Midterm Project Report

Shubham Singhal(ss77) & Vinith Krishnan(vinithk2)

1. CodeBase Details

Repo URL: https://gitlab.engr.illinois.edu/vinithk2/ece598pv-sp2020
Commit ID: 920663edb18d8349dee6ddb4473bd2e95eeccafa

2. Design

Main modules in our systems are transaction generator, miner, worker. The data structures which these modules interact with are block/blockchain, mempool, ledger_state and some auxiliary structures on which the main structures are built are H256, H160, Merkle trees etc

Modules
Transaction Generator is responsible for generating valid transactions on the basis of the state of the block at the tip of the longest chain.It runs periodically, picks utxos with recipient addresses corresponding to it's own node and generates new transactions with different recipient addresses.

Miner runs in a separate thread and periodically takes out the transactions from mempool, forms the block which is consistent with ledger state, does PoW, signs it and broadcasts to its neighbours.

Worker module handles various types of messages it receives from its neighbors and takes appropriate action based on message type. It receives both transaction and block messages. Let's discuss more with regards to transaction messages. It receives 3 types of tx messages. First is NewTransactionHashes in which it just checks tx hashes has been received previously in mempool and requests hashes which are new. In GetTransactions, it finds the tx corresponding to hashes in mempool and sends it to the neighbour. In the Transactions message which actually carries the transactions, it verifies whether the tx is properly signed and if it is, then adds it to mempool.

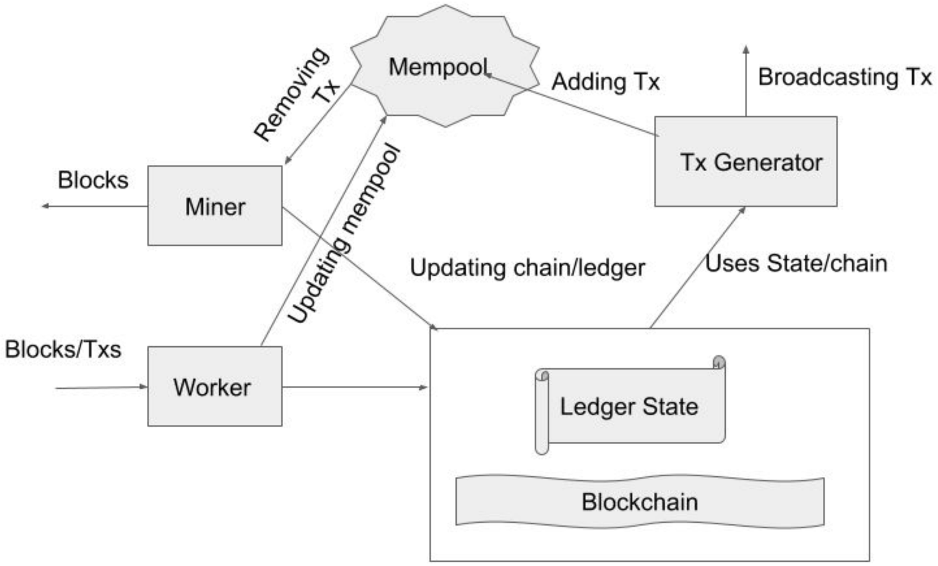
Similar to tx, it processes 3 types of block messages. They are similar to tx ones. NewBlockHashes and GetBlocks do the same thing with block hashes and it uses blockchain for the same instead of mempool. For Blocks' message, it does a couple of things. One, it validates the block is valid. Block validity is checked by making sure all tx are valid and the tx owner actually has the coins, one is spending in the tx. Also we check whether this tx is not double spend. All of this done through the help of ledger state. We maintained state per block instead of a global state, thus we don't have to deal with reversing the tx as the longest chain changes. After verifying the block, it removes the block tx from mempool and adds the block to the blockchain and update the new ledger state. All these checks are only done for blocks whose parent is present in the chain. If an orphan block arrives, then it is kept in the buffer until it's parent arrives. After the processing of its parent, similar processing is done for the child.

Data Structures

Blockchain DS is nothing but a hashmap which maps block hashes to actual block content. Even the buffer is the same but it is a separate hashmap. We also maintain the tiphash in the chain, so that whenever a new block comes, it is inserted there. Mempool consists of a queue of tx hashes for orderly processing of tx, two hashmaps - one which maps tx hash to a boolean and another to actual tx. The reason for having two hashmaps - one to tell whether tx is processed either via miner or incoming block. By that, we need not remove the tx hash from the queue - which may be present somewhere in between. Another hashmap serves as repo for tx. Though we are not cleaning up in current design, in future we will clean this hashmap when a tx is k-deep in the chain and thus confirmed. Ledger State is nothing but a hashmap of UTXO present in the system. As described previously, we maintain it per block, thus making it easier to validate tx and not having the problem of unapply them when the longest chain changes.

This might be clear for the above discussion, but we are using a UTXO model for tx. In this model, a tx is composed of multiple inputs and multiple outputs. An input is nothing but a previous tx hash and an index of the output in it from which the input is taken. An output is the address of the recipient and the value transferred.

Block Diagram



3. Contribution

Vinith worked on Blockchain, Tx Generator, Miner. Shubham worked on Worker, Ledger State, Mempool. Pair programmed for debugging any issues.