

Group Members:

Name	UFID
Neha Singh	41148924
Ruchi Desai	75103513

1] Size of the work unit:

- The work unit in our project is defined by nrOfStrings multiplied by nrOfMessages.
- Each worker that gets Work message from Master is asked to generate nrOfStrings number of random strings and the worker will return all the valid bit coins that it found by hashing these random strings.
- Master will create worker router that generates nrOfWorkers number of workers and will evenly distribute nrOfMessages number of Work messages among these workers in RoundRobin manner.

The values of the above variables that resulted in best performance are as follows:

nrOfStrings: 100

nrOfMessages: 15

nrOfWorkers: 10

2] The result of running program:

Please find the result of the program in output.txt file in the zipfile.

3] Running time for the project:

Real Time: 8m31.008s=511.008s

User Time: 19m42.586s

System Time: 0m30.493s

CPU Time = User Time + System Time=1213.079s

Ratio = 2.373

4] The coins with most 0's we managed to find:

Largest number of zeroes in the mined Bitcoins= 6

5] The largest number of working machines we were able to run our code with (where we could record the time) =1

We were also able to run the code in the following scenarios (using SBT):

- A. Master executed on one core on which it had its local pool of workers. Two workers on two different cores were able to connect with the master midway between the execution of the code and return result to the master.
- B. Master executed on one machine on which it had its local pool of workers. One worker on a different machine was able to connect with the master midway between the execution of the code and return result to the master.

Steps followed to execute on Linux machine:

To compile:

```
⇒ ./scalac -classpath "/cise/homes/rvdesai/akka-2.3.6/lib/akka/akka-actor_2.11-2.3.6.jar:/cise/homes/rvdesai/akka-2.3.6/lib/akka/common_2.10-1.0.jar:/cise/homes/rvdesai/akka-2.3.6/lib/akka/akka-remote_2.11-2.3.6.jar:/cise/homes/rvdesai/akka-2.3.6/lib/akka/netty-3.8.0.Final.jar:/cise/homes/rvdesai/akka-2.3.6/lib/akka/protobuf-java-2.5.0.jar:/cise/homes/rvdesai/akka-2.3.6/lib/akka/config-1.2.1.jar:/cise/homes/rvdesai/akka-2.3.6/lib/scala-library-2.11.2.jar" /cise/homes/rvdesai/public_html/DOSProject/BitCoinMinerFinal/Remote/src/main/scala/MasterHandler.scala
```

To run:

```
⇒ time ./scala -classpath "/cise/homes/rvdesai/akka-2.3.6/lib/akka/akka-actor_2.11-2.3.6.jar:/cise/homes/rvdesai/akka-2.3.6/lib/akka/common_2.10-1.0.jar:/cise/homes/rvdesai/akka-2.3.6/lib/akka/akka-remote_2.11-2.3.6.jar:/cise/homes/rvdesai/akka-2.3.6/lib/akka/netty-3.8.0.Final.jar:/cise/homes/rvdesai/akka-2.3.6/lib/akka/protobuf-java-2.5.0.jar:/cise/homes/rvdesai/akka-2.3.6/lib/akka/config-1.2.1.jar:/cise/homes/rvdesai/akka-2.3.6/lib/scala-library-2.11.2.jar" /cise/homes/rvdesai/public_html/DOSProject/BitCoinMinerFinal/Remote/src/main/scala/MasterHandler.scala 4
```

To stop execution:

⇒ Press Ctrl+C.

Steps followed to execute using SBT:

(Refer folder BitCoinMinerFinal in the zip file for the code)

Our project has the following hierarchy:

Remote

```
|
|-src
| |
| | |-main
| | |
| | | |-resources
| | | |
| | | | |-application.conf
| | | |
| | | | |-scala
| | | |
| | | | |-MasterHandler.scala
|-build.sbt
|
|-lib
| |-common_2.10-1.0.jar
```

Local

```
|
|-src
| |
| | |-main
| | |
| | | |-resources
| | | |
| | | | |-application.conf
| | | |
| | | | |-scala
| | | |
| | | | |-WorkerHandler.scala
|-build.sbt
|
|-lib
| |-common_2.10-1.0.jar
```

Common

```
|  
|-src  
|  |  
|  |-main  
|    |-scala  
|      |  
|      |-Common.scala  
|-build.sbt
```

To run the program:

1. Open command prompt on one machine and cd to Common folder
2. Type sbt package
3. Go to Common ->target->scala-2.10 and copy Common_2.10-1.0.jar file and paste it under lib folders inside both Remote and Local folders.
4. cd to Remote folder.
5. Type sbt "run 4".
6. Now open command prompt on another machine and cd to Local folder.
7. Copy ip address displayed by program running on first machine and paste as an argument in the following way:
Sbt "run 4 <ipaddress>"
8. To stop execution press Ctrl+C.

Other Specifications:

On linux:

Scala version = 2.11.2

Akka version = 2.3.6

With SBT:

SBT version = 0.13.6

Scala version = 2.10.1

Akka version = 2.1.1