Comparison of Blockchain Storage Methods: BlockLSM, ChainKV, and COLE

July 2024

1 Introduction

This document provides a detailed comparison of three blockchain storage methods: BlockLSM, ChainKV, and COLE. Each method addresses the inefficiencies of the Log-Structured Merge-tree (LSM-tree) based key-value (KV) storage engine in handling Ethereum's growing data size.

2 BlockLSM

2.1 Overview

BlockLSM is designed to optimize Ethereum's storage by maintaining the block structure during data storage. This method introduces the Ether-aware LSM-tree based KV store, known as Block-LSM, which aims to reduce unnecessary I/O operations caused by the traditional LSM-tree structure.

2.2 Key Features

- **Prefix-based Hashing**: Uses block numbers as prefixes for KV entries, preserving the order of Ethereum blocks.
- **Block Group-based Prefix**: Groups multiple blocks together to reduce space overhead.
- **Attribute-oriented Memory Buffers**: Separates special metadata from other data to avoid key range overlaps and reduce compaction needs.

2.3 Performance

BlockLSM consistently outperforms the original Ethereum storage engine in throughput, compaction operations, and write amplification. The method achieves significant reductions in compaction operations and write amplification coefficient (WA Coef).

3 ChainKV

3.1 Overview

ChainKV introduces a semantics-aware storage paradigm to improve Ethereum's storage management performance. It addresses the incompatibility between Ethereum semantics and the storage engine's characteristics by separating data types into different storage zones.

3.2 Key Features

- **Semantic-Aware Storage Zones**: Separates non-state and state data into different storage zones.
- **Prefix MPT**: An ADS data transformer that preserves data locality when persisting MPT nodes.
- **Space Gaming Caching Policy**: Dynamically adjusts cache sizes based on real-time workloads.
- **Node Crash Recovery**: Lightweight recovery mechanism that replaces the traditional write-ahead log (WAL).

3.3 Performance

ChainKV achieves significant performance improvements over the original Ethereum storage engine, with up to 1.99 times better synchronization and 4.20 times better query operations. The method also shows substantial improvements in write performance and read performance, especially for state data queries.

4 COLE

4.1 Overview

COLE proposes a column-based learned storage system for blockchain data. It uses learned models to index and retrieve data efficiently, significantly reducing storage size and improving system performance.

4.2 Key Features

- **Column-Based Design**: Stores each piece of data's history in a column format.
- **Learned Index Models**: Uses learned models to predict data locations and index states efficiently.
- **Asynchronous Merge Strategy**: Manages data writes without causing delays.

- **Merkle Hash Trees (MHT)**: Ensures data integrity at each storage level.

4.3 Performance

COLE reduces storage size by up to 94 percent and increases system performance by 1.4 to 5.4 times compared to MPT. The method optimizes write operations and supports provenance queries efficiently.

5 Comparison

Feature	**BlockLSM**	**ChainKV**	**COLE**
Data Storage Approach	Prefix-based Hashing	Semantic-Aware Zones	Column-Based Design
Compaction Reduction	Yes	Yes	N/A
Data Locality	Maintained via Prefixes	Preserved via Prefix MPT	Improved via Learned Index
Cache Management	Memory Buffers	Space Gaming Cache	Asynchronous Merges
Recovery Mechanism	Standard WAL	Lightweight Node Recovery	Asynchronous Merge Checkpoints
Performance Improvement	Up to 182.75%	Up to 4.20x	Up to 5.4x
Storage Reduction	Moderate	Moderate	Up to 94%

Table 1: Comparison of BlockLSM, ChainKV, and COLE

6 Conclusion

BlockLSM, ChainKV, and COLE each offer unique approaches to optimizing Ethereum's storage system. BlockLSM focuses on maintaining block order to reduce I/O operations, ChainKV introduces semantic-aware zones and dynamic caching, while COLE leverages a column-based design and learned indexes to minimize storage size and enhance performance.