COP5612 Fall 2020

September 21 2020

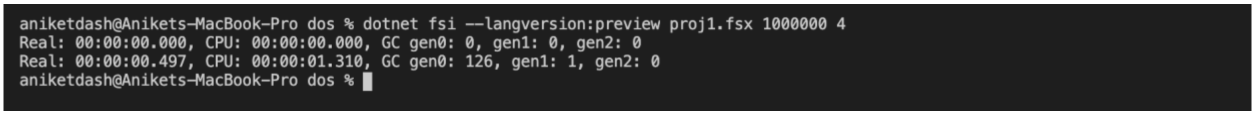
Avik Khamaru	36991467
Aniket Dash	75499549

Command to run file : dotnet fsi –langversion:preview proj1.fsx arg1 arg2

1.
The work unit we decided by dividing the entire problem range N by the no of cores in our work system. We are calling this work range. Work range = (N/ no of Cores in the system). Each successive child actor will work on the subproblem starting from beginning and ending at Work range. Each child actor will get one subproblem to solve and we are spawning actors equal to the no of cores in our system. The first child actor works from 1 to Work range, the second from (end of first actors work range) +1 to (end of first actors work range) +1+ work range. The process continues in this manner and each child actor is run Asynchronously.

We decided the Work unit as N/No of Cores, and this gave us optimal result using 100000 4 as input. we tried using large values for no of cores but there was slight dip in performance.

2.
The output for the required input

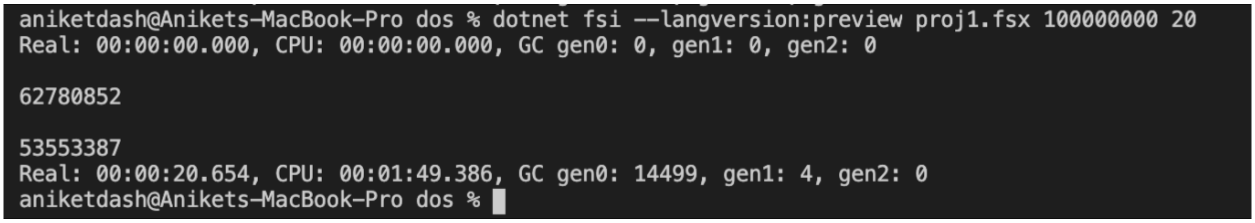


```
aniketdash@Anikets-MacBook-Pro dos % dotnet fsi --langversion:preview proj1.fsx 1000000 4
Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0
Real: 00:00:00.497, CPU: 00:00:01.310, GC gen0: 126, gen1: 1, gen2: 0
aniketdash@Anikets-MacBook-Pro dos %
```

3.
Run time fsharp proj1.fsx 100000 4 is 497 ms

The ratio of CPU time to Real Time is = 1.310/0.497 = 2.63

4.
The largest problem we were able to solve was 100000000 20



```
aniketdash@Anikets-MacBook-Pro dos % dotnet fsi --langversion:preview proj1.fsx 100000000 20
Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0

62780852

53553387
Real: 00:00:20.654, CPU: 00:01:49.386, GC gen0: 14499, gen1: 4, gen2: 0
aniketdash@Anikets-MacBook-Pro dos %
```