PROJECT 4.2 - BITCOIN PART 2

COP5615 - DOS

Group Information:

1. Sourabh Gopal Parvatikar, UFID : 79325142
2. Gurupad Hegde, UFID: 78356173

Youtube video link: https://youtu.be/8SXy-wPkLTA

What is Working:

We have implemented the following features of Bitcoin

1. Creating a wallet : A new wallet is created whenever a client wants to join the bitcoin network
2. Creating a transaction : A transaction is created when a client wants to send coins to another client. Transaction is sent to the miner.
3. Adding pending transactions : Miner checks if the transactions he received are valid and adds them to pending transactions.
4. Mining blocks : Miner mines a block by showing the proof-of-work and adds it to the blockchain.
5. Transacting coins : Miner adds the pending transactions to the mined block and then adds it to the blockchain.
6. Mining rewards : The reward coins that the miner gets for mining a block
7. Mining Difficulty : Kept it low (000 at the beginning of hash) so that blocks could be mined fast.
8. Signing and verifying transactions : The sender signs the transaction with his private_key and miner verifies it using sender’s public_key.
9. Incentive : The client can send incentive (coins) to the miner so that miner adds the clients transaction in the block.
10. Balance check: Miner checks the balance of the sender for each transaction that needs to be added in the block by going through the blockchain.
11. Validating transactions : Miner checks if transactions are valid by verifying the sender’s signature.

To simulate the distributed protocol of Bitcoin, we are using these features in the following way:

1. Initially we create 80 clients (transactors) and 20 miners.
2. The following runs repeatedly with a delay of 1 second on pressing the "Simulate" button in browser:
    a. n (random number between 1 to 5) transactions are created between n randomly chosen senders and receivers to transact m (random number between 1 to 10) coins.
    b. These transactions are broadcasted to all miners.
    c. Miners validate pending transactions and discard those that are not valid. They start mining a block with valid transactions. The first transaction in the block is the reward transaction which the miner receives (default set to 20). Mined block is added to the blockchain.
    d. Miners then broadcast the blockchain having the mined block to all its peers (99 other nodes).
    e. Peers, on receiving the blockchain, check if blockchain is valid and update their blockchain(validation includes validating each block in blockchain using current and previous hash, keeping the longest blockchain received, etc).

This setting helps us continuously perform transactions, create blocks, update blockchain and all other features mentioned above. It also gives us enough data to generate graphs in the front end.

Phoenix framework:

To simulate and visualize the bitcoin working, we have used phoenix framework to show and continuously update the following data:

1. Transactions graph - Shows the number number of transactions at any given time.
2. Mining graph - Shows the number of blocks mined at any given time.
3. Coins transacted graph - Shows the number of coins transacted at any given time.

4. Coins mined graph - Shows the number of coins mined (reward coins received by the miners) at any given time.
5. Other stats:
   Total number of clients
   Total number of miners
   Total Number of transactions
   Total Number of blocks
   Mining Difficulty

6. We have used channels feature of phoenix framework for sending the data from server to browser for updating graphs and stats continuously.

How to Run:

Note: Make sure that Phoenix, node and postgres are installed and postgres role is created.

1. Unzip bitcoinphoenix.zip
2. Run following commands:
   cd bitcoinphoenix
   mix ecto.create
   mix phx.server
3. Go to http://localhost:4000
4. Click on the Simulate button.
5. The Simulation will start the simulation process described above and the charts and other stats will be updating continuously as shown in video.