Team: **Podduturi Prathihas** & **Venkat Sai Putagumpalla.**

**Description**

Bitcoin mining using Distributed systems based on Actor model. It is done by following Server-Worker model. In which Server assigns works to workers to mine bitcoins and Server also mines bitcoins.

**Server**

- On running server, it creates a node on the current system, and it starts mining and waits for workers requests.

- The current system connects with other system. The current system acts as the server and the other system acts as worker.

**Worker**

- The other system which has connected to the Server system is the worker which runs multiple processes in the system and mines bitcoins. The multiple processes spawned at a time depends on the number of cores of the system.

- Each can spawn many workers.

- We were able to spawn **8 workers.** And even if try to increase number of worker count the parallelism is not increased

Diagram

Description automatically generated

In the working process a random string is generated and the random string is attached to our name. This string is then hashed into set of characters using SHA256 algorithm. At the end we checked the leading number of zeros in the hashed string.

For the input 4 below is the output in the form strings and their respective hash strings.

v.putagumpallaN4FkwJlOpSmcdA 00002a9a31cfe0b32dfccbb6e66fe335a6ad0a10195eea2717e4bc49e086b41d

v.putagumpallaPKOC8n8Pivw1Wg 000040d55d2c1aa00ccc2475bf578bb911f230cdce526c360b4fe43e6dd69665

v.putagumpallax8h8wl+8Bz/YLg 0000a912d6d1bef5a89759fe93fb50032bf7b528c7a2c28ac9eb3d8b850c642b

v.putagumpallaYoqkLpuYgu4UmQ 0000e0d7ef14eb9f11f32c4bdf579c5b7b6bcba6ca2cf9e309a866fb14d01c44

v.putagumpallaJHQ0jOvMBBhgQA 0000998b825511be40cee0196496a1f08af5a87d26c9944efcd73a58bf0fd6b8

v.putagumpallaNFBvSri8SejEaA 00003a64f3018a53966b5c39a4e5bfafd6e3466b1acfdfb6dfa85affa73c4ba4

v.putagumpalla92YRCvyWPDOKig 000009ec72a527df1206ab91a013557f433bbd85773158aeba2fc8f4395c443a

v.putagumpalla6LZCISzA1WguAA 0000233080812b0a795af73c9efe4eb56b2feefd08176bb9f9339bd098580af1

v.putagumpallauUCirED9kmcGNA 0000c9c175546e1ee5bffa97519d77aa86898a4690c9cddc916e664d83267af2

v.putagumpallarqTNjiA+jQjNyA 00005129f4ce7637db09dec22a9b8cc23e770d323a3f251cf5e717dcb9e287d1

v.putagumpallaj9WjLBwCpzvyHg 0000f3761e8458568362174228471b82484ce37e550a12c3717ba2ca6cbdcadc

v.putagumpallarSW1r5ql89gPsA 00009c94aab34b1eef36c1e53cc4351a0f7b4e4206aeae7ede54f6fb96ce61aa

v.putagumpallaajVSMCW3yu/tHw 0000e9814d6c72fce53c45889195b76816b4c31de86eed6bb164f0c3b5f131a4

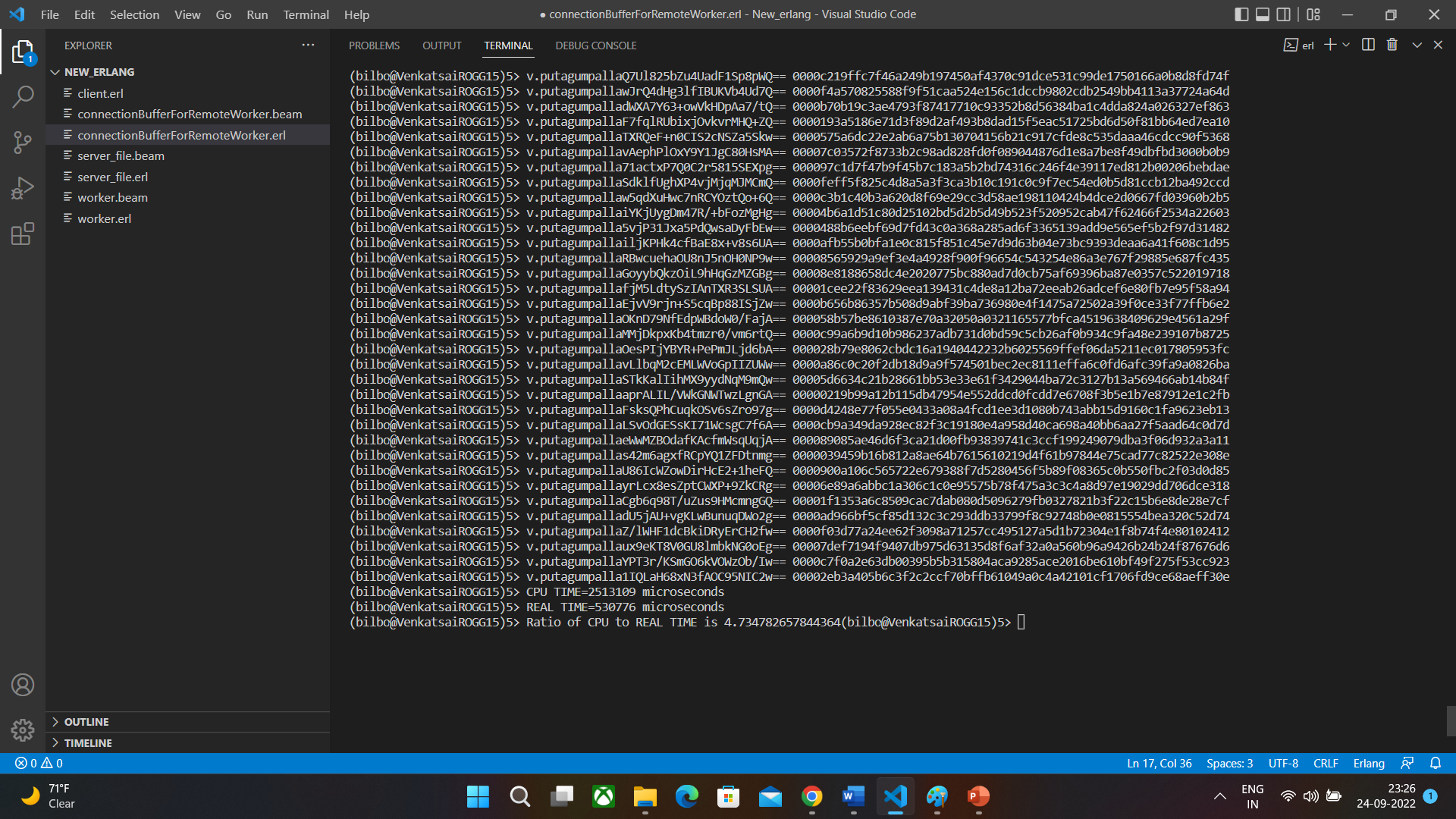
v.putagumpallaJgVqK4fwnjcTNQ 0000ead6ac25667437fb44224d124bb60d92d069ab5351dba2ddf0e1ad411de5

v.putagumpallaSnprUc4dlCtPpQ 0000d4df7c3e171d73929ce9039d1cd7b1b0b7485a6f5b90d83b4aa15a1c18f8

**Size of the work unit**

The number of sub-problems each worker gets in a single request from the boss is 20. The worker needs to mine 20 Bitcoins. As the input size increases, we need to mine Bitcoins with hash string higher than usual. So, it is difficult to mine bitcoins with a greater number of leading zeros. The worker would be working for a very long time and there might be workers waiting for the cores. So, we thought the ideal number would be 20 for each worker.

The ratio of CPU and REAL TIME is **4.734782657844364**. For achieving this ratio, we have spawned 7 workers



We were able to run it on **three working machines**.

We managed to find the coin with **8** **leading zeros**.

**Input String:**

**v.putagumpallarLuJD+qk9BGKFMv3WuZsew==**

**Output Hash: 00000000b66762b35e005dfcf250f4237ad8053db1ce3e275b93f2e0fc1bcdb2**

**Steps to start the Program:**

1. **Compile Server: c(server\_file).**
2. **Compile Worker: c(worker).**
3. **Compile connectionBufferForRemoteWorker: c(connectionBufferForRemoteWorker).**
4. **Start Server: server\_file:startServer().**
5. **Give required number of leading zeros when server asks example: 4.**
6. **Start the worker: worker:startWorkerProcess(4). Here 4 in the arguments says that we want spawn 4 workers**
7. **Enter whether server is present in remote node or not by choosing 1 or 0**
8. **If you select that server is running on remote node give host address of node on which server is running**