

COP 5615: Distributed Operating Systems Principles

Project 1

Bitcoin Mining based on Erlang Programming Language

Group Members

* Rahul Vikram Porwal UFID: 44590947 Email: rahulporwal@ufl.edu

* Pujan Narendra Kothari UFID: 70615094 Email: p.kothari@ufl.edu

Introduction

Bitcoins are a popular cryptocurrency. In this project, We will mine Bitcoins. The process of generation and circulation of new bitcoins is called bitcoin mining. Bitcoins are mined by using the computation power of hardware system. In order to mine bitcoin, We take a number of random strings which are hashed and check them to find a predefined number of zeros. The strings associated with the required number of leading zeros are the mined bitcoins. In this project, We are using Erlang programming language and actor model to achieve a solution which runs well on multiple cores.

Instructions to execute the project

- 1) Install the latest version of erlang.
- 2) Download the project1.zip and unzip it to any desired path.
- 3) Run command prompt at the file location with commands -
 `erl -name <anyname@youripaddress> -setcookie <anycookiename>`
- 4) If you want to run both the client and server, Repeat the above step on client and server side with the same cookie name.
- 5) Compile the erlang file on both the sides with following commands-
 `c(server_final).`
 `c(client_final).`
- 6) First, execute server side with the following command:
 `server_final:start_server(<leading no.of zeros wanted>)`
- 7). Second, Execute client side with the following command:
 `client_final:send_request("<server's side name>").`
- 8) Hereafter, The server and client systems will work to mine the bitcoins and display the result on the server side.

Results

1. Size of the Work Unit-

We determined the size of the work unit based on the number of leading zeros to find and upon the capability of the system on which the code is supposed to run. We are using Erlang's built-in function to find the no. of cores in a system and upon that information, we are spawning double the number of cores as workers. For example, On a 12 core system, The boss actor is generating 24 workers at a time and for 4 leading zeros, each worker works on 2000 strings to find the matches

2. The result of running our program for input 4 -

```
B> rahulporwalA36483796C73AA83BB3C764DE0585A2987AD 00001b491ddcd8d75fd8ab8aced04862ee7f49b1c734a883aeb0d5a6b6488d24
B> rahulporwalA3A6604AF7F878A3E7C139A05C6315585A01 00005c42c31867e7211adf9a8f64b0c81d818a19e7b192f6aa7f32e01f572706
B> rahulporwal268EBAEBED9450A37054ACD83BC642FBBF8B 0000e3704bd8bb50a871a0f89d0d3a816f4000c18d89cbab144ab310adddd46
B> rahulporwal8CDA38221D086BAC523274881204182AEAE5 000092eb742cf006364e57bf1070d931e1dfa2148d28657ab56d6f184b7d0906
B> rahulporwal28C53480E7547FA45FA7B73CED39431B420F 00009ba45a6114a003f225159a9f197e4cb39271806ba7df0bbf7673581ccb48
B> rahulporwalD624E1AB96F698A4427105A24283E0AB8CBE 000036246fb25ebad240c63af7d0d8645739499bdbc360e399c7e6584e3675ed
B> rahulporwal8953E23A933E337086F1CE8ED8E28B4EABF5 000095334d452548092adda14b0a0a98ccf26563ad11a643846e6bd9680d95b
B> rahulporwalFC2EB35827C881F9F2917B598E2370EFD6DF 0000e0fb67e364226c4bd3af4267690e44b17d171ea07952821ad6839cea5db9
B> rahulporwalD3D3BB6CB0198AF0BC1377BE9631D31C6B56 000044f75b524276c916bfbf4e281634cfc3d9a5a28b9c0e90ba67cc23f79d8b
B> rahulporwal33B4ECC7616F235700A0BA71883B8C073B41 0000118ef2c3536262c0e7357dfe6a37f768d80bb80fb8f55d3451287042bbbf
B> rahulporwalA813040BCE77BCD34D6F381872411E006539 0000f1675154aaa95b4ef22ca0ba53c28ea8b3c9e541f1fa0c8b976a31de83bb
B> rahulporwal8D09059E33D549AB9F1F81CDF0876F7F316F 0000a530ad02c7df4908bcb839ed9d662cddbabb2ea37b667adb28a0372208fb9
B> rahulporwal4B0551D55312B9F668481A3B778EB0A46908 0000e87f93812d69b31fc4166fef2729d1eae878b24ba89383370d0f1e0b4dbf
B> rahulporwal46C146468BB8E22DDAC122009C465E744E0 00001852396fc137eee201341b70e381b64f20b0495d1342bd2b1a746288a87b
B> rahulporwal64DB27DBBDF48B0733581C8C7B0749FFC4AF 00005e0287bd3b3626824975c44d530427020fe8d60cb7ce423461127c50add4
B> rahulporwal35A88AFB0A31C41E78F7749276FF21566C63 00003b29123fd69d4e5e91f75c1b77a5c037855da5a9dbc3c2abf948ac48b6fd
B> rahulporwal63EAAED3175ED45F2340F883C0E54893999 00000962f64714ba50ed12e842aad50b24ef14ee763fc2cf1e681b502eded47d
B> rahulporwalE5E32CA07C7B13F7414DA9170E3C20DF65E5 00005b96df0251e729496e5055912dd2fd4c4890f208e22c51fd26dd786dacec
B> rahulporwal8D885C41BE812EBF8E79F2D16F9C9F91D168 00002c68a4c00e4c40f4d10f3b5d6589847fd1ce3f4a24493970741cfe66844f
B> rahulporwal32BA407F8B565B43BE272CD0BA7307001663 0000597b30b2aa24fb6c0a25d40235fdc02abc001b0177124ba20d5b18601352d
```

3. The running time for the above :

The stats below were computed on a 12 core system, we are spawning 24 actors at a time, and each actor works on 2000 random strings.

```
(p@10.20.203.128)3> rahulporwalB8FC038E8EA9C9381129B8B4ED9A6D4E28D1 0000c5c319693e7ae7f1f33d0fcaa7e7cd4806fe3d26fb32fafd1776cc24bb3
(p@10.20.203.128)3> rahulporwalA5DC1F52735EADF3B80F91E3143633F8C6A1 0000e275d6e59add1e8993db9ebb0a8e379c9e78ddd53dc90a26a175f70bd59
(p@10.20.203.128)3> rahulporwalCA8D8678C016872CC4DB7CBA48F3498E28E5 0000ba051d4ce9f57df9943b6b826daa20a7037cde760dddec203cfd66627fabb
(p@10.20.203.128)3> rahulporwalAC5EF4739EA6D5902A76817C0AA020F63631 0000845c3cddedfee5044365c46107ae7e6ee9dfea44189867db05929849d6fa
(p@10.20.203.128)3> rahulporwal305518984CFD6901A06E17AF3135DCBD8166 00009ff90a70bae0b15b1957727ee90dd906c5a76eb7524b2500d7c7935071b9
(p@10.20.203.128)3> rahulporwalD8A47CFDCFD26316E04F681839971937AA1E 0000b594ed16b4cb0ed14addb9592709124413ff65e48ce64b9688deec0fc171
(p@10.20.203.128)3> rahulporwalB277BB477F17C120A086C51F213532237406 00006725df262bdf8baabacdce48bdef0c04f46bedd76ca23d2fce14ccbe23d
(p@10.20.203.128)3> rahulporwal5D89EEF1A4F56EF31423CD6A258850544EA0 00000c3cdd31dc7b284a7aaf054f0f55b86396e396e766678c2337063c002b83
(p@10.20.203.128)3> rahulporwal9B140CDD1BA1F6F88BCA40E1AC9DE0AAB1DA 00006f81bd3140239894c43be5b464b984ecbac6bd2933f8bc2d73f1a97c1f60
(p@10.20.203.128)3> rahulporwalAE89A7D6838521AB054687F719D760DD4324 00002f61cb43d4112247cd1bdb669ac9d75a9fac9ae8dafb0d3cd54cfaf9f49
(p@10.20.203.128)3> rahulporwal4CB8BF1E6ECCB5570313273F97D577D21738 00002e9994348d617e390466bfa487a620f5b5df3f85531afb2ca9da35557d0
(p@10.20.203.128)3> CPU time: 86094 Real Time: 7284
(p@10.20.203.128)3> Ratio (CPU:Real): 11.819604612850082
```

4. The ratio of CPU time to REAL TIME i.e, number of cores that were effectively used in the computation -

We got a ratio of 11.819 for a set of 24 actors which were spawned on a 12 core system at a time.

5. The coin with the most 0s we managed to find -

The most 0's we managed to find were 6 leading zeros.

```
4> rahulporwalD1DE92768114024EF9B304902F12AC1E96E0 00000032d2a950b4ffee99f704e97f70f5d26c9ffecb2b6615d3a403bcd5ae5d
4> CPU time: 161641 Real Time: 13606
4> Ratio (Real:CPU): 11.880126414816992
```

6. The largest number of working machines we were able to run our code with :

We were able to run our code with 2 working machines.