# DOSP_Project1

To run the program with no remote actors

```
dotnet fsi Program.fsx <Number of Zeroes you want in the output>
dotnet fsi Program.fsx 5
```

To run the program in a distributed manner, make sure that all the machines are under one LAN (use UF VPN if you want to connect alot of them), First run the server

```
dotnet fsi Server.fsx <Number of Zeroes you want in the output>
dotnet fsi Server.fsx 5
```

Then run the client

```
dotnet fsi Client.fsx <server ip:server port>
```

The server(master) has two jobs

- Mine coins
- Pass messages to clients(slaves)

The client(slave) has only 1 job

- Get the message from master and start mining

Both the client and the server again use their own local actor system to mine the coins

1. The size of the work unit for each of the slave actor is N^2 where N is the maximum length of the random string = 16. The reasons for choosing this work load are
    - For a string of length 16 which has a character set of more than 40, the total number of strings are almost 1315041316842168115200000. So, there is almost no way that two(even a hundred) randomly picked strings have the same characters.
    - For the random guessing to work, each worker has to guess alot of strings instead of wasting time by passing messages. However, there's a possibility(small) that there is more than 1 result in a given work load and we do not want workers to waste time calculating after already finding an answer.
    - The work load of N^2 was taken after several tests with other work loads such as N, N^3, N^K.
2. For the result of running my program for the input 4 is

```
Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0
Number of Zeros = 4
Number of Workers = 8
Max String Length = 16
akka://my-system/user/slave1
akka://my-system/user/slave2
akka://my-system/user/slave3
akka://my-system/user/slave4
akka://my-system/user/slave5
akka://my-system/user/slave6
akka://my-system/user/slave7
akka://my-system/user/slave8

String : harishrebollavarQcTzb1NVWlSpk0o
Hash : 00002f0ca7fe3a3006447b64c2a844cf32b013f63d4c71f3e1bf57252808fede

Real: 00:00:01.197, CPU: 00:00:06.437, GC gen0: 260, gen1: 3, gen2: 1
```

   String : harishrebollavarpQ2cD67ATpn7z Hash : 0000fd6e2a3fceeca135ba0a6f57b192f61800d7dad406470bad97552950f46a
3. For smaller inputs like 4, the ratio of CPU time/Real Time is close to 5, but for largest inputs like 6 or 7, its close to 6. My machine has 8 cores.
4. The coin with the most zeros is
    - `String : harishrebollavarfDmt4 Hash : 0000000e46766ceef8bbd0f31eb8e2b97359e0ac6d90a650989620a0c1a53bad`
5. For the distributed model, there is no limit on the number of machines. However, I've only tested my code on 3 machine with one of them being the boss machine and the other 2 just slaves.