A Hybrid Approach Leveraging Relational Database Management Systems for Fast Blockchain Data Retrieval

Fauzan Khifzan[1], Muhammad Thoriq Ramadhan[2], Dhiwa Kusumah[3], Jong Uk Choi[4]

MarkAny Inc.

{fauzan[1], thoriqramadhanm[2], dhiwa[3]}@ganeshait.com, juchoi[4]@markany.com

Abstract

Blockchain systems provide strong immutability and verifiable state transitions but suffer from slow and inefficient data retrieval due to consensus overhead and Merkle-tree based storage. Modern applications, however, require fast, complex queries that these systems cannot natively support. This paper presents a Blockchain-First Hybrid Architecture that preserves blockchain integrity while supports high-performance querying through an off-chain relational database. In this design, the blockchain remains the sole Source of Truth for all writes, while PostgreSQL serves as a materialized view cache updated exclusively by an event-driven synchronization layer. We evaluate the architecture through controlled experiments measuring concurrency performance, data complexity scaling, and throughput. Results show that the middleware delivers up to 2× lower latency, 10×– 20× faster complex queries, and 33× higher throughput compared to direct blockchain queries, with significantly more stable performance under growing schema sizes. These findings demonstrate that hybrid designs effectively bridge the gap between blockchain security guarantees and real- world application performance, offering a practical approach for scalable decentralized systems.

Ⅰ. Introduction

Blockchain technology enables transparency and immutability [1]. Through their data structures and consensus mechanisms, blockchains enable more secure transactions and promote decentralization. However, as transaction volumes grow with increased adoption, efficient querying mechanisms become critical for maintaining scalability and performance. This creates a gap between native blockchain capabilities and the requirements of modern applications [2]. In contrast, traditional databases support highly efficient querying through mature query languages and advanced indexing techniques [3]. In this study, we propose an approach for interacting with data stored in blockchain smart contracts by leveraging materialized views in relational databases. We combine these two architectures using a hybrid design classified as an External Database Integration Scheme [1], in which an intermediate layer synchronizes blockchain data into an external database to support complex and efficient queries.

Ⅱ. Method

Recent literature categorizes blockchain query optimizations into four distinct schemes: external database integration, on-chain indexing, smart contract querying, and data structure modifications [1]. Some researchers propose modifying the underlying blockchain structure. For example, SE-Chain introduces the Adaptive Balanced Merkle (AB-M) tree [4]. While SE-Chain improves query time significantly, it requires fundamental changes to the blockchain protocol. In contrast, our middleware

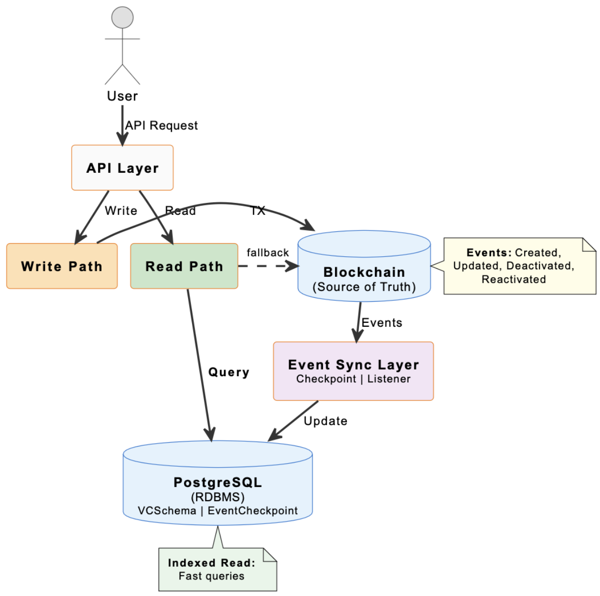

Fig. 1. System architecture of the hybrid blockchain-RDBMS approach. Write operations are directed to the blockchain to ensure immutability, while read operations are served from PostgreSQL for high performance. The Event Sync Layer maintains data consistency through checkpoint-based synchronization.

approach leverages standard RDBMS technologies without altering the consensus layer. Sestili et al. (2023) emphasize the importance of analyzing read latency and throughput [5]. Wickboldt (2019) demonstrated that system performance is decisively dependent on configuration parameters, such as transaction arrival rates and the number of concurrent users [6].

We adopted the External Database Integration Scheme (Middleware Approach) that decouples data acquisition from query processing [1]. The architecture implements a clear separation between write and read operations. All write requests flow through the API Layer directly to the blockchain, ensuring that every state mutation is cryptographically verified and permanently recorded. The blockchain emits events for each operation (Created, Updated, Deactivated, Reactivated), which are captured by the Event Sync Layer that maintains a checkpoint mechanism to track processed blocks.

The Event Sync Layer continuously monitors the blockchain through an Ethereum RPC listener, decoding smart contract events and transforming them into structured records persisted into PostgreSQL. For read operations, the system routes all queries to the PostgreSQL middleware, leveraging mature RDBMS capabilities such as indexing, joins, and query optimization to deliver sub-second response times.

The blockchain functions as a permanent and decentralized trust anchor for sensitive digital assets [7]. The ledger provides a persistent and tamper-evident log [7]. Blockchain networks address the classical Byzantine agreement problem by employing distributed consensus protocols [4]. PostgreSQL is adopted as the off-chain view-cache due to its robust query capabilities [7]. PostgreSQL provides full SQL support, enabling complex joins, range queries, and rich semantic data retrieval [1]. PostgreSQL offers advanced concurrency control—Serializable Snapshot Isolation (SSI)—which aligns well with contemporary blockchain-database hybrids [7].

We evaluate performance through three test groups. Group A (Concurrency Performance) measures system latency under varying concurrent loads (5, 20, 50 Virtual Users) while holding dataset size constant at 10 records. Group B (Data Complexity Performance) examines query performance as dataset size scales from 10 to 50 to 100 records, testing pagination operations and ID-based filtering. Group C (Throughput Performance) measures maximum transaction processing capacity (TPS) under continuous load across dataset sizes. TPS directly reflects system capacity and scalability. Tail latency is assessed using the 99th percentile (p99).

Ⅲ. Conclusion

The system prototype was implemented across three distributed nodes: a MacBook Air M4 client running K6, a MacBook Pro M1 middleware node hosting PostgreSQL 18.1, and a remote VPS running Hyperledger Besu. Under light load (5 VUs), the middleware achieved 2.12× faster response time (63.98 ms vs 135.87 ms). Under moderate load (20 VUs), the middleware maintained a 1.68× performance advantage (125.09 ms vs 209.62 ms). Under heavy load (50 VUs), the middleware provided 1.43× better latency (403.39 ms vs 575.05 ms).

Figure 3 compares throughput (TPS) between a traditional database and a blockchain system as schema complexity increases (10, 50, and 100

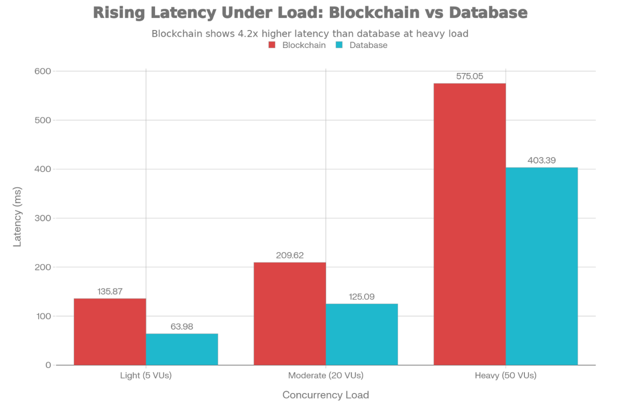

Fig. 2. Read Latency test diagram for Blockchain and RDMS query on 3 different different concurrency load.

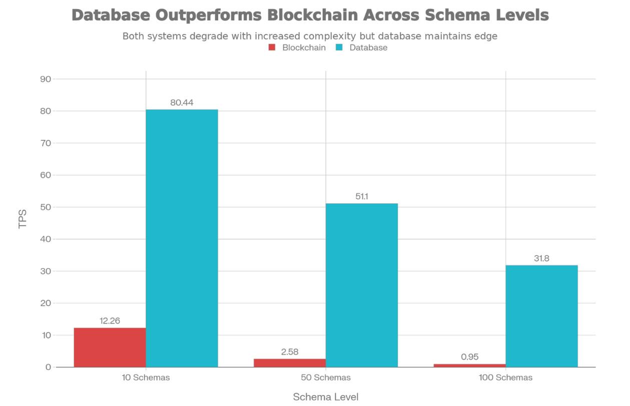

Fig. 3. Throughput test diagram for Blockchain and RDMS query on 3 different different schema sizes

schemas), showing that the database consistently and significantly outperforms the blockchain, achieving 80.44 TPS vs. 12.26 TPS at 10 schemas, 51.1 TPS vs. 2.58 TPS at 50 schemas, and 31.8 TPS vs. 0.95 TPS at 100 schemas. Although both systems experience performance degradation as schema complexity grows, the decline is much steeper for blockchain due to consensus and validation overhead, demonstrating that databases are better suited for high-throughput, schema-intensive workloads, while blockchain is more appropriate for integrity-critical metadata in hybrid architectures.

This study demonstrates that the Blockchain-First Hybrid Architecture effectively resolves the tension between blockchain integrity and application-level performance. By separating write integrity (on-chain) from read efficiency (off-chain), the hybrid model preserves immutability, verifiability, and decentralization while delivering query speeds appropriate for real-world applications.

ACKNOWLEDGMENT

This research was supported by MarkAny Inc. and also reviewed and edited by other GaneshaDCERT project team members. We thank the reviewers for their insightful comments which improved the quality of this paper.

REFERENCES

[1] Emmanuel Acheampong Asiamah, Eliel Keelson, Andrew Selasi Agbemenu, Eric Tutu Tchao, TheresaSamuelle Adjaidoo, and Griffith Selorm Klogo. Optimizing blockchain querying: A comprehensive review of techniques, challenges, and future directions. IEEE Access, 2024.

[2] Qingsong Guo, Chao Zhang, Shuxun Zhang, and Jiaheng Lu. Multi-model query languages: taming the variety of big data. Distributed and Parallel Databases, 42(1):31-71, 2024.

[3] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. Blockbench: A framework for analyzing private blockchains. In Proceedings of the 2017 ACM international conference on management of data, pages 1085- 1100, 2017.

[4] Da-Yu Jia, Jun-Chang Xin, Zhi-Qiong Wang, Han Lei, and Guo-Ren Wang. Se-chain: a scalable storage and efficient retrieval model for blockchain. Journal of Computer Science and Technology, 36(3):693- 706, 2021.

[5] Leonardo Mostarda, Andrea Pinna, Davide Sestili, and Roberto Tonelli. Performance analysis of a besu permissioned blockchain. In International Conference on Advanced Information Networking and Applications, pages 279- 291. Springer, 2023.

[6] Clemens Wickboldt. Benchmarking a blockchain-based certification storage system. Technical report, Discussion Paper, 2019.

[7] Zerui Ge, Dumitrel Loghin, Beng Chin Ooi, Pingcheng Ruan, and Tianwen Wang. Hybrid blockchain database systems: design and performance. Proceedings of the VLDB Endowment, 15(5):1092- 1104, 2022.