PROJECT 4.2
REPORT
DISTRIBUTED OPERATING SYSTEMS PRINCIPLES
COP 5615

Group Info:

	Name	UFID
1.	Manishkumar Chopra	17967121
2.	Nimish Kochhar	61394423

What is working:

From the bitcoin protocol, we have implemented the following:

1. Creating a blockchain with a Genesis block - A blockchain with just the starting block
2. Calculating the block's hash - Block's hash need to be calculated to verify the integrity of the block
3. Adding pending transactions - Blockchains have pending transactions that are added every time a new block is mined
4. Mining the block - Miners need to show the proof-of-work to mine a block
5. Allowing to pay a fee with the transaction - This fee is received by the miner in addition to Mining rewards, making it more lucrative to include a particular transaction for mining
6. Mining difficulty - Kept the difficulty low so that the coins can be mined very fast
7. Creating transactions - Allowing to transact from one client to another and broadcasting it
8. Transacting bitcoins - Transfer of bitcoins from one wallet to another
9. Signing transactions - Making sure that the client signs for its own wallet and nobody tampers with a transaction
10. Calculating the balance in a client's wallet - Making sure the amount sent, amount received, mining rewards and fee received are taken into account
11. Checking the validity of the transaction - Check if it's not tampered with and is properly signed
12. Checking the validity of the blocks - Check if the block's untampered and all the transactions in it are valid
13. Checking the validity of the blockchain - Make sure all the blocks are valid in the blockchain and all the blocks are in the correct order. Also make sure that the hash of the current block is the previousHash of the next block
14. Calculating the Merkle tree root - It which will be sent in the block header to other peers to ensure that the transactions list has not been tampered with

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

1. Initially, we take the input and create the inputted number of nodes. Every node works both as a client and a miner.
2. The following runs after a delay every time we press the simulation is started:
    - Every node creates a transaction every 10 seconds of random amount less than it's balance between one of it's neighbour chosen at random
    - These transactions are broadcasted to all the other nodes
    - Every node validates pending transactions and discard those which are invalid. Then, 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 100). Mined block is added to the blockchain.
    - Miners then broadcast the blockchain having the mined block to all of its peers.
    - Peers, on receiving the blockchain, check if the 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 working of the bitcoin, we have used the Phoenix framework to show and continuously update the following data:

- Transactions graph - Shows the number number of transactions at any given time.
- Mining graph - Shows the number of blocks mined at any given time.
- Coins transacted graph - Shows the number of coins transacted at any given time.
- We have used channels feature of phoenix framework for sending the data from server to client, a browser, for updating graphs continuously.

How to run the program:

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

1. Unzip bitcoin_complete.zip and cd into that directory
2. Run the following commands:

    mix ecto.create

    mix phx.server
3. Go to http://localhost:4000 on your browser
4. Enter the number of nodes in the network and click on the start button.

5. This will create the said number of nodes and will start the simulation process described above and the charts will continuously as shown in video.