Scalable On-chain Storage Mechanism for DApps with High Transactions Rate and Low Latency, over a Permissionless Blockchain Multichain System. By PhD Student Juan David Peña Melo

Advisors: PhD. Lil María Rodríguez Henríquez, PhD. Julio César Pérez Sansalvador

Introduction

Blockchain technology is an innovative digital ledger system that provides secure record-keeping by storing and redundantly verifying transactions on a distributed network of nodes. This technology bifurcates into two primary classes: public (or permissionless) and private (or permissioned) blockchains. Permissionless blockchains are open access and allow the participation of any individual or entity, while permissioned blockchains require credential validation or sometimes an economic incentive to allow collaboration in the network. Permissionless blockchains have driven the development of decentralized applications, also known as DApps, which exhibit features such as distributed business logic, resilience to failures at central points, and guarantee of data Immutability.

Motivation

Currently, there are platforms to developer DApps without implementing the decentralized infrastructure.

1. Transactional Performance.
2. Permissionless and public system.
3. Redundant Storage for Latency, Availability, and Consistency.
4. **No Scalable Storage.**

Figure 1: Descentralized topology

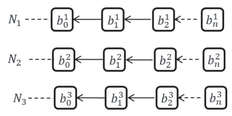


Figure 2: Multichain

Problem Statement

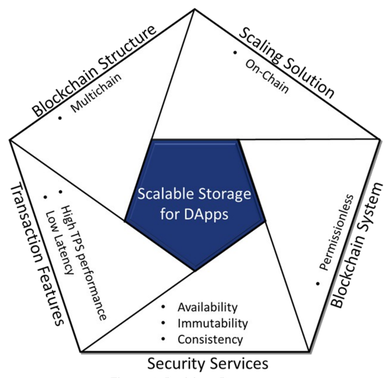


Figure 3. Problem Parameters.

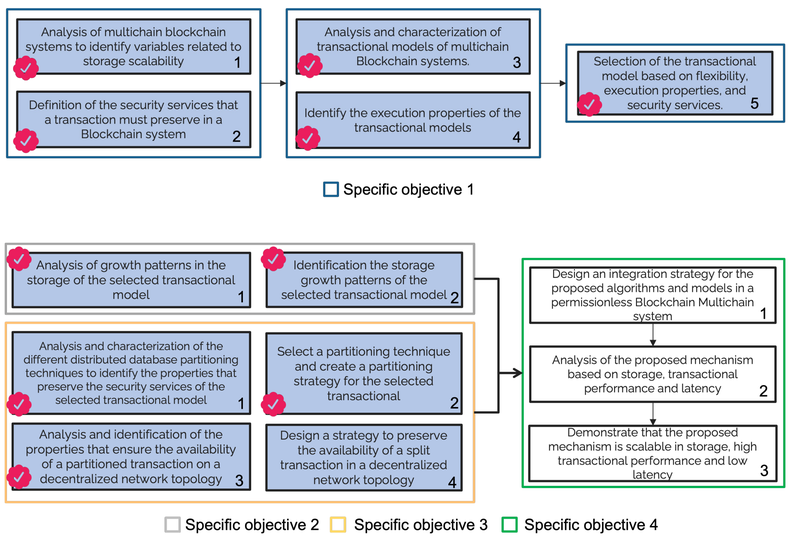
Related Work

Issue	Enhancing Throughput	Proposal	Blockchain System	Throughput	Advantages	Disadvantages
Throughput	Block size	SegWit 2015	Permissionless	20 TPS	Enhances storage efficiency and latency	Increases capacity, not scalable solution.
	Block size	London 2021	Permissionless	85 TPS	Increase throughput reduce fees	Increased gas and block size
	Off-chain	Polygon 2018	Permissionless	65,000 TPS	Ethereum compatibility low fees	Inconsistency and security risks
	Off-chain	Lightning 2015	Permissionless	1 Million TPS	Instant transactions low fees	Funds blocked in payment channels
	Sharding	Elastico 2016	Permissionless	40 TPS	Parallelizes transactional processing	Increase storage 1 GB per day
	Sharding	Omniledger 2018	Permissionless	4000 TPS	Ensures that nodes redistributed	Increase storage 28 GB per day
	Sharding	RapidChain 2018	Permissionless	7380 TPS	Efficiency in network configuration	Increase storage 159.6GB per day
Issue	**Reducing Storage**	**Proposal**	**Blockchain System**	**Saving Storage**	**Advantages**	**Disadvantages**
Storage	Centralized	CUB 2018	Permissioned*	90% Saving	Block Allocation Optimization	Assume all nodes are honest
	Centralized	Jidar 2019	Permissionless	98% Saving	Only stores transaction relevant	Extra storage for transaction
	Centralized	SASLedger 2021	Permissioned*	93% Saving	Guarantee integrity of the database	Requires remote database server
	Decentralized	SE-Chain 2021	Permissionless	70% Saving	The consistent blocks stored fewer replicas	High Latency Low availability
	Decentralized	Lightweight 2021	Permissioned*	46% Saving	Optimization scheme based on Reed-Solomon	Limited full-scale Compatibility
	Decentralized	Scalable Storage for VANET 2023	Permissioned*	33%-50% Saving	Collaborative ledger storage, dynamic copy	Coordination and latencies challenges
	Centralized	RESS 2023	Permissionless	90% Saving	Efficient erasure coding enhanced scalability	Additional resources for coding/decoding

Main Objective

Develop a scalable on-chain storage mechanism for DApps that supports high transaction rates and low latency while ensuring security services such as immutability, availability, and consistency over a permissionless Blockchain Multichain System.

Methodological Procedure



Previous Advances

Bitcoin Transactions Types and Their Impact on Storage Scalability

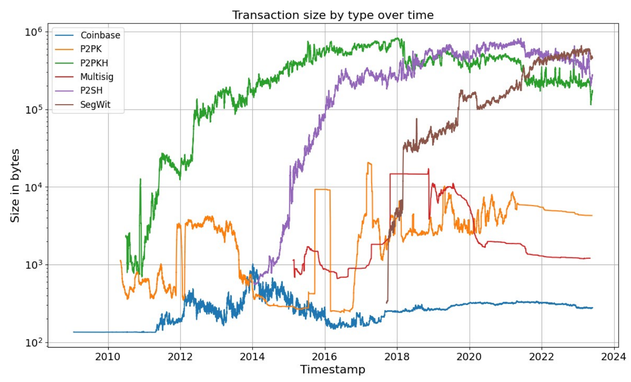


Figure 4. Evolution of storage size by transaction type on the Bitcoin network over time. The x-axis represents the timestamps of the transactions, and the y-axis shows the storage size in that time range.

Defining the UTXO Model as a Digraph

Formally, it is represented as a tuple $G = (V, R)$, where V is a finite set of vértices, and R is a set of edges, such that:

- The set of vertices represent the outputs of the UTXO model and is divided into two subset $V = \Xi \cup \Theta$. Here, Ξ is the set of spent outputs, and Θ is the set of unspent outputs.
- The set of edges R is determined by the Spent-by relation, which specifies how the Θ and Ξ are related.

Spent By Relation "← "

To define the Spent-by relation, we begin by partitioning Graph G into a Subgraph $H = (V'R')$, as illustrated in Figure 4, where $\Xi_s \subset \Xi$, $\Theta_s \subset \Theta$, such as $V' = \Xi_s \cup \Theta_s$

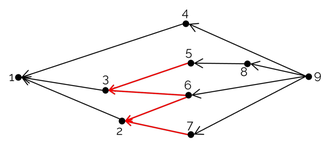


Figure 5. Visualization of a subset of the UTXO model represented as a DAG, where the highlighted subgraph H delineates the relationship between spent and unspent outputs within the system.

Let us define the set of edges R', which satisfies the following properties:

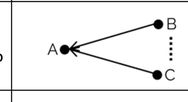
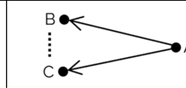
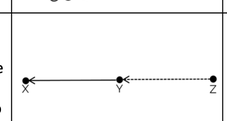
- $R' = \{(x, y)$ This represent all pairs (x, y), where x and y are elements of the set Ξ_s and Θ_s respectively.
- $|R'| = |V'| - 1$. This means that the number of edges in R' is one less than the number of vertices V'.

The Spent-by relation defines the set of relations that exist between subsets of unspent outputs and spent outputs. Formally, we define the Spent-by relation as a subset R' of the Cartesian product $\Xi_s \times \Theta_s$:

$$x \leftarrow y, where\ x \in \Xi_s\ and\ y \in \Theta_s$$

Based on the cardinality relation between Ξ_s and Θ_s different transactional behaviors are observed: Splitting, Merging, and Transferring.

$$|\Theta_s| > |\Xi_s| \vee |\Theta_s| < |\Xi_s| \vee |\Theta_s| = |\Xi_s|$$

Case	Pattern	Output per Pattern	Storage per Pattern
Splitting: this pattern involves dividing one or several unspent outputs into smaller parts		$\sigma_s = \frac{\kappa_s}{t}$	$\omega_s = \tau\sigma_s\eta$
Merging: the consolidation of multiple outputs into a reduced set of unspent outputs		$\sigma_m = \frac{\kappa_m}{t}$	$\omega_m = \tau\sigma_m\eta$
Transferring: represents the exchange of ownership between parties without the need to engage in computational processing to split or merge unspent outputs		$\sigma_t = \frac{1}{t}$	$\omega_t = \tau\sigma_t\eta$

Advances

Relationship between Throughput and Storage

Transactional throughput refers to the system's capacity to process transactions over a time interval, and each transaction in environments such as Bitcoin can generate multiple outputs.

- **Outputs across transactional patterns** this parameter, denoted as k represents the total number of outputs generated by all transactional patterns (splitting, merging, and transferring). It is the sum of the outputs from each pattern, expressed as follows:

$$k = k_m + k_t + k_s$$

- **Number of outputs of all transactional pattern in a time interval** this parameter, denoted as σ,represents the total number of outputs generated by all transactional patterns per time interval. It is calculated by dividing the total number of outputs k by the time interval t expressed as:

$$\sigma = \frac{k}{t}$$

- **Average number of outputs per transaction** this parameter, denoted as λ represents the average number of outputs generated per transaction. It is calculated by dividing the total number of outputs k by the total number of transactions Tx expressed as:

$$\lambda = \frac{k}{Tx}$$

- **Transactional Throughput** we define transactional throughput (tps) as the number of transactions processed per second. If σ is the total number of outputs generated in a time interval t and λ is the average number of outputs per transaction, then transactional throughput is calculated as:

$$tps \approx \frac{\sigma}{\lambda}$$

- **Throughput-Storage Relationship** the storage generated by each transactional pattern is related to the average output size τ the number of outputs generated per time interval σ, and the number of nodes in the system η. Therefore, the relation between the transactional throughput and storage is given by:

$$\omega \approx \tau\sigma\eta$$

By increasing the transactional throughput tps,we also increase the number of outputs per interval of time σ, and therefore, the required storage increases.

Availability and Immutability in Blockchain

Specific Objective 3

3. To design and implement an algorithm on a permissionless Blockchain System that partitions storage growth patterns of a UTXO, preserving the necessary properties to ensure the availability and immutability of transactions.

Classification and Visualization of Bitcoin Blockchain Based on Transactional Patterns

Classification	Transactions	Inputs	Outputs	Avg. Size	Percentage
Tiny	239549749	239020774	382447776	210.85	24.72%
Very Light	184538894	184537204	368641067	225.48	19.04%
Light	157463844	157583090	326928176	262.00	16.25%
Medium	193859227	282466682	357225612	354.87	20.00%
Heavy	145111619	387395120	460528469	592.60	14.97%
Massive	48313147	1244701239	817890608	5454.39	4.98%

Table 1: Categorisation of Bitcoin Transactions by Storage Size

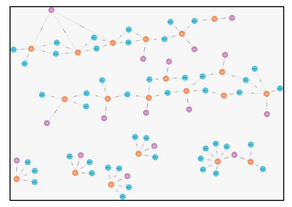


Figure 6. Visualization of the Bitcoin Blockchain using Neo4j

Analysis of Bitcoin Blockchain Focused on Services of Availability and Immutability

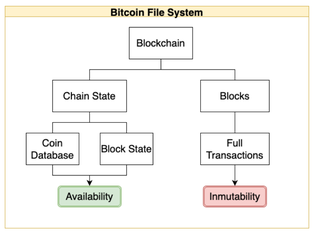


Figure 7. Topology of the Bitcoin File System and Security Services

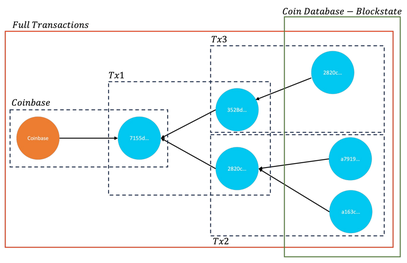
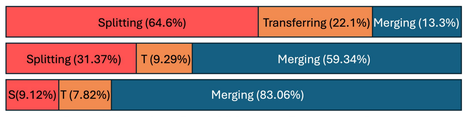


Figure 8. Abstraction of Bitcoin File System for Security Services



- Frequency
- Storage
- Storage/Frequency

Figure 9. Chain State Trade-Off Based on Transactional Patters

Practical Implementation of Blockchain Fragmentation and Chain State Trade-Off in the Bitcoin Blockchain

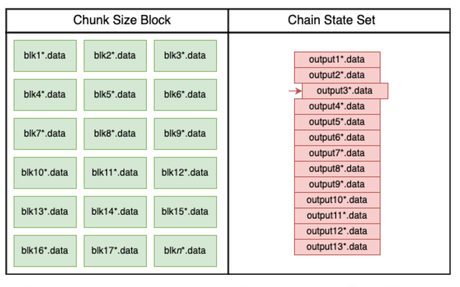
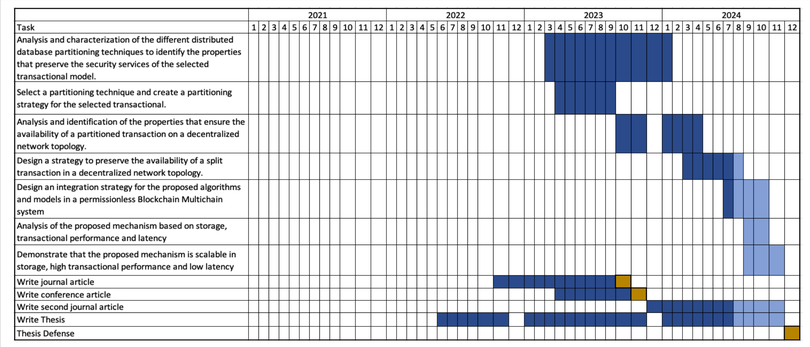


Figure 10. Implementation Mechanism of Fragmentation and Trade-Off.

Schedule



References

D. Melo, S. P. Hernandez, L. Rodríguez and J. C. Pérez-Sansalvador, "Bitcoin Transactions Types and Their Impact on Storage Scalability," 2023 IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Paris, France, 2023, pp. 1-6, doi: 10.1109/WETICE57085.2023.10477780.

Melo D, Pomares-Hernández SE, Rodríguez-Henríquez LMX, Pérez-Sansalvador JC. Unlocking Blockchain UTXO Transactional Patterns and Their Effect on Storage and Throughput Trade-Offs. Computers. 2024; 13(6):146. https://doi.org/10.3390/computers13060146

A. M. Antonopoulos, Mastering Ethereum. O'Reilly Media, Inc., 1st ed., 2018.

G. Zheng, Ethereum Smart Contract Development in Solidity. (Springer Singa-pore)., 1st ed., 2021.

J. von zur Gathen, Cryptography Theory and Practice. Springer Heidelberg NewYork Dordrecht London, library of congress control number: 2015957084 ed.,2015.

S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," tech. rep., 2008.

W. Cai, Z. Wang, J. Ernst, Z. Hong, and C. Feng, "Decentralized applications: The blockchain-empowered software system," IEEE Access, vol. 6, pp. 53019– 53033, 10 2018.