DISTRIBUTED OPERATING SYSTEM PRINCIPLES

Project 1 : Bitcoin-Mining using Erlang Actor Model

Sumanth Devarasetty - UFID: 57829275 ; Praveen Kumar Dande UFID: 62316196

Instructions to run the code

Case 1: Server and Client files are on different systems.

- Download server.erl and client.erl from the zip file.
- On the device with server.erl, move to the relevant directory and run the following commands in the terminal. In the command replace ipAddress1 with your machine ipv4 address.

  ```
  erl -name server@ipAddress1 -setcookie bitcoin.
  c(server).
  server:start(number of zeros).
  ```

- On the device with client.erl, move to the relevant directory and run the following commands in the terminal. Similarly replace ipAdress2.

  ```
  erl -name client@ipAddress2 -setcookie bitcoin.
  c(client).
  client:test('server@ipAddress1').
  ```

- Execute "client:test('server@ipAddress1')." after executing "server:start(number of zeros)."
- Server receives "number of zeros" input from the user.

Case 2: Server and Client files are executed in multiple terminals on the same system.

- Download server.erl and client.erl from the zip file.
- Move to the relevant directory where the files are stored.
- In one of the terminals run the following commands for server.

  ```
  erl -sname server -setcookie bitcoin.
  c(server).
  server:start(number of zeros).
  ```

- Commands to run on the other terminal for client.

  ```
  erl -sname client -setcookie bitcoin.
  c(client).
  client:test(server@name).
  ```

- "server@name" can be found in the terminal where the commands related to server file are executed.
- Execute "client:test(server@name)." after executing "server:start(number of zeros)."
- Server receives "number of zeros" input from the user.

1. Work Unit

The worker receives an input from the master with an upper limit on the total number of coins to be mined in this program, but there is no independent limit for the master or the worker till the max number of coins are mined.

The workunit is 3 in size. A single worker

- Spawns an independent erlang actor.
- Does SHA-256 encoding.
- Checks if the encoded String has the required number of zeros

We used a variable, "CONCURRENCY" i.e. number of parallel mining process, which varies for each machine depending on the number of cores in the machine, to get the optimal CPU Time/Real Time ratio.

2. Result of running the program for an input 4

- Sample Output (Hash Value, Nonce of last 5 bitcoins mined) and results for input 4 and a total of 20 bitcoins with both server and Client running.
- Input String used for all the computations is "pdande62316196"
- Input string + Nonce --> SHA 256 --> Hash

```
"00007f476b690ab08fcec916a31d363cc779c51c22ed2242206d917ebd10b584"   "wCsBXcGks2I="
"0000d13b6318f6f339a8fc19a85bc4432e0d331417a15d8d36f5ea40ebdul1f795" "UnPIfAm5Kx0="
"00007b04fa449c672036a9002c69deb3124bb892a75a8d8bdd427b744c26413b"   "9TdeAkARXkY="
"0000094bf5f39f85fa837574fcc398c8c5c2b310217d95d40962065f48a76904"   "q1NbKbt0H0w="
"0000d8472bad759e57119e6fe3572a0b8af7dcc0beab668729922bcfc1e8af0e"   "ixhrW98YGkE="
20 coins mined.

Cpu Time:2725  Real Time: 398  Ratio: 6.846733668341709
```

- Sample Output (Hash Value, Nonce of last 5 bitcoins mined) and results for input 4 and a total of 20 bitcoins with only server running.

```
"0000401ebdb2f518405a3715e1e9538e489dccf5a6c099f5eb097a3af33b97b4"   "3Z+4G6Wywas="
"0000c6acde20565265e091b80093d13b98dd6adecfd3cbc96cc6be5f5e9be3cf"   "azDXP3CnmYQ="
"0000410ad200b5828a7904672c1465a565919e32a5a20f25514ca747e12c7ddf"   "4uimakQ6f5M="
"000029aa970a4aaf9593d8a022a4ac617bc2886a55692bb14fc20131eb16d7dc"   "u7x1HCssxfA="
"0000ba921844434630370e5e75f78a18830384cf9c480014249ecd63760c1d82"   "YA6qsKnw9u4="
20 coins mined

Cpu Time:5333  Real Time: 1587  Ratio: 3.360428481411468
```

- In both implementations CPU Time/ Real Time is >1, therefore paralellism exists.

3. Max number of Zeros

The coin with the most number of zeros we generated has 8 zeros.

```
"00000000d3539a55ff08746734ee2e5242102760cd93dfd70654012cee71b6f5"   "Kr924acl7tg="

Cpu Time:8806517  Real Time: 2212164  Ratio: 3.9809512314638518
```

4. The largest number of working machines you were able to run your code with.

Implementation Details

We implemented our codes on 4 machines. It was the highest number of machines we were able to get access to and both the codes worked fine in all of the machines. We were able to implement server-client connection between all the pairs possible.

- Apple M1 chip: 8-core CPU with 4 performance cores and 4 efficiency cores
- Intel® Core™ i5-6200U Processor, 2 core processor
- Apple M2 chip: 8-core CPU with 4 performance cores and 4 efficiency cores
- Intel® Core™ i7-8750H Processor, 6 cores