COP5615 – Project 4.2

Neel Rami, UFID: 7712-3151

Ma Haodi, UFID:7719-2198

Email: nrami@ufl.edu , ma.haodi@ufl.edu

Dec 13, 2018

1. Introduction:

The goal of this (and the previous) project is to implement Bitcoin and (part 2) to build a large simulator to determine its behavior.

In this part, we accomplish the following goals:

1) Finish the distributed protocol.
2) Implement a simulation with at least 100 participants in which coins get mined and transacted.
3) Implement a web interface using the Phoenix that allows access to the ongoing simulation using a web browser (use the matching JavaScript library that allows Phoenix messages to be received in the browser). For charting, we Charts.js library. As part of the simulation, we manage to capture various metrics and send them via Phoenix to the browser.

Assumption:

- The number of processes is 100 and is hard coded. But the project can still run for any number of processes.

Requirement to run the project:

- **The machine that the code is tested on should install phoenix.**
- **Change the password from neel@123 to the password of the postgresql on which the code is tested.**
- **Change the password in the dev.exs file which is located in the config folder.**

```
# Configure your database
config :example, Example.Repo,
  adapter: Ecto.Adapters.Postgres,
  username: "postgres",
  password: "neel@123",
  database: "example_dev",
  hostname: "localhost",
  pool_size: 10
```

- **During the transaction, inputs should be provided as numbers and nothing else.**
- **In the transaction functionality, do not enter decimal values for amount and transaction fee and also do not enter random strings such as "asdf" etc in any of the four fields**
- **Sender and Receiver can take values between 1 to 100 and Transaction Fee can take 0,1,2,3,4.**

2. Functionalities Implement:

1) **Basic functionalities:** 2 Functionalities are implemented on the webpage.
    - **Mining:** By clicking this button a mining process will be started and the result will be shown by a table as well as chart.
    - **Transaction:** By clicking this button, two miners will be randomly chosen and transact a random amount of coins between each other. The result will also be shown in a table as well as chart.

3. Implementation Details:

1) **Base58.ex:**
    - This file contains functions that encode given data to base58.
2) **Base58Check.ex:**
    - This file contains functions that encode given data to base58Check.
3) **Blocks.ex:**
    - This file writes the abstraction of a block. And contains various methods for the task of proof of work and block validation.
4) **Bitcoin.ex:**
    - Endpoint of the program.
5) **Chain.ex:**
    - This file contains various GenServer methods for the task.
6) **Mempool.ex:**
    - This file implement the functionality of mempool.
7) **MerkleTree.ex:**
    - This file contains methods to build a Merkle tree.
8) **MerkleTreeNode.ex:**
    - This file defines the abstraction of the node of Merkle tree.
9) **TransactionInput.ex:**
    - This file defines the abstraction of the transaction input.
10) **TransactionOutput.ex:**
    - This file defines the abstraction of the transaction output.
11) **Transactions.ex:**
    - This file defines the abstraction of the transaction and various methods to build a transaction.

12) UTXO.ex:

- This file defines the abstraction of UTXO and implement the functionality of calculating balance.

13) Wallets.ex:

- This file defines the abstraction of the wallet and contains methods to sign and verify transactions.

14) WalletFunctions.ex:

- This file contains utility functions for implementing the functionality of wallets

4. Instruction to run the project

To start your Phoenix server

1. Install dependencies with `mix deps.get`
2. Create and migrate your database with `mix ecto.create and change postgres details as stated above
3. Install Node.js dependencies with `cd assets && npm install`
4. Start Phoenix endpoint with `mix phx.server`
5. If hex is not available install using “mix archive.install https://github.com/phoenixframework/archives/raw/master/phx_new.ez “
6. Now you can visit [`localhost:4000`](http://localhost:4000) from your browser.

5. Working of the project:

There are two buttons on the website page:

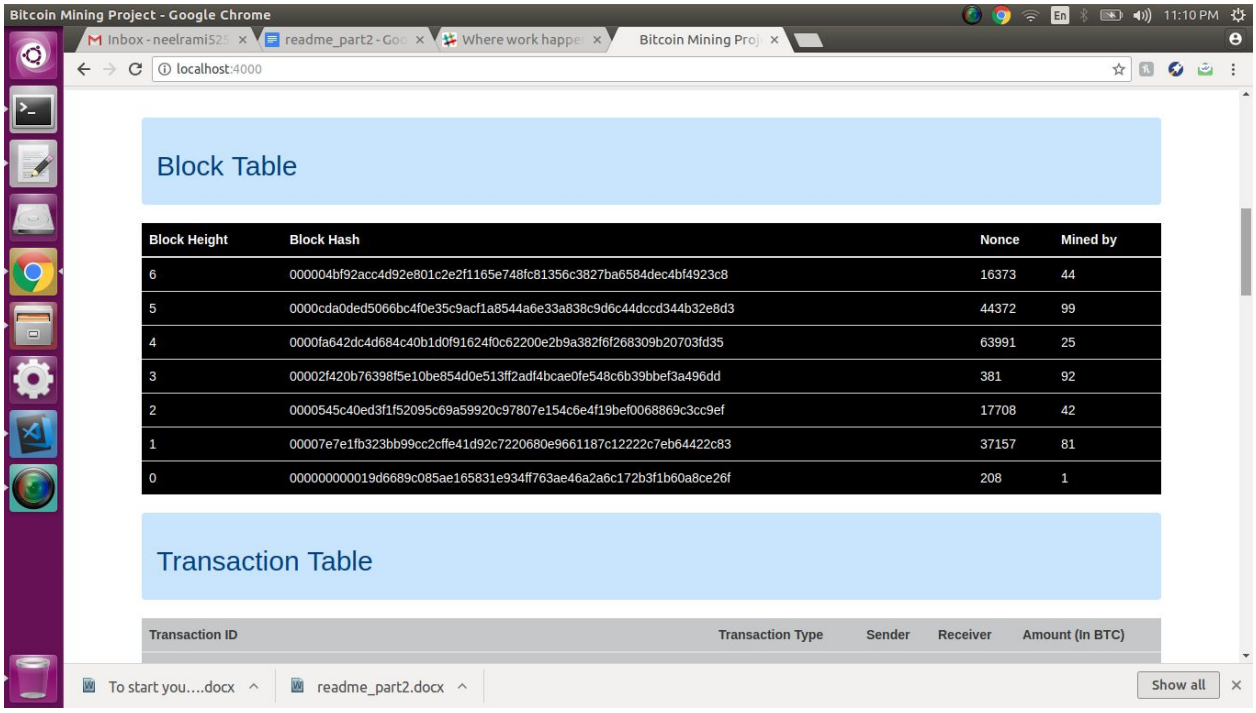
- **Mining**: When clicking on the mining button, a block will be mined and the information will be displayed in a mining table and chart below.
- **Transaction**: When clicking on the button, the user needs to provide four parameters: the process number of the transaction sender, process number of the transaction receiver, amount of this transaction and the transaction fee. Then, a mining process will be started and a new block will be mined and the information will also be displayed. Besides, the transaction information will also be displayed in a transaction chart below.

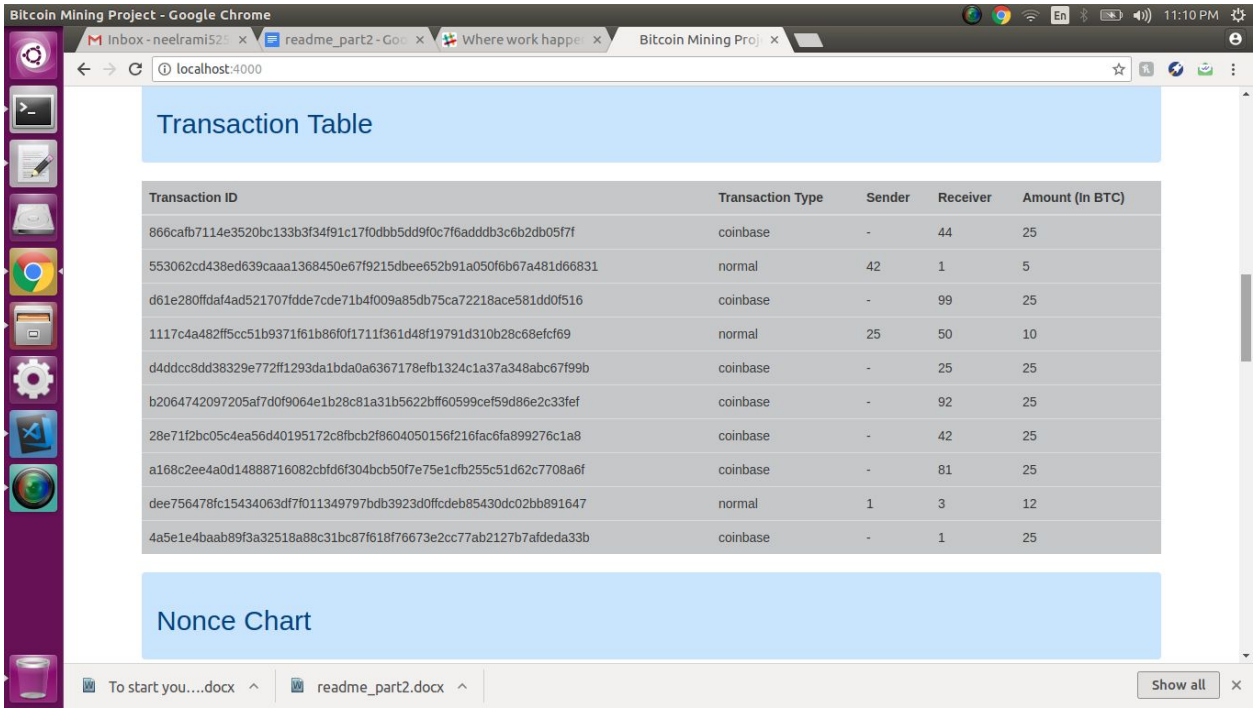
There will be 4 charts on the website page:

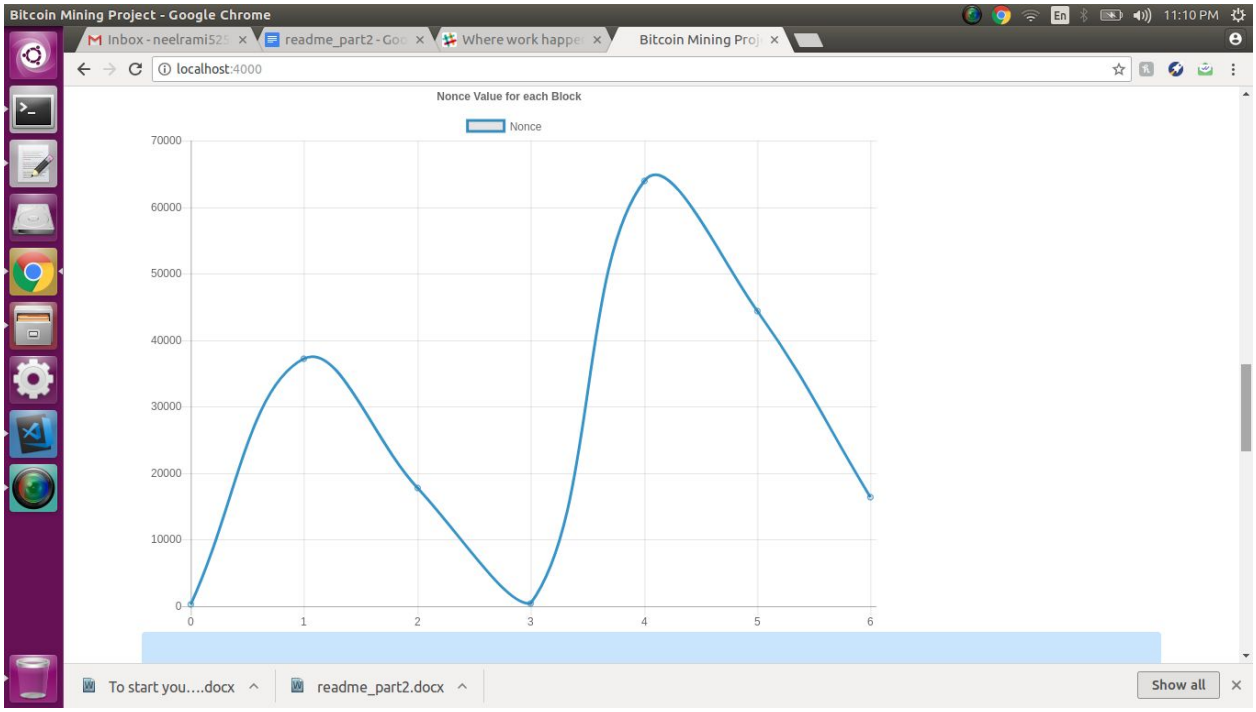
- **Chart 1**: Nonce for every block
- **Chart 2**: Number of blocks up to the present time

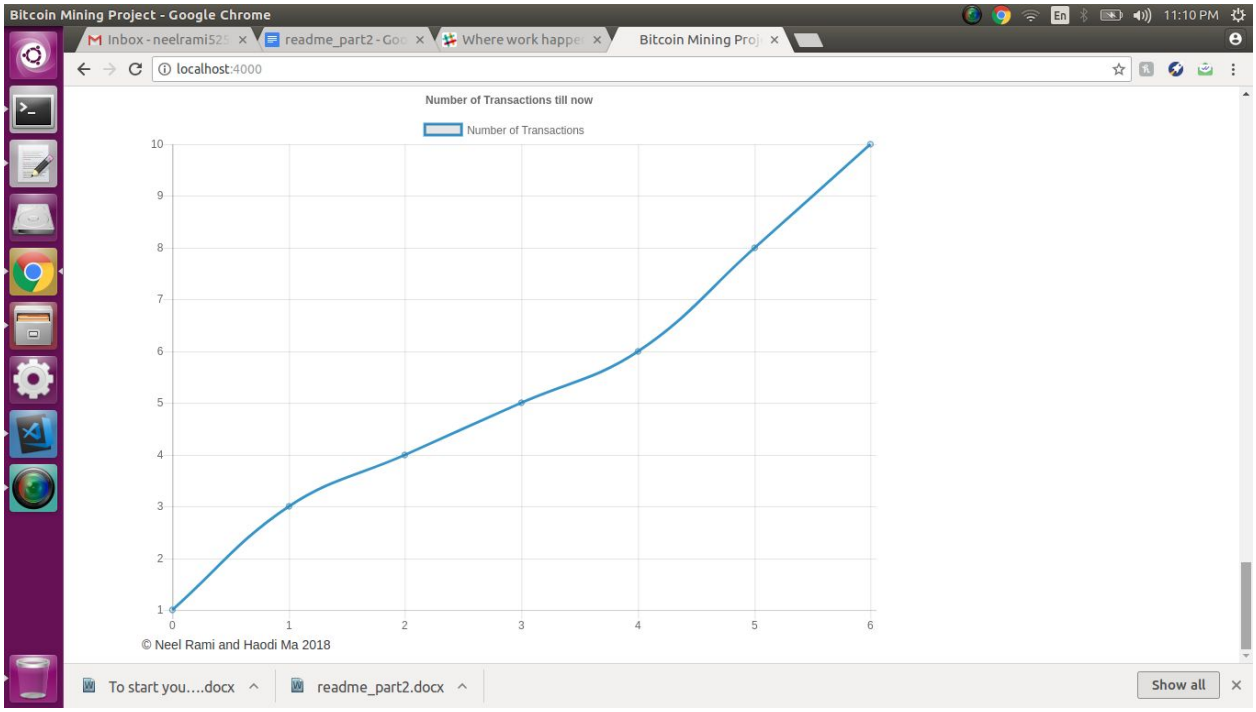
- **Chart 3**: Number of transactions mined up to the present time

6. Result






Bitcoin Mining Project - Google Chrome
11:10 PM
Inbox - neelrami52
readme_part2 - Goo
Where work happe
Bitcoin Mining Proj
localhost:4000
Nonce Value for each Block
Nonce
70000
60000
50000
40000
30000
20000
10000
0
0
1
2
3
4
5
6
To start you....docx
readme_part2.docx
Show all


Bitcoin Mining Project - Google Chrome
11:10 PM
Inbox - neelrami52
readme_part2 - Goo
Where work happe
Bitcoin Mining Proj
localhost:4000
Number of Transactions till now
Number of Transactions
10
9
8
7
6
5
4
3
2
1
0
1
2
3
4
5
6
© Neel Rami and Haodi Ma 2018
To start you....docx
readme_part2.docx
Show all