Group Members -

- Bhakti Armish Kantariya - UF ID: 3928 1182
- Purva Sanjay Puranik - UF ID: 6017 3612

Steps to run -

Establishing a connection between the server and remote worker:

- **Server-side**

1. Run the following command on the server console:

   Move to server directory:

   ```
   cd server
   ```

2. Identify the server IP address and use it in the following command:

   ```
   erl -name server@<serverIPaddress> -setcookie cookiename
   ```

- **Remote Worker side -**

1. Run the following command on the client console:

   Move to the working directory:

   ```
   cd worker
   ```

2. Identify the remote worker IP address and use it in the following command:

   ```
   erl -name worker@<workerIPaddress> -setcookie cookiename
   ```

3. To check the connection, ping the server node by running the below command:

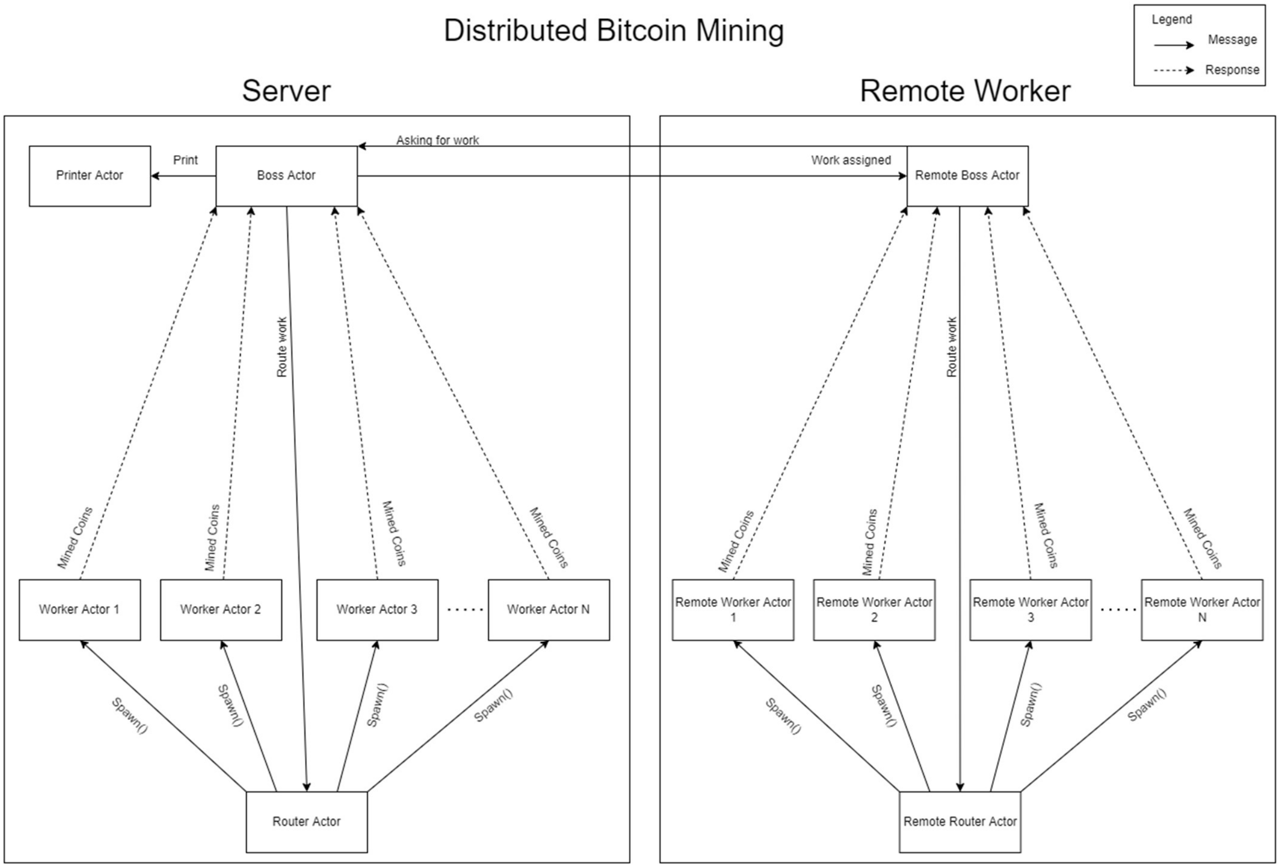
   ```
   net_adm:ping(‘server@<serverIPaddress>').
   ```

Program execution:

1. Run the following command on the server console:
   - To compile
     - *c(bitcoinserver).*
   - To start the server
     - *bitcoinserver:start(N)*
     - where N is the number of leading 0s

2. Run the following command on the remote worker console:
   - To compile
     - *c(bitcoinworker).*
   - To start worker participation in mining
     - *bitcoinworker:start(‘server@<serverIPaddress>').*

Architecture and working of the distributed execution



1. The above diagram depicts the architecture of distributed bitcoin mining project. This is implemented using Erlang and concepts of functional programming and actor-model to achieve the distribution of work.
2. The architecture consists of a single server and has the ability to get connected with multiple remote workers to distribute the workload. The server can work independently on its own for mining the bitcoins and has the ability to accommodate remote workers as and when they are available.
3. When the server is started, it intakes the leading number of zeroes, that a bitcoin should start with, and creates a boss actor who is responsible for coordinating the actors within the server. The boss actor creates two actors further.
    a. Work Router actor – It is responsible for distributing the work among multiple work actors within the server, such that the given problem is divided into multiple subproblems and each work actor is working on that specific sub-problem independent of other work actors within the server. The router actor which is spawned by the boss actor, determines the number of worker actors and the unit of computation to be used, to distribute the work amongst the worker actors.
    b. Printer actor – It is responsible for printing the information about the bitcoins mined.
4. The work unit is dependent on the configuration of the machine (number of cores) and the number of leading zeroes to be looked up for, in the bitcoins(hashes) generated.
5. Precisely, the number of actors created on each node is computed as

    *Number of Actors = Number of cores in a machine * 500.*

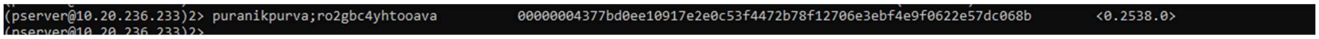
6. Whereas, each sub-problem (finding the leading zeroes from the hashes that are being generated) which is solved by the worker actor, is termed as the unit of computation, decided as below -

    *Unit of computation = Number of leading Zeroes to be looked up for * 100000.*

7. The moment any actor is able to mine the bitcoin (ie. Find a hash that has a leading number of zeroes equal to that is requested), it sends the message back to the boss actor and continues mining further if the unit of work is not exhausted.
8. The boss actor is then responsible for redirecting the work back to the printer actor, who takes care of printing the findings on the server.
9. When a remote worker is available, it requests the server to assign it to work. The server then passes on the information of a leading number of zeroes, that a bitcoin should start with, to this remote worker.
10. The remote worker creates a boss actor of its own, who further takes care of spawning a work router actor within the remote worker. This work router actor spawns individual worker actors, who then mine the bitcoins independently.
11. Whenever the remote worker work actor is able to mine the bitcoin, it sends the message to its own worker boss actor and continues mining further if the unit of work is not exhausted. The remote worker boss actor then sends this update to the server boss actor, and the server boss actor takes care of forwarding the same to its printer actor.

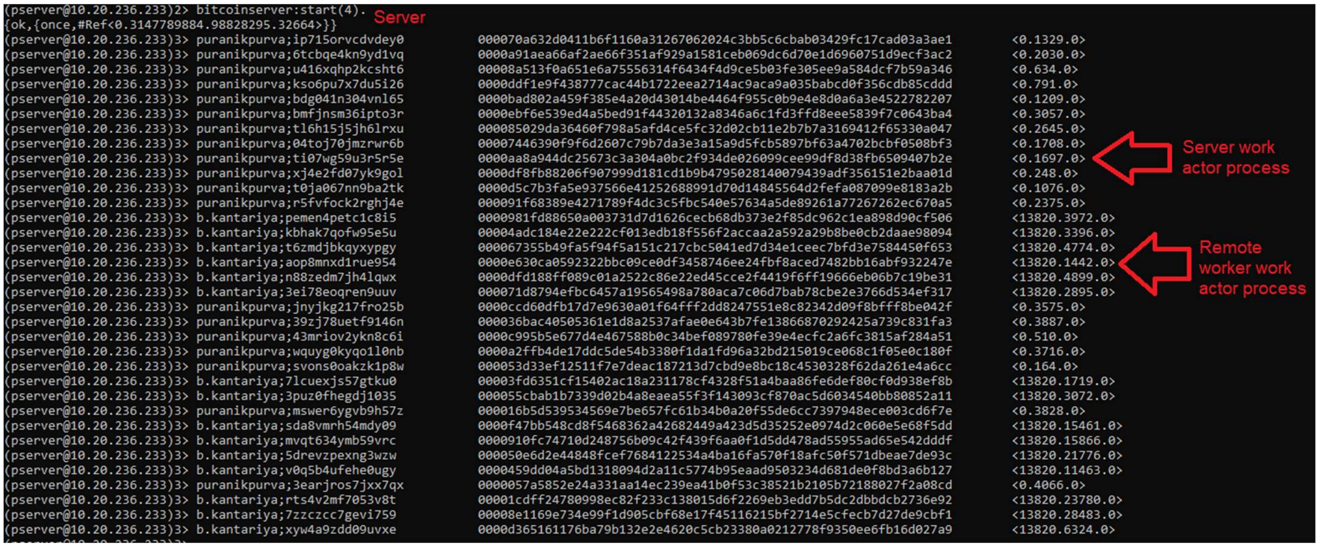
Largest coin mined -

With one server and one remote worker, we were able to find one coin with 7 leading zeroes. A snapshot of the same is below -



```
(pserver@10.20.236.233)2> puranikpurva;ro2gbc4yhtooava          00000004377bd0ee10917e2e0c53f4472b78f12706e3ebf4e9f0622e57dc068b        <0.2538.0>
(pserver@10.20.236.233)2>
```

The result of running your program for input 4 -



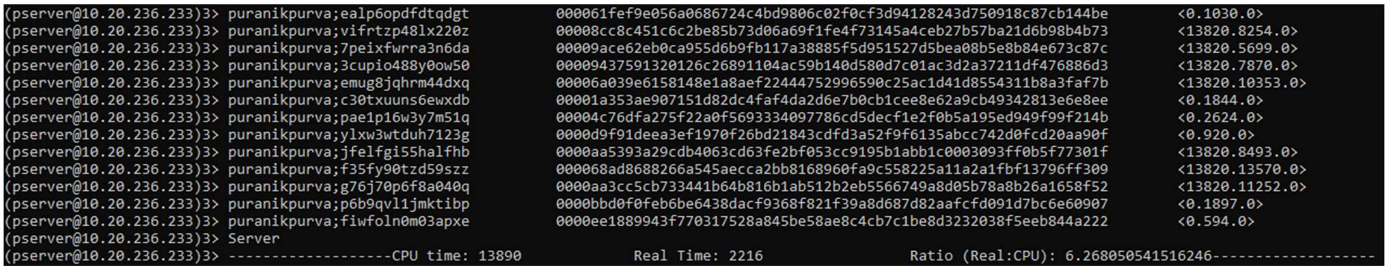
```
(pserver@10.20.236.233)2> bitcoinserver:start(4).
{ok,{once,#Ref<0.3147789884.98828295.32664>}}
(pserver@10.20.236.233)3> puranikpurva;ip715orvcdvdey0        000070a632d0411b6f1160a31267062024c3bb5c6cbab03429fc17cad03a3ae1        <0.1329.0>
(pserver@10.20.236.233)3> puranikpurva;6tcbqe4kn9yd1vq        0000a91aea66af2ae66f351af929a1581ceb069dc6d70e1d6960751d9ecf3ac2        <0.2030.0>
(pserver@10.20.236.233)3> puranikpurva;u416xqhp2kcsht6        00008a513f0a651e6a75556314f6434f4d9ce5b03fe305ee9a584dcf7b59a346        <0.634.0>
(pserver@10.20.236.233)3> puranikpurva;kso6pu7x7du5i26        0000ddf1e9f438777cac44b1722eea2714ac9aca9a035babcd0f356cdb85cddd        <0.791.0>
(pserver@10.20.236.233)3> puranikpurva;bdg041n304vnl65        0000bad802a459f385e4a20d43014be4464f955c0b9e4e8d0a6a3e4522782207        <0.1209.0>
(pserver@10.20.236.233)3> puranikpurva;bmfjnsm36ipto3r        0000ebf6e539ed4a5bed91f44320132a8346a6c1fd3ffd8eee5839f7c0643ba4        <0.3057.0>
(pserver@10.20.236.233)3> puranikpurva;tl6h15j5jh6lrxu        000085029da36460f798a5afd4ce5fc32d02cb11e2b7b7a3169412f65330a047        <0.2645.0>
(pserver@10.20.236.233)3> puranikpurva;04toj70jmzrwr6b        00007446390f9f6d2607c79b7da3e3a15a9d5fcb5897bf63a4702bcbf0508bf3        <0.1708.0>
(pserver@10.20.236.233)3> puranikpurva;ti07wg59u3r5r5e        0000aa8a944dc25673c3a304a0bc2f934de026099cee99df8d38fb6509407b2e        <0.1697.0>
(pserver@10.20.236.233)3> puranikpurva;xj4e2fd07yk9gol        0000df8fb88206f907999d181cd1b9b4795028140079439adf356151e2baa01d        <0.248.0>
(pserver@10.20.236.233)3> puranikpurva;t0ja067nn9ba2tk        0000d5c7b3fa5e937566e41252688991d70d14845564d2fefa087099e8183a2b        <0.1076.0>
(pserver@10.20.236.233)3> puranikpurva;r5fvfock2rghj4e        000091f68389e4271789f4dc3c5fbc540e57634a5de89261a77267262ec670a5        <0.2375.0>
(pserver@10.20.236.233)3> b.kantariya;pemen4petc1c8i5         0000981fd88650a003731d7d1626cecb68db373e2f85dc962c1ea898d90cf506        <13820.3972.0>
(pserver@10.20.236.233)3> b.kantariya;kbhak7qofw95e5u         00004adc184e22e222cf013edb18f556f2accaa2a592a29b8be0cb2daae98094        <13820.3396.0>
(pserver@10.20.236.233)3> b.kantariya;t6zmdjbkqyxypgy         000067355b49fa5f94f5a151c217cbc5041ed7d34e1ceec7bfd3e7584450f653        <13820.4774.0>
(pserver@10.20.236.233)3> b.kantariya;aop8mnxd1rue954         0000e630ca0592322bbc09ce0df3458746ee24fbf8aced7482bb16abf932247e        <13820.1442.0>
(pserver@10.20.236.233)3> b.kantariya;n88zedm7jh4lqwx         0000dfd188ff089c01a2522c86e22ed45cce2f4419f6ff19666eb06b7c19be31        <13820.4899.0>
(pserver@10.20.236.233)3> b.kantariya;3ei78eoqren9uuv         000071d8794efbc6457a19565498a780aca7c06d7bab78cbe2e3766d534ef317        <13820.2895.0>
(pserver@10.20.236.233)3> puranikpurva;jnyjkg217fro25b        0000ccd60dfb17d7e9630a01f64fff2dd8247551e8c82342d09f8bfff8be042f        <0.3575.0>
(pserver@10.20.236.233)3> puranikpurva;39zj78uetf9146n        000036bac40505361e1d8a2537afae0e643b7fe13866870292425a739c831fa3        <0.3887.0>
(pserver@10.20.236.233)3> puranikpurva;43mriov2ykn8c6i        0000c995b5e677d4e467588b0c34bef089780fe39e4ecfc2a6fc3815af284a51        <0.510.0>
(pserver@10.20.236.233)3> puranikpurva;wquyg0kyqo1l0nb        0000a2ffb4de17ddc5de54b3380f1da1fd96a32bd215019ce068c1f05e0c180f        <0.3716.0>
(pserver@10.20.236.233)3> puranikpurva;svons0oakzk1p8w        000053d33ef12511f7e7deac187213d7cbd9e8bc18c4530328f62da261e4a6cc        <0.164.0>
(pserver@10.20.236.233)3> b.kantariya;7lcuexjs57gtku0         00003fd6351cf15402ac18a231178cf4328f51a4baa86fe6def80cf0d938ef8b        <13820.1719.0>
(pserver@10.20.236.233)3> b.kantariya;3puz0fhegdj1035         000055cbab1b7339d02b4a8eaea55f3f143093cf870ac5d6034540bb80852a11        <13820.3072.0>
(pserver@10.20.236.233)3> puranikpurva;mswer6ygvb9h57z        000016b5d539534569e7be657fc61b34b0a20f55de6cc7397948ece003cd6f7e        <0.3828.0>
(pserver@10.20.236.233)3> b.kantariya;sda8vmrh54mdy09         0000f47bb548cd8f5468362a42682449a423d5d35252e0974d2c060e5e68f5dd        <13820.15461.0>
(pserver@10.20.236.233)3> b.kantariya;mvqt634ymb59vrc         0000910fc74710d248756b09c42f439f6aa0f1d5dd478ad55955ad65e542dddf        <13820.15866.0>
(pserver@10.20.236.233)3> b.kantariya;5drevzpexng3wzw         000050e6d2e44848fcef7684122534a4ba16fa570f18afc50f571dbeae7de93c        <13820.21776.0>
(pserver@10.20.236.233)3> b.kantariya;v0q5b4ufehe0ugy         0000459dd04a5bd1318094d2a11c5774b95eaad9503234d681de0f8bd3a6b127        <13820.11463.0>
(pserver@10.20.236.233)3> puranikpurva;3earjros7jxx7qx        0000057a5852e24a331aa14ec239ea41b0f53c38521b2105b72188027f2a08cd        <0.4066.0>
(pserver@10.20.236.233)3> b.kantariya;rts4v2mf7053v8t         00001cdff24780998ec82f233c138015d6f2269eb3edd7b5dc2dbbdcb2736e92        <13820.23780.0>
(pserver@10.20.236.233)3> b.kantariya;7zzczcc7gevi759         00008e1169e734e99f1d905cbf68e17f45116215bf2714e5cfecb7d27de9cbf1        <13820.28483.0>
(pserver@10.20.236.233)3> b.kantariya;xyw4a9zdd09uvxe         0000d365161176ba79b132e2e4620c5cb23380a0212778f9350ee6fb16d027a9        <13820.6324.0>
```

```
(pserver@10.20.236.233)4> puranikpurva;7bisjfs336w90xg        00005c9aa7d4f47ab62fdaf97d6fe75a2613efad3af601150a5a60964f7763f4        <0.964.0>

(pserver@10.20.236.233)4> puranikpurva;bak4w280sn9mmae        00002110d1529b489d1bcdf55139ccccecb334a3c1700a663dd0dc26b51b97ae        <0.2496.0>

(pserver@10.20.236.233)4> puranikpurva;4f5ruea89p1dro8        0000bf49567fdb38e6bcc9273151b529b9850de1d790f41c9048b68b0243f67e        <0.1283.0>

(pserver@10.20.236.233)4> b.kantariya;vey1b6jgestmglq        0000ae2ff5cba68e3145b6660698a4e581157536a72ac08ca9683f16f16f1fda        <13820.7666.0>

(pserver@10.20.236.233)4> b.kantariya;icw2t9gimiurl49        000099c402d506c31c8691ca61083e06412b93df4a39cd1aa599e81de2153829        <13820.10165.0>

(pserver@10.20.236.233)4> b.kantariya;jn72ams16andu0m        00003245d478026e242dbcca73a267349c5a2828b6d47d2d040b653313c31b91        <13820.5201.0>

(pserver@10.20.236.233)4> b.kantariya;2mxiautjim19lr7        000097a82927fb9da1ff0741f1b917d5c3f6a8c0a4a16bb75091e551eab75f9e        <13820.3981.0>

(pserver@10.20.236.233)4> puranikpurva;pwkq86z4d132m0v        000029016a713e24b9e9c7088822c405fa8093527dda02b67bd154dbd30b1b5a        <0.959.0>

(pserver@10.20.236.233)4> puranikpurva;t4ujcuqgtms4ctf        00001fe234bc32f07f0adc782f0720b0a44f3937da535e9f0c0c4387af136c8a        <0.1544.0>

(pserver@10.20.236.233)4> puranikpurva;ph0jy32n211ct7y        0000c3fc7f386a10f57df3e82ddd6c53c1d7144759bff1d53981625271e5fe51        <0.1804.0>

(pserver@10.20.236.233)4> puranikpurva;1mrz0symrl0ntlt        00009dc1cd1c20436bb5c81a7cc63cd426a3257b539bec2cf895befea78c9960        <0.622.0>
```

Statistics obtained for computation and configuration of the machines –

Configuration of the machines used –

1. **Server** –
    a. Processor - Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz   2.11 GHz
    b. Number of Cores: 4
    c. Actors: 2000
    d. Work size per actor (Given Number of Zeroes = 4) : 400000



```
(pserver@10.20.236.233)3> puranikpurva;ealp6opdfdtqdgt         000061fef9e056a0686724c4bd9806c02f0cf3d94128243d750918c87cb144be        <0.1030.0>
(pserver@10.20.236.233)3> puranikpurva;vifrtzp48lx220z         00008cc8c451c6c2be85b73d06a69f1fe4f73145a4ceb27b57ba21d6b98b4b73        <13820.8254.0>
(pserver@10.20.236.233)3> puranikpurva;7peixfwrra3n6da         00009ace62eb0ca955d6b9fb117a38885f5d951527d5bea08b5e8b84e673c87c        <13820.5699.0>
(pserver@10.20.236.233)3> puranikpurva;3cupio488y0ow50         0000943759132012 6c26891104ac59b140d580d7c01ac3d2a37211df476886d3        <13820.7870.0>
(pserver@10.20.236.233)3> puranikpurva;emug8jqhrm44dxq         00006a039e6158148e1a8aef22444752996590c25ac1d41d8554311b8a3faf7b        <13820.10353.0>
(pserver@10.20.236.233)3> puranikpurva;c30txuuns6ewxdb         00001a353ae907151d82dc4faf4da2d6e7b0cb1cee8e62a9cb49342813e6e8ee        <0.1844.0>
(pserver@10.20.236.233)3> puranikpurva;pae1p16w3y7m51q         00004c76dfa275f22a0f5693334097786cd5decf1e2f0b5a195ed949f99f214b        <0.2624.0>
(pserver@10.20.236.233)3> puranikpurva;ylxw3wtduh7123g         0000d9f91deea3ef1970f26bd21843cdfd3a52f9f6135abcc742d0fcd20aa90f        <0.920.0>
(pserver@10.20.236.233)3> puranikpurva;jfelfgi55halfhb         0000aa5393a29cdb4063cd63fe2bf053cc9195b1abb1c0003093ff0b5f77301f        <13820.8493.0>
(pserver@10.20.236.233)3> puranikpurva;f35fy90tzd59szz         000068ad8688266a545aecca2bb8168960fa9c558225a11a2a1fbf13796ff309        <13820.13570.0>
(pserver@10.20.236.233)3> puranikpurva;g76j70p6f8a040q         0000aa3cc5cb733441b64b816b1ab512b2eb5566749a8d05b78a8b26a1658f52        <13820.11252.0>
(pserver@10.20.236.233)3> puranikpurva;p6b9qvl1jmktibp         0000bbd0f0feb6be6438dacf9368f821f39a8d687d82aafcfd091d7bc6e60907        <0.1897.0>
(pserver@10.20.236.233)3> puranikpurva;fiwfoln0m03apxe         0000ee1889943f770317528a845be58ae8c4cb7c1be8d3232038f5eeb844a222        <0.594.0>
(pserver@10.20.236.233)3> Server
(pserver@10.20.236.233)3> -------------------CPU time: 13890          Real Time: 2216             Ratio (Real:CPU): 6.268050541516246-------------------
```

2. **Worker** –
    a. Processor – Apple M1
    b. Number of Cores: 8
    c. Actors: 4000
    d. Work size per actor (Given Number of Zeroes = 4): 400000

3. CPU time – 13890, Real time=2216, Ratio (Real: CPU) : 6.2680. This ratio tells us how many cores were effectively used in the computation.
4. According to our observations, the best work unit value that we found is the multiple of the number of cores available on the machine. For example, we tried setting the number of actors to be created in multiples of 10, 50,100,500,600, time that’s of the number of cores. And we found that the best work unit was when the number of workers was 500*Number of cores available on the machine.

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

We were able to run our code with two machines, one being a server and another being worker.