CS5280
Concurrent Execution of Blockchain Smart Contract transactions (SCTs)

Darpan Gaur CO21BTECH11004

Yoshita Kondapalli CO21BTECH11008

Option 1

Intorduction and Motivation

Blockchain technology has emerged as a transformative paradigm, offering decentralized, secure, and immutable transaction processing. A fundamental component of blockchain systems is smart contracts, self-executing programs that facilitate trustless interactions between parties. However, these self-executing agreements, commonly implemented on blockchains, are typically validated sequentially within a block, leading to performance bottlenecks. This bottleneck limits the performance of blockchain networks, especially as adoption grows and transaction volumes increase.

To address these limitations, parallel execution of smart contract transactions (SCTs) has been proposed. By executing multiple SCTs concurrently, blockchain networks can achieve higher throughput and lower latency, improving the system's overall performance. However, concurrent execution of SCTs introduces challenges related to consistency, isolation, and atomicity, which must be addressed to ensure the integrity and security of the blockchain network.

Approach: Multi-Bin Parallel Scheduler (MBPS)

MBPS framework facilitates parallel transaction execution while preserving a deterministic order, thereby leveraging the capabilities of multicore systems to enhance the efficiency of blockchain ecosystems. MBPS framework undergoes three crucial stages:

- Conflict Detection: Identify conflicts between transactions to assign them to the correct bins.
- Bin Assignments: Assign bins to the transactions such that any two transactions within the same bin do not conflict or update the same data items.

- Transaction Execution: Transactions within the same bin are executed concurrently.

Approach: DAG-based Parallel Scheduler and Validator

This framework integrates parallel scheduler and smart validator modules into the blockchain node architecture. Parallel scheduler detects dependencies (conflicts) among transactions and schedules them for parallel execution. The smart validator module ensures that the transactions are executed correctly and consistently.

- Parallel Scheduler: Detect transaction dependencies and schedule them for conflict-free parallel execution. It involves two stages:
  - DAG Creation
  - Transaction Execution
- Smart Validator: Validates the transactions to ensure correctness and consistency. It involves two stages:
  - DAG Sharing
  - DAG Validation

References

- Piduguralla, Manaswini, Saheli Chakraborty, Parwat Singh Anjana, and Sathya Peri. ”An efficient framework for execution of smart contracts in hyperledger sawtooth.” arXiv preprint arXiv:2302.08452 (2023).
- Ravish, Ankit, Akshay Tejwani, Manaswini Piduguralla and Sathya Peri. “Unleashing Multicore Strength for Efficient Execution of Transactions.” ArXiv abs/2410.22460 (2024): n. pag.