Distributed Operating Systems, Project 1: Bitcoin Miner

Date: September 17, 2017

Group Members:

1. Sanket Achari, UFID: 71096329, sanketachari@ufl.edu

2. Sushmit Dharurkar, UFID: 14969578, sushmitdharurkar@ufl.edu

Instructions to RUN:

1. Make sure epmd daemon is running. Run epmd -daemon
2. Run server first. Run following commands from the directory which has mix.exs

```
mix escript.build
./project x
```

Here x can be any number from 1 to 63.

3. Run client after server got setup and started mining. Run following command

```
mix escript.build
./project1 x.x.x.x
```

Here in place of x.x.x.x insert the IP address of server.

To get results instantly i.e. within microseconds, we recommend to provide coins with less than 5 leading zeros.

If you give more than 6 leading zeros as input, you may have to wait for few minutes.

Instruction to Abort execution:

Press ‘ctrl + C’ in server console as well in client console to abort the execution.

Iterative technique to mine Bitcoin:

1. We have used actor model in the implementation to parallely mine the bitcoins. We have 3 modules

i. Project1: Checks whether input is number or IP and delegates work to the Manager.

ii. MiningManager: If the input is number it spawns the workers and keep on printing results sent by the workers. If the input is IP address then it spawns the node and contacts the MiningManager of server. Based on the leading zeros replied by server, it starts mining. And when client's workers get the strings whose sha256 matches with the leading zeros, it sends the result to server.

iii. Local Miner: This has the core logic of mining. Each worker executes the logic present in this module. This increments a given string and encrypts the complete string using sha256 hashing. Then if the encrypted string has given leading zeros then it sends the result to MiningManager. If there is no match, then it again increments the string and keeps on validating the result.

2. Base 36 Encoding: We have used Base36 binary to text encoding which represents the binary data in an ASCII format. So, when we say we increment the string, then the istring is encoded into Base36 and then its binary value is incremented and then we validate the corresponding ASCII value of new binary data.

This is the main idea of our mining technique.

3. Workload: We have given work of 100 million iterations to each worker.

Result for 4 leading zeros:

```
Sankets-MacBook-Pro:project1 sanket$ time ./project1 4
Calling Miners
sanketachari;99i
0000c741b451142e61eb4b887f15c37f8f3caa97e832684917360d0c6e8cdb71
sanketachari;sy8
000041a8b787d5365457afacdcb3341b9e53104135339bb698f077440084d2da
sanketachari;1nkkll
00001fc96698957dd28e969e9a296e85d793348e4c791ffb5f7752748b5f561c
sanketachari;1h02
0000b32385a990cad3f931a05629697532964db9190f8773e409eee74145c5b6
sanketachari;1u2m
000067d2bd83af8b70816e9038da3b3cb260646d221db61e1e95298fc742efbf
sanketachari;4ynrsz
00004110d41eb854e7884118d452feceaa2fd621d7a9ddc29ed690a665d8f9cf
^C

real    0m1.033s
user    0m3.245s
sys     0m0.122s
```

Real time = 1.033s CPU time = 3.245s

No of cores = CPU time / Real time = 3.245 / 1.033 = 3.14

This means that 4 cores has been used.

Result for 5 leading zeros:

```
Sankets-MacBook-Pro:project1 sanket$ time ./project1 5
Calling Miners
sanketachari;4ypv9d
0000088c6e3818b82035a2eb9538328aaac53a9babc004b64f3f2b362204bd38
sanketachari;6r1t
00000b6200cf428353b25bf52db17544323b850a88c83eba4e4471d2942d0214
sanketachari;4ysb2f
00000a47ac4f49e69d32be0d20587f8a6d511b8566ce2687ef43bb74c6d6dec5
^C

real    0m2.702s
user    0m9.675s
sys     0m0.181s
```

Real time = 2.702s CPU time = 9.675s
No of cores = CPU time / Real time = 9.675 / 2.702 = 3.58

This means that 4 cores has been used.

Details of Machine Used for above results:
MacBook Pro: i7 4 cores, 3.1 GHz ,16GB RAM

We were able to mine coins with 7 leading zeros.

```
sanketachari;dj2efv
0000000b22c95457891d9408fedfa9210bd82a6732ae517bce455e13654224de
```

We connected 2 machines :
MacBook Pro: i7 4 cores, 3.1GHz, 16 GB RAM
Lenovo Y50: i7 4 cores, 8 Hyperthreaded, 2.5 GHz, 8 GB RAM

We have also tested our code using **4 docker containers** each having 2 cores.