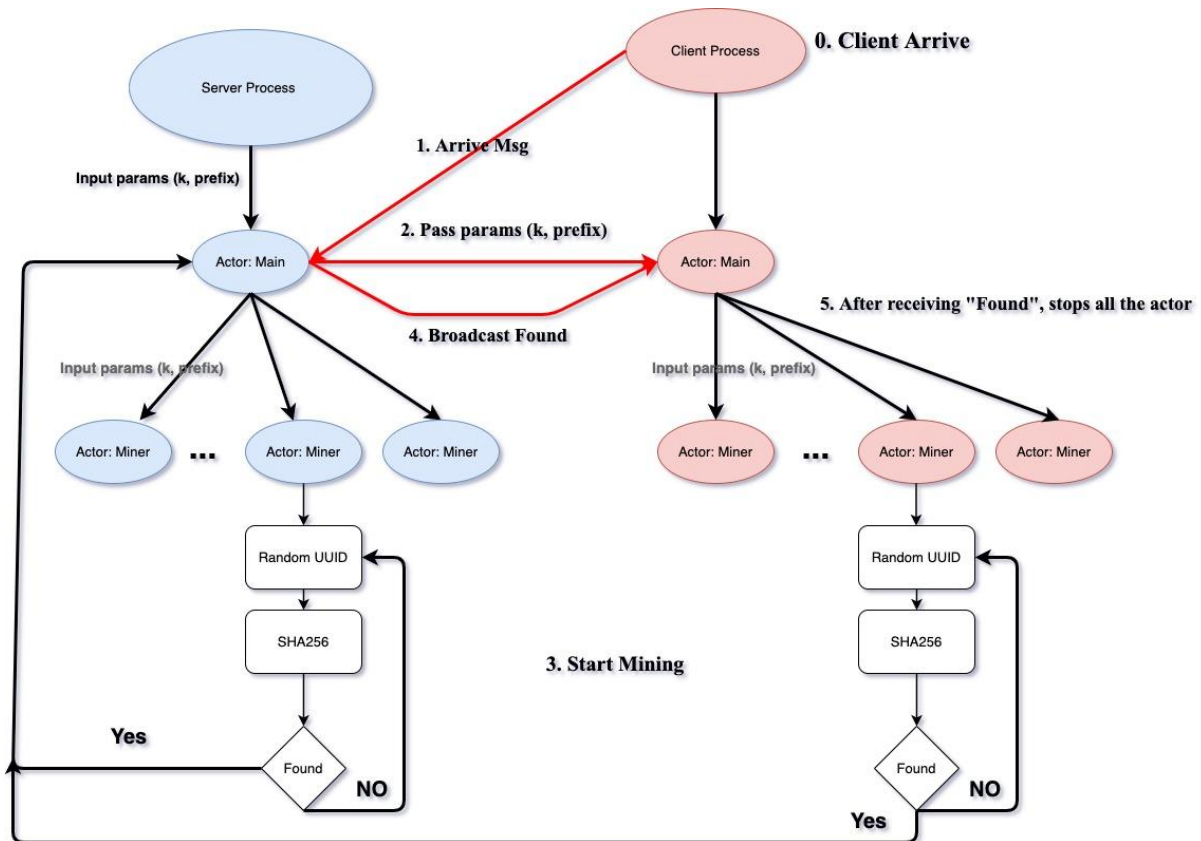


# COP5615 Project 1 Report

Yi-Ming Chang UFID: 83816537

Date: Sep 22th, 2021

## System Diagram



## Project Questions

1. Size of the work unit that you determined results in the best performance for your implementation and an explanation of how you determined it.

By observing htop and CPU/Real ratio. The decision is to assign a number that is less equal than the core on the CPU. Since other multiprocessing architectures also create their own subprocess according to the number of core on the CPU.

**After the experiment of observing the CPU% we create actor as below:**

**1. 4 actors on 8 cores macOS**

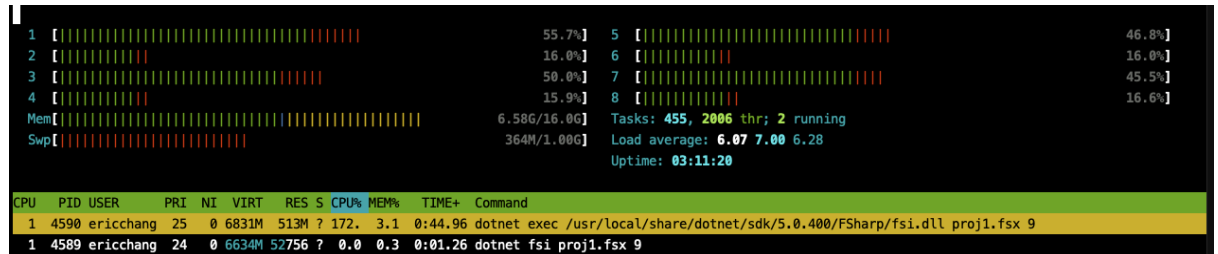
**2. 8 actors on 8 cores Windows OS**

- We test several large numbers of work units.  
For example, using 500 workers on one machine. However, the performance did not dramatically increase. After checking the CPU / Real ratio is showing that most of the workers are redundant (dead threads).
  - Guessing most of the thread is just context switching (worse performance)
- Use a number that is less equal than the number of the cores in the CPU  
→ trying 4 ~ 8
  - **8 actors, the outcode is not expected (Figure1)**
    - it seems that 50% of CPU numbers on **Mac** will have a better outcome
  - **4 actors, the outcode is efficient (Figure2)**
    - It seems to have limited usage on macOS, each PID seems can only deploy 50% of CPU numbers
    - **So, we decide to apply 4 actors on master and 4 actors on the local client**

### Figure 1. 8 actors for each process

The main process with 8 actors up only provide 172% CPU%

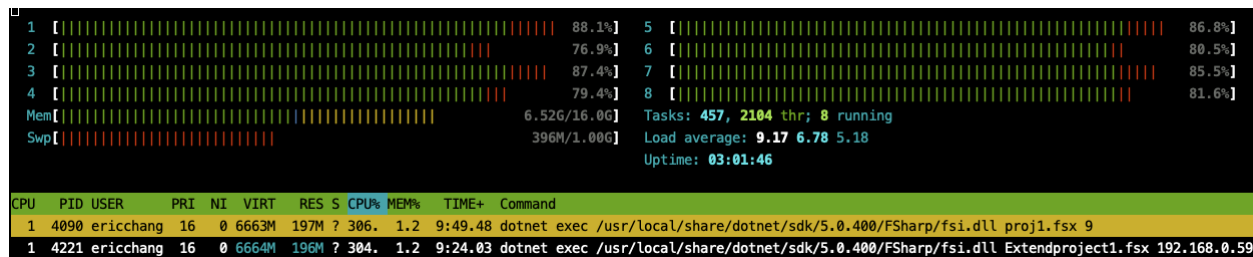
- Guessing the reason is too many actors which also causing context switching instead of executions.



### Figure 2. 4 actors for each process → better outcome

As a result of the experiment, the CPU is using in an efficient way showing in htop is as below

2 Process with different PID all have 300% up CPU%



## 2. The result of running your program for input 4

Input-String:

yimingchang;0996fdaf-a7ac-4ef0-92ab-4a1d82a1a841

Hashed String:

00006bd195bac5684ba68d006afba699291ef6ae861c9297ef0d5b68eaf77c27

**CPU/REAL Ratio = 2.75 with 4 workers**

```

Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0
Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0
Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0
Real: 00:00:00.006, CPU: 00:00:00.005, GC gen0: 0, gen1: 0, gen2: 0
Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0
[INFO] [09/23/2021 14:43:32] [Thread 0001] [remoting (akka://proj1Master)] Starting remoting
[INFO] [09/23/2021 14:43:32] [Thread 0001] [remoting (akka://proj1Master)] Remoting started; listening on addresses : [akka.tcp://proj1Master@192.168.0.59:5566]
[INFO] [09/23/2021 14:43:32] [Thread 0001] [remoting (akka://proj1Master)] Remoting now listens on addresses: [akka.tcp://proj1Master@192.168.0.59:5566]
input arguments: [|"proj1.fsx"; "4"|]
"Start course [COP5612] DOS Project 1 Mine Coins, start server on ip: 192.168.0.59"

yimingchang;0996fdaf-a7ac-4ef0-92ab-4a1d82a1a841 00006bd195bac5684ba68d006afba699291ef6ae861c9297ef0d5b68eaf77c27

[INFO] [09/23/2021 14:43:34] [Thread 0030] [remoting-terminator] Shutting down remote daemon.
[INFO] [09/23/2021 14:43:34] [Thread 0030] [remoting-terminator] Remote daemon shut down; proceeding with flushing remote transports.
[INFO] [09/23/2021 14:43:34] [Thread 0010] [remoting (akka://proj1Master)] Remoting shut down
[INFO] [09/23/2021 14:43:34] [Thread 0024] [remoting-terminator] Remoting shut down.
Real: 00:00:01.573, CPU: 00:00:04.321, GC gen0: 450, gen1: 1, gen2: 0
~/Documents/UFL/2021_Fall/COP5615-DistOperSysPrinc/COP5612_DOS/projects/Project1 structure/do-all-locally

```

3. The running time for the above as reported by time for the above and report the time. The ratio of CPU time to REAL TIME tells you how many cores were effectively used in the computation. If you are close to 1 you have almost no parallelism (points will be subtracted).

4. The coin with the most 0s you managed to find.

8 leading zeros, with 3 Machines,

Master (macOS 4 actors)

CPU/REAL = 3.092 (2389.551s / 748.999s)

Remote Local (macOS 4 actors)

CPU/REAL = 3.083

Remote 2 Windows (Windows 8 actors)

CPU/REAL = 5.6

Input-String:

**yimingchang;9c62437a-17de-4169-a5a1-4bac4ea8146c**

Hashed String:

**00000000e197e6baa6c1d9608c5bb7802042b3fa7f8e6d58205594623d071714**

```

yimingchang;9c62437a-17de-4169-a5a1-4bac4ea8146c 00000000e197e6baa6c1d9608c5bb7802042b3fa7f8e6d58205594623d071714

broadcast Coin found message to "akka.tcp://proj1Slave@192.168.0.140:5567/"
broadcast Coin found message to "akka.tcp://proj1Slave@192.168.0.59:5567/"
Recieve coin found signal, stop mining
CPU time = 2309551ms
Absolute time = 748999ms

```

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

4 machines (1 master and 3 client) in the same local network running  $k=6$