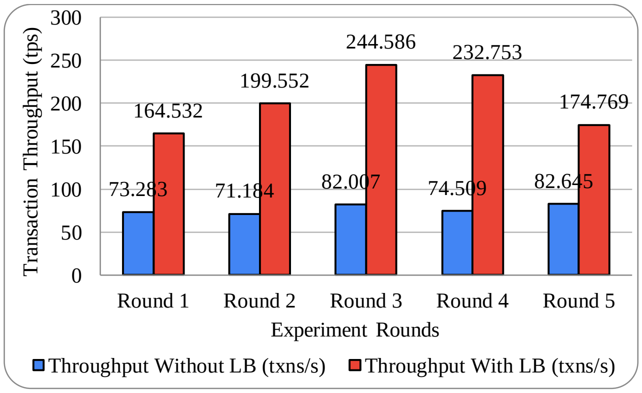

Fig. 8 Transaction throughput in tps (100 txns)

Figure 8 shows that the load balancer has a positive impact on the system's ability to handle transactions on all rounds with a 100 arrival rate.

Table 4. Average mining time per block (HMP, 100txns)

Rounds	Duration without LB (s)	Duration with LB (s)
Round 1	0.0081	0.0039
Round 2	0.0088	0.0031
Round 3	0.0065	0.0029
Round 4	0.0075	0.0029
Round 5	0.0061	0.0037

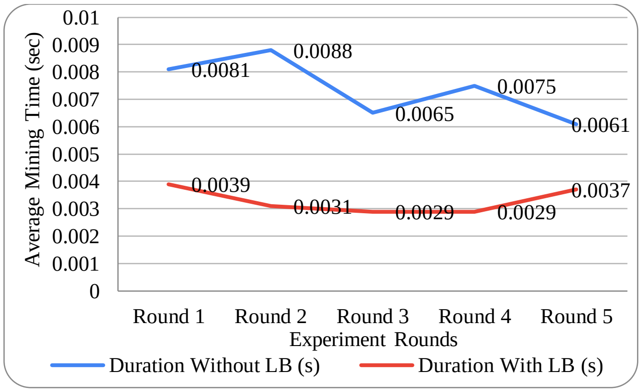

Fig. 9 Average mining time per block (HMP)

Figure 9 compares the average mining time per block of a Higher Mining Pool (HMP) for a system with and without a load balancer across five rounds. The Y-axis represents the time in seconds, and the X-axis represents each round of mining. From the figure, it is clear that the load balancer is exhibiting a lower average time for mining, indicating an improvement in efficiency.

Table 5. Average mining time per block (LMP, 100 txns)

Rounds	Duration without LB (s)	Duration with LB (s)
Round 1	0.0070	0.0042
Round 2	0.0064	0.0037
Round 3	0.0072	0.0027
Round 4	0.0068	0.0028
Round 5	0.0060	0.0036

Figure 10 illustrates the average mining time per block of a Lower Mining Pool (LMP) in a system with and without a load balancer across five rounds. The figure clearly demonstrates that the load balancer shows a reduced average time for mining.

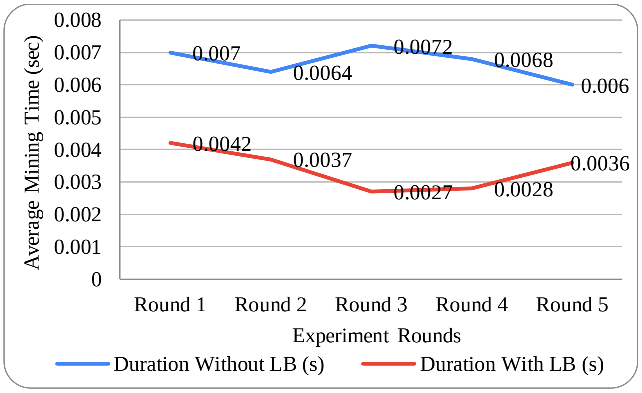

Fig. 10 Average mining time per block (LMP)

4.2.4. Performance Evaluation with Arrival Rate 500 Transactions for Each Round

Table 6. Performance evaluation with an arrival rate of 500

Rounds	Duration Without LB (s)	Duration with LB (s)	Throughput without LB (txns/s)	Throughput with LB (txns/s)
Round 1	6.319	2.650	79.116	188.678
Round 2	7.130	2.219	70.116	225.233
Round 3	7.797	2.175	64.126	229.856
Round 4	7.476	2.250	66.875	222.139
Round 5	6.858	2.193	72.906	227.974

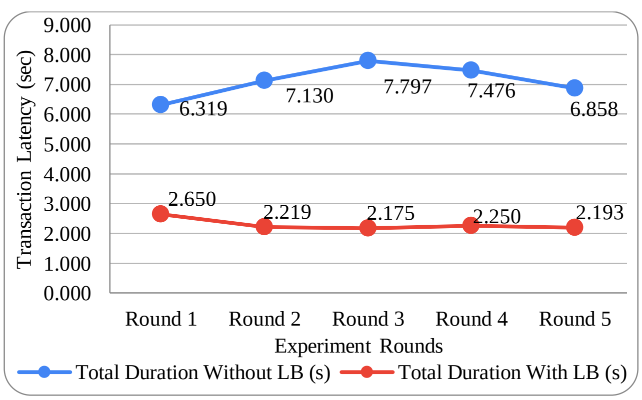

Fig. 11 Transaction latency in sec (500 txns)

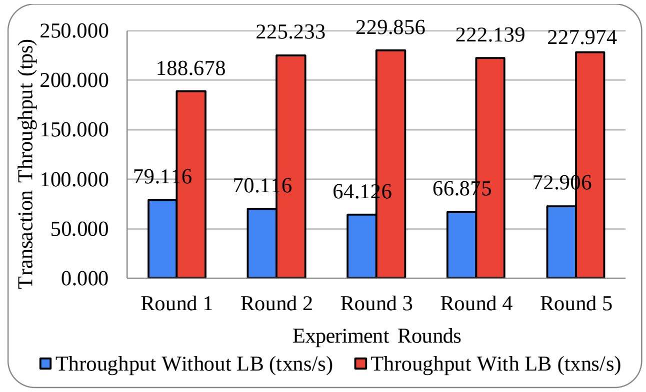

Fig. 12 Transaction Throughput in tps (500 txns)

From Figure 12 it is evident that the transaction throughput is improved consistently through all rounds of the experiment for arrival rate 500.

4.2.5. Performance Evaluation with Arrival Rate 1000 For Each Round

Table 7. Performance evaluation with an arrival rate of 1000

Rounds	Duration Without LB (s)	Duration with LB (s)	Throughput without LB (txns/s)	Throughput with LB (txns/s)
Round 1	13.623	5.117	73.400	195.413
Round 2	12.450	5.024	80.316	199.030
Round 3	12.782	5.921	78.229	168.883
Round 4	13.914	5.100	71.865	196.073
Round 5	13.012	4.950	76.846	201.982

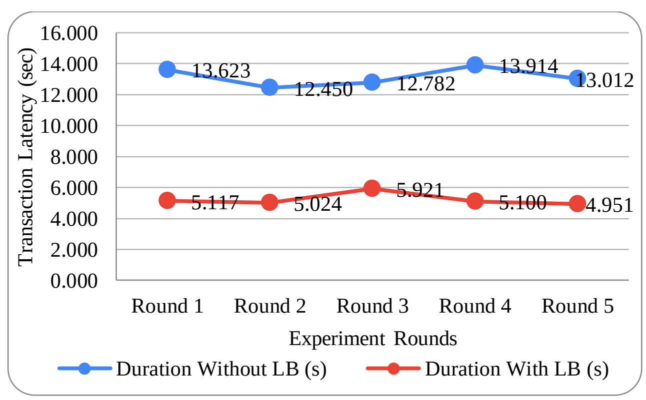

Fig. 13 Transaction latency in sec (1000 txns)

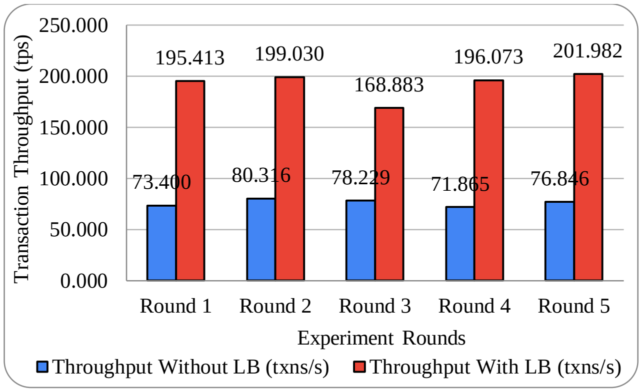

Fig. 14 Transaction Throughput in tps (1000 txns)

From the above, Figures 13 and 14, including the load balancer, show better throughput compared to without a load balancer for an arrival rate of 1000 for every round.

4.2.6. Overall Performance Evaluation

Table 8. Overall transaction latency of all transaction volumes

Transaction Volumes	Avg. Duration without LB (s)	Avg. Duration with LB (s)
10	0.396	0.135
50	0.655	0.287
100	1.31	0.524
500	7.117	2.464
1000	12.717	5.062

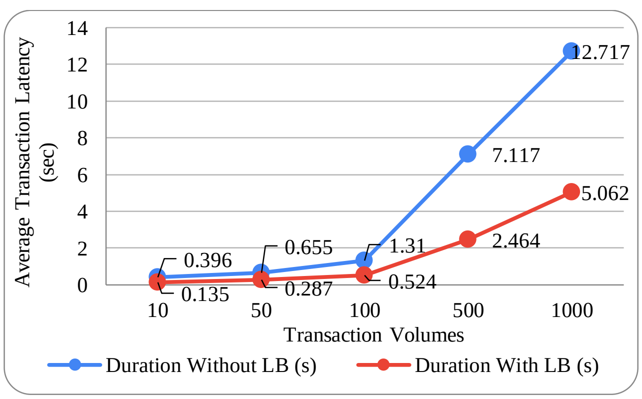

Fig. 15 Average transaction latency of all transaction volumes

Figure 15 illustrates that the average duration without a load balancer exhibits more duration times compared to the duration with a load balancer as transaction volumes increase. This suggests that the load balancer's role in efficient workload management becomes increasingly significant as the system handles a greater number of transactions.

The average duration with a load balancer implies that it maintains a shorter experiment duration by evenly dividing the workload among mining pools, hence reducing processing time as transaction volumes increase.

Table 9. Overall performance evaluation of all transaction volumes

Transaction Volumes	Avg. Duration without LB (s)	Avg. Duration with LB (s)
10	53.716	126.996
50	72.802	174.174
100	76.336	190.956
500	70.288	202.839
1000	74.568	197.638

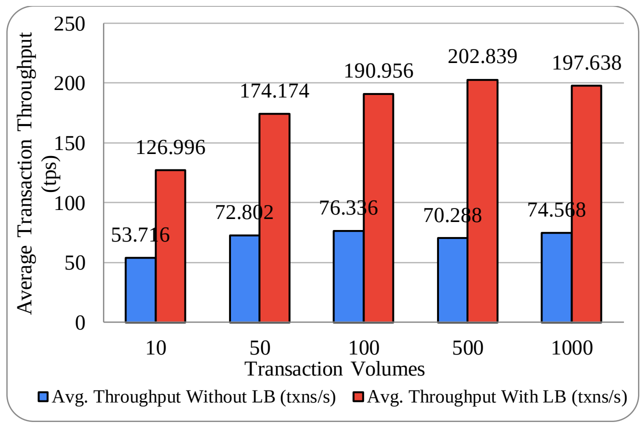

Fig. 16 Overall transaction Throughput (tps)

Figure 16 depicts the ability of a load balancer to efficiently sustain an ongoing and higher level of throughput as the volume of transactions increases. It consistently maintains a high level that implies that the load balancer effectively manages incoming transactions, ensuring constant performance with respect to increasing transaction volumes.

5. Conclusion

The Split-Join blockchain framework aims to improve the performance and scalability of the blockchain networks. The experiments conducted by adding the load balancer to this framework have shown significant improvement in TPS. A load balancer avoids overloading a single mining pool of the framework while the other is experiencing heavy transaction arrivals. This allows for better performance and reduces unnecessary overheads. To address the scalability challenges of blockchain technology, there is a need to redesign its core architecture and adopt new methods that promote scalability. Results signify that the load balancer improves the overall performance of the split-join blockchain and enhances the scalability to make it suitable for large applications. Future research includes identifying the efficient load-balancing techniques and consensus algorithms for the split-join blockchain framework.

References

[1] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *Bitcoin*, 2008. [Google Scholar] [Publisher Link]

[2] Arthur Gervais et al., "On the Security and Performance of Proof of Work Blockchains," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3-16, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[3] N'guessan Patrice Akoguhi, and M. Bhavsingh, "Blockchain Technology in Real Estate: Applications, Challenges, and Future Prospects," *International Journal of Computer Engineering in Research Trends*, vol. 10, no. 9, pp. 16-21, 2023. [CrossRef] [Publisher Link]

[4] Daniel Larimer, "Delegated Proof-of-Stake (DPOS)," *Bitcoin Forum*, 2014. [Google Scholar] [Publisher Link]

[5] Leela Mahesh Reddy, and Srinath Doss, "Turbocharging Blockchain: Cutting-Edge Load Balancing for Split-Join Architecture," *International Journal of Computer Engineering in Research Trends*, vol. 11, no. 5, pp. 9-23, 2024. [CrossRef] [Publisher Link]

[6] Ali Vatankhah Barenji, Yaling Zhang, and M. Bhavsingh, "A Blockchain-Based Framework for Enhancing Privacy and Security in Online Transactions," *International Journal of Computer Engineering in Research Trends*, vol. 10, no. 11, pp. 1-9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Samy T., and Sunil Vijaya Kumar Gaddam, "Leveraging Quantum Computing for Enhanced Cryptographic Protocols in Cloud Security," *International Journal of Computer Engineering in Research Trends*, vol. 11, no. 5, pp. 1-8, 2024. [CrossRef] [Publisher Link]

[8] Mohammed Adam Kunna Azrag et al., "A Novel Blockchain-based Framework for Enhancing Supply Chain Management," *International Journal of Computer Engineering in Research Trends*, vol. 10, no. 6, pp. 22-28, 2023. [CrossRef] [Publisher Link]

[9] Anatoly Yakovenko, "Solana: A New Architecture for a High Performance Blockchain v0.8.13," Whitepaper, 2018. [Google Scholar] [Publisher Link]

[10] John T. Mentzer et al., "Defining Supply Chain Management," *Journal of Business Logistics*, vol. 22, no. 2, pp. 1-25, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[11] Friorik P. Hjalmarsson et al., "Blockchain-Based E-Voting System," *IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, pp. 983-986, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Lokhande Gaurav, Maloth Bhavsingh, and Jaime Lloret, "QuantumShield framework: Pioneering Resilient Security in IoT Networks through Quantum-Resistant Cryptography and Federated Learning Techniques," *International Journal of Computer Engineering in Research Trends*, vol. 11, no. 1, pp. 61-69, 2024. [CrossRef] [Publisher Link]

[13] Andrei-Dragos Popescu, "Decentralized Finance (DeFi)–the Lego of Finance," *Social Sciences and Education Research Review*, vol. 7, no. 1, pp. 321-349, 2020. [Google Scholar] [Publisher Link]

[14] Di Yang et al., "A Review on Scalability of Blockchain," *Proceedings of the 2020 2nd International Conference on Blockchain Technology*, pp. 1-6, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Anamika Chauhan et al., "Blockchain and Scalability," *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, Lisbon, Portugal, pp. 122-128, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[16] Naoya Okanami, Ryuya Nakamura, and Takashi Nishide, "Load Balancing for Sharded Blockchains," *Financial Cryptography and Data Security*, pp. 512-524, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Birger Larsen et al., "NebulaSafeguard Prototypes: Advancing Information Integrity through Unified Decentralized Verification Schemes," *International Journal of Computer Engineering in Research Trends*, vol. 11, no. 1, pp. 43-50, 2024. [CrossRef] [Publisher Link]

[18] Victor Youdom Kemmoe et al., "Recent Advances in Smart Contracts: A Technical Overview and State of the Art," *IEEE Access*, vol. 8, pp. 117782-117801, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Vitalik Buterin, and Gavin Wood, "Ethereum White Paper," *Github Repository*, vol. 1, pp. 22-23, 2013. [Google Scholar] [Publisher Link]

[20] Qiheng Zhou et al., "Solutions to Scalability of Blockchain: A Survey," *IEEE Access*, vol. 8, pp. 16440-16455, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Guangsheng Yu et al., "Survey: Sharding in Blockchains," *IEEE Access*, vol. 8, pp. 14155-14181, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Yoad Lewenberg, Yonatan Sompolinsky, and Aviv Zohar, "Inclusive Block Chain Protocols," *Financial Cryptography and Data Security*, pp. 528-547, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[23] Jiaping Wang, and Hao Wang, "Monoxide: Scale out Blockchains with Asynchronous Consensus Zones," *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation*, pp. 95-112, 2019. [Google Scholar] [Publisher Link]

[24] K. Samunnisa, and Sunil Vijaya Kumar Gaddam, "Blockchain-Based Decentralized Identity Management for Secure Digital Transactions," *Synthesis A Multidisciplinary Research*, vol. 1, no. 2, pp. 22-29, 2023. [Publisher Link]

[25] S. Kiran, and Sreekanth Rallapall, "Innovative Blockchain Split-Join Architecture for Optimized Data Management," *Synthesis A Multidisciplinary Research*, vol. 1, no. 3, pp. 1-11, 2023. [Publisher Link]

[26] Sukhveer Kaur et al., "Round-Robin Based Load Balancing in Software Defined Networking," *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, pp. 2136-2139, 2015. [Google Scholar] [Publisher Link]

[27] Liangshuai Zhu, Jianming Cui, and Gaofeng Xiong, "Improved Dynamic Load Balancing Algorithm Based on Least-Connection Scheduling," *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, pp. 1858-1862, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[28] Leela Mahesh Reddy, and K. Madhavi, "Blockchain Split-Join Architecture: A Novel Framework for Improved Transaction Processing," *Frontiers Collaborative Research*, vol. 1, no. 3, pp. 20-29, 2023. [Publisher Link]