

Dataset:

Nowadays, due to the lack of dataset of consortium blockchain, we select a real-world Bitcoin dataset from Chen et al.'s work [1] to make the evaluation. The dataset has 1048576 rows. Each row is a transaction, including the trader, receiver, price and time information. According to our query operators, the dataset lacks the property, block and timestamp information. So, we abstract the price as the property. We categorize the price into five levels and label each level as a property. Besides, we collect the transactions of the same day as a block, and select the generation timestamp of the last transaction as the block's timestamp.

Why selecting FISCO BCOS:

The storage of blockchain is databases, so our storage design is based on the existing interfaces between blockchain and databases. FISCO BCOS is a open source blockchain platform. It supports the relational storage MySQL and the interfaces to customize the schema and index. It also supports the customized query operators by smart contracts. Therefore, we select FISCO BCOS as the experiment platform.

Network set up:

For the blockchain network, we generate a single-node blockchain node network. We use "**bash build_chain.sh -l -p**" command to set up the network.

Then, we use the console to create TreIndex, TreSchema, and other solutions in the evaluation, and store the dataset as the schemas' form into MySQL.

After that, we write the smart contract to make five types of queries. FISCO BCOS supports the CRUD operations by smart contracts. We define the query operators as interfaces and implement their "call" functions in the smart contract.

Finally, we overload "call" functions of the smart contract, deploy it on the single-node, and monitor the CPU cost and the query time cost.

Reference:

[1] Weili Chen, Jun Wu, Zibin Zheng, Chuan Chen, and Yuren Zhou. Market manipulation of bitcoin: Evidence from mining the mt. gox transaction network. In IEEE Conference on Computer Communications, pages 964–972, Paris, France, April. 2019. IEEE.