VQL: Providing Query Efficiency and Data Authenticity in Blockchain Systems

Zhe Peng, Haotian Wu, Bin Xiao, Songtao Guo
Query Design Motivation

- Blockchain techniques (cryptocurrency, business transactions, supply chain, insurance, medical care, etc.)

Illustration of blockchain structure

Immutability and verifiability in trustless and distributed environment!

Low query efficiency!
Previous Work

Existing query supported blockchain systems:

• Toshi [1]: provide basic query of block information in Bitcoin
• Ethereum [2]: maintain the current balance of each account in each node
• Etherchain [3]: extend Ethereum basic API to query block time and count transactions
• ECBC [4]: build a tree structure to efficiently query historical transactions of an account

Limited query services

Various data analytical tasks focus on the blockchain:

- [5] analyses Bitcoin transactions and proves that Bitcoin is not a fully anonymous system
- [6] proposes a multi-variant relation model with time series dataset to detect money laundering
- [7] builds a reputation network for blockchain users to reduce transaction risks


Motivation

A query supported blockchain system:

• How to efficiently support various data analytical tasks on top of blockchain systems?
• How to provide trusted query results?
Problem

➢ How to provide efficient query services with verifiability guarantees for blockchain system:
  • Verifiability of querying results by public
  • Querying efficiency
  • Data storage efficiency
Architecture

- Service model
  - Blockchain, Middleware layer, Application layer
System Overview

Blockchain
- Transactions
- Verification

Micro database
- Transactions
- Fingerprint

Key database
- Transactions
- Fingerprint

Applications
- Query
- Data analysis

Construct

Verify

Query
System Design

Middleware architecture

- Key database, Micro database with hash values
  - Store hash values in blockchain
  - Integrity and authenticity functions

- Hash value of database can be verified by miners
- Databases are dynamically updated and merged
Middleware Update Algo.

- Middleware update every month
  - Each day
    - Construct a new Micro database
    - Calculate its hash
  - End of each month
    - Merge all Micro databases into Key database
    - Calculate Key database’s hash
    - Delete all Micro databases
System Design

- Efficient query services
  - Data Query
    - Block
    - Transaction

- Data storage efficiency
  - Periodically store snapshot and hash value of database
  - Merge databases to save space
System Design

- Database verification
  - Data in the middleware are consistent with the blockchain
Database Verification Algo.

Miner Database verification

• Download and re-construct databases
  • Data files will be published by the middleware layer

• Calculate fingerprints and compare
  • hash value published by the middleware layer
  • hash value calculated based on the re-constructed database
  • hash value calculated based on the blockchain data

• Write verified fingerprints into blocks
Experimental Implementation

- Blockchain
  - Ethereum

- Middleware layer
  - MongoDB
Performance Evaluation

• Throughput

• Block query time by number of blocks

• Transaction query time by number of transactions

Fig. 4: Throughput comparison between Ethereum and VQL.

Fig. 5: Comparison of block query time with Ethereum and VQL.

Fig. 6: Comparison of transaction query time with Ethereum and VQL.
Conclusion

- **Query problems** in blockchain system
  - Querying efficiency
  - Verifiability of querying results by public

- Our solution: A Verifiable Query Layer
  - The *middleware layer*
  - Dynamically construct, update, and merge databases
  - Verify the consistency of constructed databases

- Experimental analysis
Thank You!!!