COP 5615 – PROJECT 1 (README FILE)

Group Members: Prerna Mandal
Jyotsana Walia

Bit coins are the most popular crypto-currency in common use.

In this project we aim to use actor model in akka to implement Pooled Mining where multiple generating clients contribute to the generation of a block, and then split the block reward according the contributed processing power.

Local Actor: (BitcoinLocalMiner)

This serves as a Master, which takes following arguments

- Number of zeroes
- launchRemoteToo – If remote workers are desired in the application

and uses input string prefixed by the gatorlink ID of one of the team members(prernamandal) to calculate the SHA256 hash

This has the ability to mine coins on its own by creating local workers. As remote workers become available it utilizes the remote workers to mine fast. We have used future to perform many operations in parallel– in an efficient and non-blocking way.

Size of the work unit
The search space for mining bit coins is a long string of random alphanumeric characters. This string is divided into smaller chunks of 5 characters prefixed with the gatorid – 'prernamandal'. The work unit is decided based on the number of zeroes needed and the number of local or remote processors available to the master.
The number of workers is determined by the number of processors in the machine (3/2 times number of processors) and they are assigned work in round robin fashion. The data is divided into smaller chunks rather than splitting it equally amongst all available actors (for proper load balancing)
(Running time: A Loop goes from 1 to 100000 – this counter can be increased/decreased to control the duration for which the project runs.)
We calculated the CPU to REAL time ratio by changing number of workers, time allotted to each worker, Maximum number of tries

Number of zeroes	Time for each worker	No. of Worker	CPU to real time ratio
3	2700 ms	5 times processor's count	3
4	2700 ms	5 times processor's count	3
3	1000	2 times processor's count	2.3
3	10000	1 times processor's count	< 1

Conclusion: Optimal parallelism of more than 3 times when ,time of each worker ~ 3 seconds and number of worker = 5 times the processor count.

2. The result of running program for k-

the number of leading zeros required as 4 is : Bitcoins mined from Remote workers

prernamandalworker11LN5wnn53418 :
0000941285f54f1b1b3bd7e232cabe610de7b7d64a85c81ab83f74557a867a52
prernamandalworker11LN5wnn54503 :
0000e349b70d4c937cb05f9553fd45591da003fb93771e303e6370ace6a1f4db

3. We observed that multiple actors run at the same time and so multiple cores were utilized.

Running time for the above

real 0m14.139s
user 0m37.863s s
ys 0m1.415s

Ratio ~ 2.8

4. Coin with maximum number of zeros (5 leading zeros):

prernamandallocalw1271VWby19614
0000076c59e0480631f9ab305230e037731b743e7f82b47e82226235ea9447ec

5. Largest number of working machines: We were able to run on 2 machines.

Observation for size of the work unit:

.

- 5 times available processors = the number of workers

Steps to execute:

This project submission includes a folder BitcoinMiningProject which has subfolders in it. In order to run the program (using SBT), open 2 command prompt terminals.

In terminal 1, reach: BitcoinMiningProject\BitcoinMasterLocal Type time sbt "run 4 launchRemoteToo" and hit enter. In terminal 2, reach:BitcoinMiningProject\BitcoinMinersRemote Type sbt run and hit enter. All bitcoins mined from both - - Server and Client will be displayed in 'LocalMiner.scala', the server. (Server has the capability to mine by itself even if client is not available). - Update IP address of application.config present in BitcoinMinersRemote folder to client machine's IP and update IP address in LocalMiner.scala file to client's IP address. - Update IP address of application.config present in BitcoinMasterLocal folder to server machine's IP and update IP address of RemoteMiner.scala to server's IP address.