

## Distributed Systems Project 1

### BitCoin Mining

---

#### Group Details:

##### Members

1: Manjary Modi ,UFID: 38408368, manjary.modi@ufl.edu

2: Rameshwari Obla Ravikumar,UFID: 16161302, rameshwari.oblar@ufl.edu

#### Steps to run the application on the Server machine:

1. In shell, navigate to the directory where mix.exs exists

2. Run command

>mix escript.build

>./project1 <number of leading 0s>

After step2, the server program will start mining and the generated bitcoins are displayed on the server terminal.

Concurrently, a server node starts that looks out for possible worker connections recursively.

When worker machines are available, they establish connections with the server and the server application automatically creates and schedules mining jobs to that connected worker as when they contact the server. The worker machine performs the mining process and the generated bitcoins are displayed back in the server terminal.

#### Steps to run the application on the Worker machine:

1. In shell, navigate to the directory where mix.exs exists

2. Run command

>mix escript.build

>./project1 <Server Ip Address>

After step2, The worker program will start a node at its end and establishes connection with the server machine. The job gets scheduled instantly and the mining process happens in the background, that is, as soon as the coins are generated it is sent back to the server and nothing gets printed on the worker terminal.

As per the requirement, the mining process happens indefinitely in all the machines. When the server is stopped, the mining process in the worker machines also get stopped.

Note: The workers and the server machines should be connected to the same LAN for the application to work as per the requirement.

#### Explanation for each of the functions used in the application

##### 1. Main

Used to run the code via escript.build. If the argument is an IP address, it performs worker functionalities else performs server functionalities

## **2. startServerNode**

->Spawns a process to start the server node

->Concurrently, spawns multiple processes to mine bitcoins in the server machine

## **3. startNode**

Starts the server node

## **4. waitForConnection**

Recursively looks up for connected workers

## **5. distributeWork**

Distributes mining jobs to the worker node

## **6. listen**

Stops worker jobs when the server node goes down. This is needed to ensure that workers stop computing as soon as there is no interface available to send the results.

## **7. startClientNode**

Starts the Worker node.

## **8. startConnection**

Used by worker node to establish connection with the server.

## **9. serverMining**

Random input strings of a fixed length are generated and their hashed outputs are checked. If the output contains the required number of leading zeroes, the hashed output is displayed to the server terminal. The whole mining process happens indefinitely.

## **10. workerMining**

Random input strings of random length (different from server input string length) are generated and their hashed outputs are checked. If the output contains the required number of leading zeroes, the hashed output is sent back to the server terminal. The whole mining process happens indefinitely.

## Requirement Specifications

### **1. Work Unit:**

As per the document, a work unit is the number of sub-problems that a worker gets in a single request from the boss.

That is, number of jobs that a worker gets in a single request from the sever.

For our application to get the optimal performance, the server has to spawn atleast as many jobs as the number of cores a worker machine has.

->We tested with a cluster of 3 quadcore machines, one acting as a server and the other 2 acting as worker nodes.

->We varied the number of jobs distributed to each of the two worker machines and noticed the CPU utilizations. When the number of jobs were greater than equal to the number of cores in the machine, the CPUs were fully utilized.

#ServerProcess	#WorkerProcesses	RealTime	UserTime	SystemTime	CPU time(U+S)	Ratio
4	4	0m32.948s	2m2.132s	0m1.932s	124.064s	3.765
8	8	1m17.893s	5m0.172s	0m2.176s	302.348s	3.88
1	4	1m5.792s	2m10.600s	0m0.748s	131.348s	1.996

## 2. Result for running the program for ./project1 4

rameshwari.oblar;b2C1nQ-6

00000361bc5be79a9eb3cfb91971e6547f1e7aef588709dd5933d7e1e61a249e

rameshwari.oblar;As5aPcWP

0000a4aaf4928942adaf5cb547f9c62504d7a0d5a50d5a0727db64c6ef174181

rameshwari.oblar;TZznGExM

0000c56f3db91d40ff1ab1f8bb348c11d9318d0262faa818a069a506174ef6c3

rameshwari.oblar;Y0NdY9fA

0000399bd6d6c5f3532b1a1e771df811e45f45b151bec159cf545c067eacd375

rameshwari.oblar;nttoNDcZ

000085fb98e9287f72aef08f1280d15c7a3eb6ac1f7dd2037fd76878bef9938c

rameshwari.oblar;c97F2kCg

0000d135166a1b01faa8ba513ba5183c01efa9e887690c183aba89b6fa53fe2d

rameshwari.oblar;-m5mXw2b

00004d287359d8e2579b48d5683d9ad194f02c7df799d63cb0ed45156b28092f

rameshwari.oblar;CBK79XCb

00008db3ee4a23ecac708ff29b74870371d1918714aa910e3d18e10a388615bf

rameshwari.oblar;rcXfpl\_V

0000ab127029adb6f1a4132543b1daa65dbde12b006f3ba7ab8a9dfb064a4f6f

rameshwari.oblar;C9KyvV4x

0000f4164c8ed7c1c7534c8b6a5d3d910b68b6b2a095eadb277f50c3ce345341

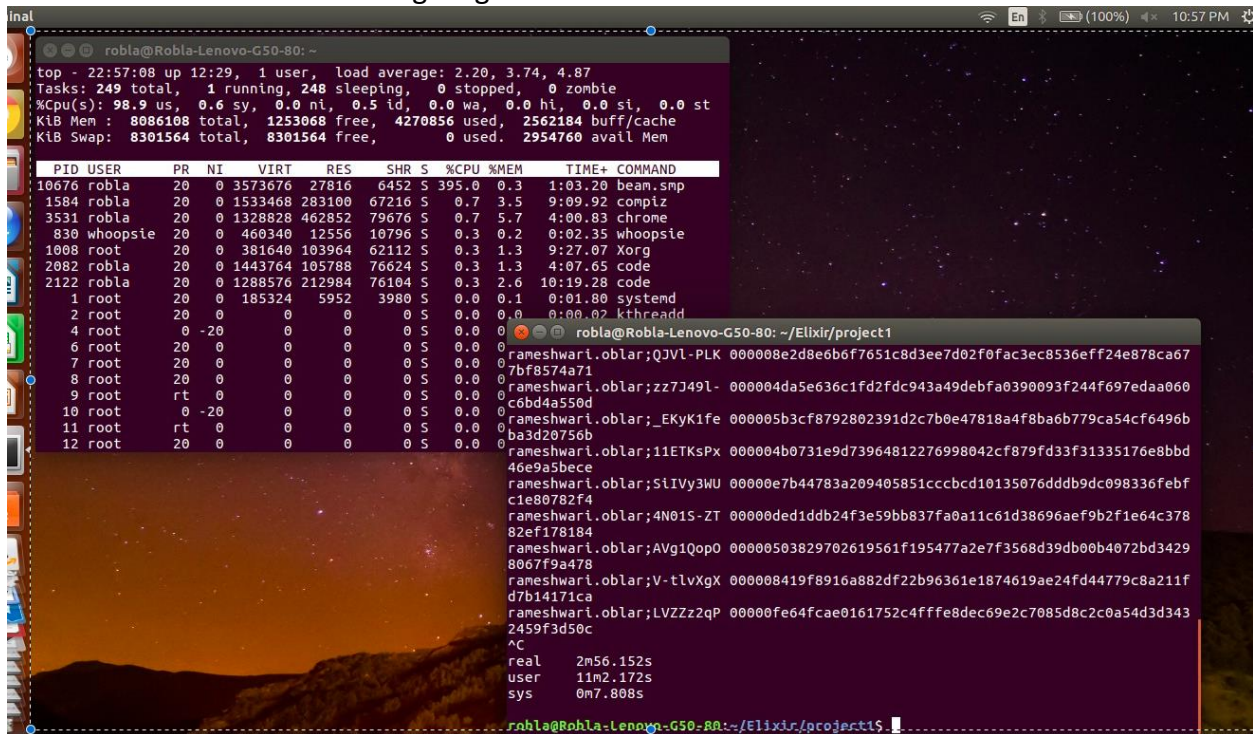
rameshwari.oblar;LNwukvX3

0000d9d6e8ad9acab5e20b5c370d9254f01a4ab9454a321403b32a1a360bcc57

```
Terminal
rob@Robla-Lenovo-G50-80: ~/Elidj/project1
rameshwari.oblar;ZHUcHTn 00004ff780deb2073acaab8d1917e3193a8dc892aad61348f8b92a3da3df7
rameshwari.oblar;vt39_e99 0000dd550ba036135ad0b9e0c60e7af9f3a412c10e1eebdf34af3d04ef4e9
rameshwari.oblar;5FfeaFDL 0000e631dda4eb145d5d35ca1157a5c86a5d5ac7f7850e0ab23e8df288e4ef4e
rameshwari.oblar;4koQuV3z 0000e0bae4193d3f51025c104449340b961fba02997b060153392gef7d58f56
rameshwari.oblar;jl5p0R3x 0000992649a47c5b7e0ca9301ee9878379528cc119970c30f2cd07442b098
rameshwari.oblar;0tuk5fj5 00005127874a7875dd26214fc19138a61c6fac3bb572fe22c01876f83e3425f
rameshwari.oblar;IQNS1Jfp 00006758112aa5b4f17ed734a0b73aabce240ba9f2ba39347b2f5eb4cf9cf2be
rameshwari.oblar;Cu4d7ryw 0000848afe06c341f0b2ac97d396eb5c7524dc1d034f6b353bd5cb79448c2232
rameshwari.oblar;QEH9Wp90 00001b0e70e0764d976e13a536716318f0e7b2c6af8a2cc4d82c23493203215
rameshwari.oblar;3besRtl1 000052798c275f249c5e3308f34c94aaa4d1ee9ee21232bd4cb06e3c5d85f0
rameshwari.oblar;gltty0ER 00004f1057c7f6066161dae7f8f814f0b143ed9073ae9cfa58607aa4cffe37
rameshwari.oblar;313woaX7 00008246dd748b90f3631d3d7c1eaa396c3ede9bc6c75827439ac3f1bb00e919
rameshwari.oblar;vgCzJ-Za 00008c071a8f3bc9f8ce77f7f6ebffcab420eaa81e14d465263a0a8991f56f0
rameshwari.oblar;AHFERIQf 00002f9812f6d384fc04f0e0840de6d0cd24d0c156d0c041cd8fb8f43ba70
rameshwari.oblar;jhY30wh 00006073139b1f1b0ed2c275ec7f92550f08f59a1533bed3541e0e0991f05e6
rameshwari.oblar;SnsEFhco 0000bcfbcd32ebd896057611a89fed238a245eeff3fb57e397e91a059d14a2bb
rameshwari.oblar;3sqFYwCA 00004f438c2d610dbed795f1f9e0613b6207f92316646aedb14cecaabeb0db36
rameshwari.oblar;rg8WNIJC 0000853a020f1846796d29f1006bd480fbd9c7c60c0d06c0c7ae9981d465d57
rameshwari.oblar;CktJ2MLQ 00006ae8ca2a417021f81d05d9e721d87d93cd78d1a11da923c78c746856c71
rameshwari.oblar;wXLY-jw 0000576a22df86d77700ab169c530246ed1d8f3f199099b025b1cadba9e8a
rameshwari.oblar;QGB2LE4Q 0000f97ba4da7b510bee48dc30eac8ef6ec2e527ac3b32877f8e8c5b071b3a
rameshwari.oblar;jd-2MEWIN 000027f4fa29e09f6895c713e77250e5aa670621ad1e101940b71b7f2d84d63
rameshwari.oblar;10FWKRLP 000060994400a704dc28911cbca4054f4bbd0aa169879e9321e1bd1746c45e4
rameshwari.oblar;89dnnjvL 00001a98573484fc7321eacbb09a8ff9f0e718a5c4e3bc25b26d38dd4e973
rameshwari.oblar;07AueXf6 0000576a22df86d77700ab169c530246ed1d8f3f199099b025b1cadba9e8a
rameshwari.oblar;GHtr7XaQ 00002482ce05ba977ed972574cbfead0506f5e6b32ce99657799e4d1e981f82
rameshwari.oblar;wxbfXW-S 000017a0198e24851be545d612470a3ac1346539db9d8c5a01028ae347338ff
rameshwari.oblar;gvz5xC28 0000f9c9dc1b944d1552e75c83c537f3d2873ce9ba7de79f0fb92eb86284c6c2
rameshwari.oblar;nbmXJLC 0000edf0f17be78989c9369a8e74f20b4d0e9504643544df5f4a60c11b2520
rameshwari.oblar;v20ZcX58 00007424da98de45f241cb5350e0dc9d5118f50c4dc59a489a6d5db08ef
rameshwari.oblar;KKGjv3Z5 000068459bfbb488462fa420d903bed2aa155ed3211a61feb5ab85a582c81
rameshwari.oblar;v4fJfT82 0000ec5a61c4e91fa27a2ee3dac613b9f6cee4552b314409a294d109f2ed287d
rameshwari.oblar;BryDko0m 0000a28a11b88be300e3fd28ef752352bda996da152339ad367b81aa26741d
rameshwari.oblar;qgnz2BCC 00001183544744c4952cf9253f0e08036170ded10a0777f70322aacc41854c0
rameshwari.oblar;1z2e9-HW 00003126c405d7A81137eb0061cfbd2302708540e40a33e9d7c4c03cd8530d
rameshwari.oblar;87GpS075 000033a41b47c7384f7a39d4c62bb330e0ef63c563b5197c6fe32573948f666
rameshwari.oblar;TFEporEC 0000617d5b5ea60cd93c03f3d8a8f8e959d13e865817bef06e15e9bf2aa1e47
rameshwari.oblar;h0cGAT4 000091682e4eac7adc4ccdf3cdd7d101ff781d639be976bf9489f94a369a0
rameshwari.oblar;-o-BXNEd 0000a49ac3f19ef0427d42d0aacc7fbo7ad9cc5e45ebf69c321d1088b545c86
rameshwari.oblar;-dWpV3oh 0000f48f833187ef27a70c3b205c55ccf8a4eb7f13215b060e1c206d712c
```

### 3. Running time for ./project 5

Below is the screenshot for string length 5 and CPU time to Real Time = 3.795



The screenshot shows a terminal window with two main sections. The top section displays the output of the 'top' command, showing system statistics and a list of running processes. The bottom section shows the output of a program that generates a list of strings, each preceded by a tab character and a string of 15 characters.

```
robila@Robla-Lenovo-G50-80: ~  
top - 22:57:08 up 12:29, 1 user, load average: 2.20, 3.74, 4.87  
Tasks: 249 total, 1 running, 248 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 98.9 us, 0.6 sy, 0.0 ni, 0.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
KiB Mem : 8086108 total, 1253068 free, 4270856 used, 2562184 buff/cache  
KiB Swap: 8301564 total, 8301564 free, 0 used, 2954760 avail Mem  
  
  PID USER      PR  NI   VIRT    RES    SHR   S  %CPU  %MEM    TIME+  COMMAND  
10676 robila    20   0 3573676 27816  6452  S 395.0  0.3   1:03.20 beam.smp  
1584  robila    20   0 1533468 283100 67216  S   0.7  3.5   9:09.92 compiz  
3531  robila    20   0 1328828 462852 79676  S   0.7  5.7   4:00.83 chrome  
830   whoopsie  20   0 460340  12556 10796  S   0.3  0.2   0:02.35 whoopsie  
1008  root      20   0 381640 103964 62112  S   0.3  1.3   9:27.07 Xorg  
2082  robila    20   0 1443764 105788 76624  S   0.3  1.3   4:07.65 code  
2122  robila    20   0 1288576 212984 76104  S   0.3  2.6  10:19.28 code  
1    root     20   0  185324    5952   3980  S   0.0  0.1   0:01.80 systemd  
2    root     20   0      0         0      0  S   0.0  0.0   0:00.02 kthreadd  
4    root     20   0      0         0      0  S   0.0  0.0   0:00.00  
6    root     20   0      0         0      0  S   0.0  0.0   0:00.00  
7    root     20   0      0         0      0  S   0.0  0.0   0:00.00  
8    root     20   0      0         0      0  S   0.0  0.0   0:00.00  
9    root     rt    0      0         0      0  S   0.0  0.0   0:00.00  
10   root     20   0      0         0      0  S   0.0  0.0   0:00.00  
11   root     rt    0      0         0      0  S   0.0  0.0   0:00.00  
12   root     20   0      0         0      0  S   0.0  0.0   0:00.00  
  
robila@Robla-Lenovo-G50-80: ~/Elixir/project1  
rameshwari.oblar;QJVL-PLK 000008e2d8e6b6f7651c8d3ee7d02f0fac3ec8536eff24e878ca67  
7bf8574a71  
rameshwari.oblar;zz7J49l- 000004da5e636c1fd2fdc943a49debfaf0390093f244f697edaa060  
c0bd4a550d  
rameshwari.oblar;_EKyK1fe 000005b3cf8792802391d2c7b0e47818a4f8ba6b779ca54cf6496b  
ba3d20756b  
rameshwari.oblar;11ETksPx 000004b0731e9d73964812276998042cf879fd33f31335176e8bbd  
46e9a5bece  
rameshwari.oblar;SlIVy3MU 00000e7b44783a209405851ccc8cd10135076dddb9dc098336febf  
cie80782f4  
rameshwari.oblar;4N01S-ZT 00000ded1ddb24f3e59bb837fa0a11c61d38696aef9b2f1e64c378  
82ef178184  
rameshwari.oblar;AVg1Qop0 00000503829702619561f195477a2e7f3568d39db00b4072bd3429  
8067f9a478  
rameshwari.oblar;V-tlvXgX 000008419f8916a882df22b96361e1874619ae24fd44779c8a211f  
d7b14171ca  
rameshwari.oblar;LVZZz2qP 00000fe64fcae0161752c4fffe8dec69e2c7085d8c2c0a54d3d343  
2459f3d50c  
AC  
real    2m56.152s  
user    11m2.172s  
sys     0m7.808s  
robila@Robla-Lenovo-G50-80: ~/Elixir/project1
```

### 4. The coin with the most 0s we managed to find is 8

rameshwari.oblar;y\_H\_R4Gm

00000000abc35575e405b9b2c79c4de458137867805fc28ba7dee24be802e993

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

3 quadcore machine