Group member

Tao Zhang UF-ID:7636-6624

Haowen Chen UF-ID:8141-1485

1. Size of Work Unit:

64 work units that my determined results in the best performance for my implementation. The total number of tasks we set is 512. When the worker is equal to 8, the worker will get the number of sub-questions from the boss as 64. Each worker gets equal of work at every instant. The size of our work unit is determined by regulating the actors and determining the ideal distribution of sub-problems for a fixed number of workers.

2. Result for k=4 (4 0's in the hash notation) 30rows

```
Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0
chenhaowen;dZ%5o 16 00005d0de9d1a9ed6a81bced57077b2b01d224427a280da46573c8e311dff6a9
chenhaowen;V(4D| 16 000005fd793e6ab1ccc7f5971a17853c6db5c4d425b5124b01153e1d1b481767
chenhaowen;-h(E( 16 0000169be87c9e39d6005a066f19a16c1182a41a2a80d8b59da035e335814262
chenhaowen;>e>#U 16 0000bf6b1857d8949aefae978daa76525e31f005db0f2b97d3898d2409de73be
chenhaowen;ET}X# 16 0000246ab28b36af6bf244f9fbf376298cb00b8271f0c132d65dae9640eeabe5
chenhaowen;ewAM4 16 00001317ad4a369d849c79e3c30c038e0148f6c83666a3580a9a695b8de04d0c
chenhaowen;'XMz` 16 0000b320e4239930e74c9f53c17f911ff87d209ba4d471e73366d1669c321e29
chenhaowen;QjA!g 16 0000a9e6d96bc5afb6d16ddec8d6eafce3e1758728068b72e99852824dc5e784
chenhaowen;fwn3A 16 0000d968527b74a55f9c29f2029b3f8da2a4d5d4ce5157897f83573e1c0067a8
chenhaowen;PyGv/ 16 0000bc799fc5d63f5951c47d3c17bb8f08e45e7b5ac33ceea3e9d30975944ce3
chenhaowen;DI'-= 16 000003b86992bc34b9e3ea4b6ec74900a998325501bf4fd79e918c4055dc390c
chenhaowen;E_AEj 16 000023655e3545ced5e384578bcc8dbcea079de7d5cbcf962513e11e529a2fd6
chenhaowen;AO"W) 16 0000347fe5f67009dfa5345155f05ab2ad3973e3368b8a2b3e151190c94c8f8b
chenhaowen;1*-OB 16 000028586ed698f4e1789282a50973af2f7d53b39768de6959778a6c8c8e6af1
chenhaowen;9-jx, 16 0000304af203348a1eedcc85e553c076aeb834990eb14af9ae5fa0192eb28d5b
chenhaowen;PGoA< 16 0000e2e26c18d1e8e242d053d3152f755b5496c166144a96233cb31b0e0cacc8
chenhaowen;iaq>3 16 0000d3fa84498fa5fc6bc94983eaf42af54f06e7f88c9166a7b02ed2652bcc55
chenhaowen;KFop+ 16 000074a1311c29adf54d51bb9d5a037cbb85ff9f8ce091f54fb98b8534b38fd1
chenhaowen;4\5K1 16 0000415f79458ac285843a0a2c895a070364a76da0e1c5666ec7c552d9998631
chenhaowen;Oj*mf 16 0000a52cfda258f6ae62ef9ed62c8a3bb9e4a343eace0c5efcecefb1c873264f
chenhaowen;%1]Jd 16 00003dcc7c303e368a4539eafed7ff634db432aa955c9626c07a719d479e8c85
chenhaowen;=P15Q 16 000023b78994b731fbbd7a40c29ca206acaefa1783f0a94b1dbbe7d36edb5765
chenhaowen;F#P~{ 16 000002c2fc6c53564c4b8fa0925aba488825a2361e06b182459a7ec91d39b531
chenhaowen;qx1)| 16 0000cdfdb2ac3aa91e46ef361b4329d33d85d9ea12322c6ee9bd000f383ae04c
chenhaowen;zQSPG 16 000083d87fb7d6133b9c71f588d1d6a25c0147b7475fc794df74ff15d2101f6e
chenhaowen;lO*t; 16 00004f0af8e05c832b201c58de53d05551e899898960d45fca5bc3c5ebcf66bf
chenhaowen;@7,`@ 16 00000d929e5358d6c38e6551ef42e8bef31fb65ff425267e96a1ce75239f54c5
chenhaowen;G69NU 16 00000b1a4f951d3853b5cbd645cb5526ff77ca5a7d7da1f17ca1e9ca48bf6e9f
chenhaowen;u.ruO 16 0000468a8247cbafaa4ed1c3ac8cb18d68af0ca775969416011a890f9b6d423a
chenhaowen;CZ?KK 16 000037325e4481e2829e4649f7122b17a143262f9d11b9157feb3a01f4b0dfaf
```

3. Running time for worker = 1, 2, 4, 8, 16, 32, 64, 128, 256.

512 rows in total (4 core machine)

Worker = 1, ratio = CPU/Real ≈ 1.00

```
Real: 00:08:05.815, CPU: 00:08:13.843, GC gen0: 65070, gen1: 54, gen2: 4
```

Worker = 2, ratio = CPU/Real ≈ 1.82

```
Real: 00:05:04.660, CPU: 00:09:24.531, GC gen0: 69014, gen1: 36, gen2: 3
```

Worker = 4, ratio = CPU/Real ≈ 3.22

```
Real: 00:02:59.224, CPU: 00:09:47.328, GC gen0: 69404, gen1: 21, gen2: 1
```

Worker = 8, ratio = CPU/Real ≈ 5.48

```
Real: 00:02:17.454, CPU: 00:12:31.359, GC gen0: 63269, gen1: 16, gen2: 1
```

Worker = 16, ratio = CPU/Real ≈ 4.50

```
Real: 00:02:59.250, CPU: 00:13:30.015, GC gen0: 65442, gen1: 20, gen2: 1
```

Worker = 32, ratio = CPU/Real ≈ 4.70

```
Real: 00:03:04.934, CPU: 00:14:26.062, GC gen0: 70156, gen1: 22, gen2: 1
```

Worker = 64, ratio = CPU/Real ≈ 3.71

```
Real: 00:02:50.337, CPU: 00:12:59.156, GC gen0: 62675, gen1: 22, gen2: 1
```

Worker = 128, ratio = CPU/Real ≈ 4.28

```
Real: 00:03:18.082, CPU: 00:14:09.312, GC gen0: 68450, gen1: 21, gen2: 1
```

Worker = 256, ratio = CPU/Real ≈ 4.69

```
Real: 00:03:06.290, CPU: 00:14:33.625, GC gen0: 69752, gen1: 20, gen2: 1
```

Worker = 512, ratio = CPU/Real ≈ 4.57

```
Real: 00:02:53.148, CPU: 00:13:11.343, GC gen0: 63812, gen1: 20, gen2: 1
```

In a word, worker=8, ratio is the maximum.

4. Coin with max k=7

chenhaowen;A>g}%

0000000ab8895931c3c9209d413d73725f5a6654c0a4ab0e75375b29548415d3

5. Largest Number of working machines used to run code: 2

6. Client.fsx and Server.fsx

In the same WiFi environment.

Server: Haowen Chen's machine IP:10.136.97.12

Client: Tao Zhang's machine IP:10.136.32.178

The first step: first open the server.fsx, so that the server can run autonomously to mine Bitcoin, the result is shown in Figure 1.

```
C:\Users\76485>dotnet fsi --langversion:preview server.fsx
Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0
[INFO][2021/9/23 3:06:16][Thread 0001][remoting (akka://RemoteFSharp)] Starting remoting
[INFO][2021/9/23 3:06:16][Thread 0001][remoting (akka://RemoteFSharp)] Remoting started; listening on addresses : [akka.tcp://RemoteFSharp@localhost:9002]
[INFO][2021/9/23 3:06:16][Thread 0001][remoting (akka://RemoteFSharp)] Remoting now listens on addresses: [akka.tcp://RemoteFSharp@localhost:9002]
server recive a message from server: 886
chenhaowen;sv>'j 16 0000a4fbe32c627cb0e628fe8e3809fd08fc42117a1b9587a94804e7cdd2d2fe
chenhaowen;jJahy 16 0000917aa633c2883c916e947963423ca8d1e184fc07819b29e12843d1ee3096
chenhaowen;NwgUY 16 0000d31a0e353c66d5cfb65d82f1ec0a97d9b5a89572c63f8faf5a3c4de780da
chenhaowen;x< qV 16 0000c1d4c8a0262d25d6dc4aae3ed369f9de87c35a818d2d79191109f57c78bb
chenhaowen;O=<2m 16 0000c7c7de602c2b09ffc7f11fbdbac0e6640698b8ac11df682be56edd95d6af
chenhaowen;sM?xk 16 00008be9f14bf36ad2fff80c885a17aaa4e5179d5df2e89692a9a85599540915
chenhaowen;L J[S 16 000017b63700ffc48bde42d17360c8b202a2bffafb5364f7c4d0c9cb9d43296a
chenhaowen;4diZ, 16 000086bdd0bdae9bad871916f253a6944c1ee016bd8a2e9ac1ea64f8dbf0e349
chenhaowen;8=k:G 16 0000e051301186997339bad8c96c93e6421209442c6e0d363eae153105092bd3
chenhaowen;+vC'7 16 0000ee0c9a3a2082733202e979042b72043f0979b0db00acf7158b935e851d50
chenhaowen;Ua(OF 16 0000fad7155df36e2e868c3ea5749e79e4f05990c88e8a0b1531b65276fb980b
chenhaowen;IQy\n 16 00007d0b01b89394f756d223ab7e44e2ce04e0bdae3022d3692579c26de74e40
chenhaowen;O|;d* 16 00001c384667bc19e3e7007902731bcc7ee2ebdbe0d8451180f6191100e92972
chenhaowen; B7%3 16 0000ff767bfa89c58e179b189ef3031773ca239665bb3ad70ea600fde986ceb1
chenhaowen;[pWq" 16 0000814b1cb337694039a8e019bfa4871195cabbadbd4758c3fee5569af0f0eb
chenhaowen;m8U(E 16 0000f618d58b211a739f6a01e220fa1e82448fab45bbbc891f31821268f06f8c
chenhaowen;k=' <k 16 00003e2639a7d3650db79fea0293438cc16b817edf8e25cf57e3e2a1056a74b0
chenhaowen;2+ai[ 16 0000f02f78a2552159049565a968b3b26166d89f1d6083a7a78b841ba3dda702
chenhaowen;-/n[t 16 0000826e7211a6a30787ed24e67c8e7f82090031ef5019a3cc8905bda605e6f4
chenhaowen;_ #mj 16 0000689d4321f1f81603f8a9d784762cc7f412108d3dc0ea27bc489b94f5bd6a
```

Figure 1

Step 2: Then open the client.fsx, the service.fsx will connect to the service remotely, and then the client will receive the information from the service, and when the client receives the information, it will start mining bitcoin, the result is shown in Figure 2.

```
C:\Users\76485>dotnet fsi --langversion:preview client.fsx
Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0
[INFO][2021/9/23 3:07:35][Thread 0001][remoting (akka://RemoteFSharp)] Starting remoting
[INFO][2021/9/23 3:07:36][Thread 0001][remoting (akka://RemoteFSharp)] Remoting started; listening on addresses : [akka.tcp://RemoteFSharp@localhost:2552]
[INFO][2021/9/23 3:07:36][Thread 0001][remoting (akka://RemoteFSharp)] Remoting now listens on addresses: [akka.tcp://RemoteFSharp@localhost:2552]
client recive a message: start!
Worker 1 start working
chenhaowen;Lrq`g 16 000072aa31692d4a221c2774b72aae95e58b1b387dccb544be61dd244a938463
chenhaowen;g 1", 16 0000487e8b05f5cc8f2e5adb43930b81b5586017d33a4795d7babd5d939f6a1c
chenhaowen;11ET- 16 000049d964a7a6b4fdc26cf9a8ff01b3df056db9e298788ac118f6486433b38e
chenhaowen;@8Fdm 16 0000bf289587c048b12a8bd79d187554885726c2de0ae9b83e7b2c7a6b227d06
chenhaowen;xak|4 16 0000b17565d109d4e3d2704f6d2f3249dcf26251a1bbc564bb6cbac859803f7f
 Worker 1 finished its job
Worker 2 start working
chenhaowen;fPxYj 16 000010de1b07f72ded70555c374f52b104cf92ca08a089006f101dff5bedab3f
chenhaowen;``8oD 16 0000fcd583aabb9cc29a1d9b9e024f8c2a2ef04f079ff4621fc08527746392fd
chenhaowen;h/:<` 16 000088576df0ea84a387ebc8262103f5bf61b97ee6e25ab4ff03b06107020a80
chenhaowen;`A1M| 16 00005009880e3798f87f1dc85194e7e592c4c40c7e459ca0039b89eb63a1ee77
chenhaowen;([xaL 16 000083bc3391fc9a65f452d2a371bee563389683afefdce5b386411eace4f189
 Worker 2 finished its job
Worker 3 start working
chenhaowen;hGk}R 16 00005ccc92bb4387962066ad9f36883961b846947f15eddd8f1c29263e4c6ae6
chenhaowen;5y=:/ 16 0000568b392ccaa94f4fbc5382414ac8e3d6a5d52c312a255b8eca43bed5ba54
chenhaowen;}`sQQ 16 0000c9b768d93f875c5afebec57c4b8af4148d0f80f7c83c7287090475460d1c
chenhaowen;%So_z 16 0000692470a4ee21565f67ad6634b949d6931d7ca878c4e9717b6490fb5eadc3
chenhaowen;/*o/b 16 0000de248394d3f89abe234353f3fba410a61625d073bbac74de0a86394a791f
 Worker 3 finished its job
Worker 4 start working
chenhaowen;F1/4! 16 0000b1ec536efcaaaa78ec3492eafe0784246ea1e3aa8034e2fef8d81e6702c5
chenhaowen;4oE#x 16 0000f28b75567ac3daf03cf9a03cf1e74458291d58e12d64992f72ec9ef8a216
chenhaowen;`U)]] 16 0000a981fbb35e154592851ad1d1bab6710cd567aac1c042ebc22f114e41df49
chenhaowen;E;'`i 16 000038633eab29ff3ac29a8811bf1bc96aad1e290c7dd6c71b16354a7e635d63
chenhaowen;34T,! 16 0000a6412fe4955644a51d419b489caf60044fbc7b996d5c1cb8c5aabffe4f70
 Worker 4 finished its job
```

Figure 2

Step 3: The client results will be displayed in the service.

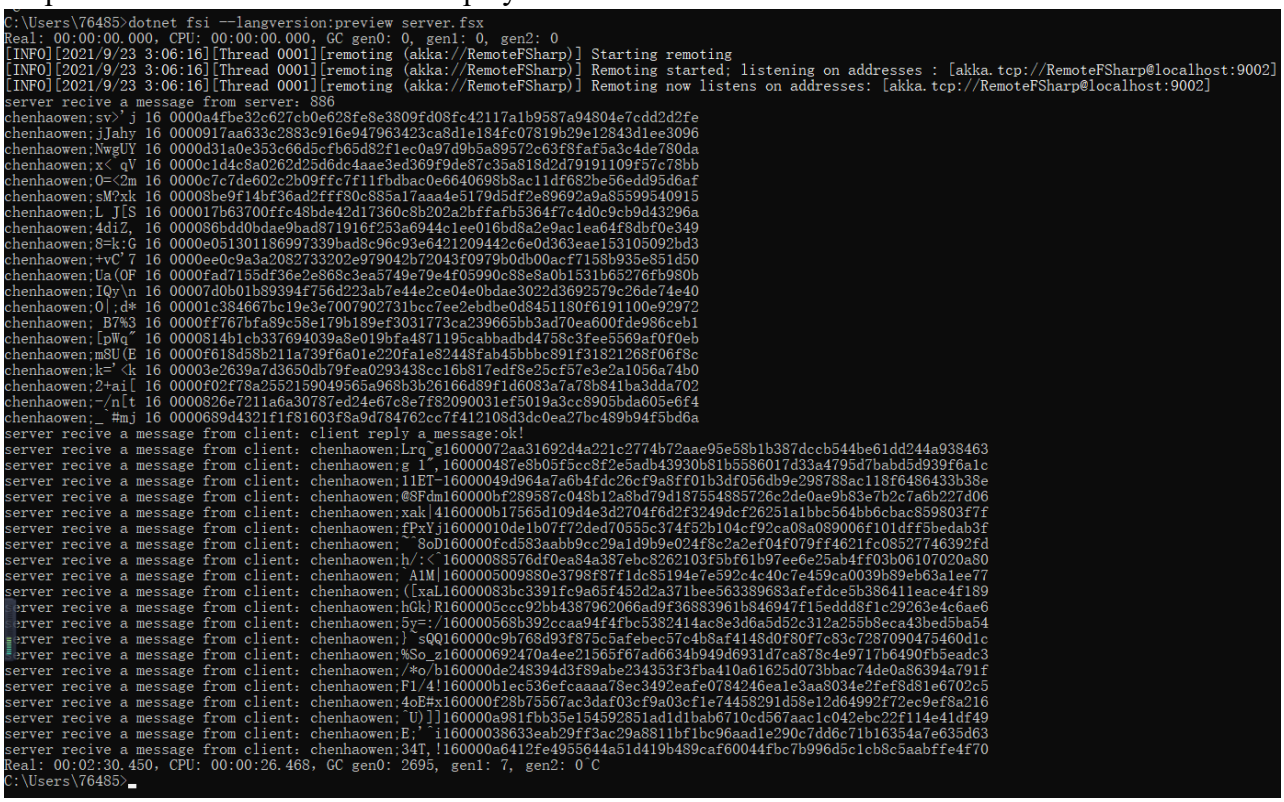

Figure 3

7. Note:

How to run the code: open the command prompt(cmd+enter)

(1) project1.fsx dotnet fsi --langversion:preview project1.fsx

(2) server.fsx dotnet fsi --langversion:preview server.fsx

(3) client.fsx dotnet fsi --langversion:preview client.fsx

My find256 function means that "chenhaowen" is gatorlink ID as “prefix”, 5 means random string length as n. 4 means 4 0's in the hash notation as k. “count” means Number of bitcoin output. The 16 in the output means "chenhaowen" + ";" + "5 random string length". Total length = 10+1+5=16.