

Floyd-Warshall – Daniela Garcia 03/10/20

Problems

The thing I had the most trouble with was figuring out where to start and stop the timer in order to be able to time each individual thread and the program as a whole. I overcame this problem by just trying a lot of different things and figuring out what worked and what didn't.

Bugs

I am not able to run the program with 6 and 8 threads, it runs into an infinite loop

Time

It took me a bit over 3 hours to complete this assignment

Performance measurements

1 thread:

```
(Base) petegpete-Alienware-Area-51-R6:~/Desktop/parallel-floyd-warshall-dgarcia$ mpirun -n 1 python parallel.py

time for thread 0 : 0.7015855312347412
.....Running program with 1 threads.....
.....

Successful execution!

Total program time: 0.7209563255310059

.....subarray.....
[ 0 ][ 33 ][ 30 ][ 33 ][ 32 ][ 51 ][ 37 ][ 20 ][ 35 ]
[ 44 ][ 0 ][ 45 ][ 39 ][ 36 ][ 44 ][ 30 ][ 45 ][ 40 ]
[ 36 ][ 32 ][ 0 ][ 37 ][ 38 ][ 43 ][ 42 ][ 35 ][ 30 ]
[ 43 ][ 35 ][ 40 ][ 0 ][ 51 ][ 47 ][ 44 ][ 29 ][ 30 ]
[ 39 ][ 17 ][ 46 ][ 43 ][ 0 ][ 37 ][ 35 ][ 30 ][ 37 ]
[ 52 ][ 50 ][ 33 ][ 44 ][ 33 ][ 0 ][ 37 ][ 50 ][ 30 ]
[ 35 ][ 30 ][ 26 ][ 30 ][ 37 ][ 10 ][ 0 ][ 17 ][ 27 ]
[ 18 ][ 44 ][ 40 ][ 13 ][ 50 ][ 42 ][ 30 ][ 0 ][ 47 ]
[ 22 ][ 29 ][ 40 ][ 34 ][ 34 ][ 49 ][ 42 ][ 37 ][ 0 ]

.....expected output.....
[ 0 ][ 33 ][ 30 ][ 33 ][ 32 ][ 51 ][ 37 ][ 20 ][ 35 ]
[ 44 ][ 0 ][ 45 ][ 39 ][ 36 ][ 44 ][ 30 ][ 45 ][ 40 ]
[ 36 ][ 32 ][ 0 ][ 37 ][ 38 ][ 43 ][ 42 ][ 35 ][ 30 ]
[ 43 ][ 35 ][ 40 ][ 0 ][ 51 ][ 47 ][ 44 ][ 29 ][ 30 ]
[ 39 ][ 17 ][ 46 ][ 43 ][ 0 ][ 37 ][ 35 ][ 30 ][ 37 ]
[ 52 ][ 50 ][ 33 ][ 44 ][ 33 ][ 0 ][ 37 ][ 50 ][ 30 ]
[ 35 ][ 30 ][ 26 ][ 30 ][ 37 ][ 10 ][ 0 ][ 17 ][ 27 ]
[ 18 ][ 44 ][ 40 ][ 13 ][ 50 ][ 42 ][ 30 ][ 0 ][ 47 ]
[ 22 ][ 29 ][ 40 ][ 34 ][ 34 ][ 49 ][ 42 ][ 37 ][ 0 ]
```

2 threads:

```
(base) petegp@pete-Alienware-Area-51-R6:~/Desktop/parallel-floyd-warshall-dgarciaherk$ mpirun -n 2 python parallel.py

time for thread 1 : 0.3878915389966066

time for thread 0 : 0.3762633880586592
-----Running program with 2 threads-----
-----

Successful execution!

Total program time: 0.39728689193725586

-----Subarray-----
( 0 )( 33 )( 30 )( 33 )( 32 )( 51 )( 37 )( 20 )( 35 )
( 44 )( 0 )( 45 )( 39 )( 36 )( 44 )( 30 )( 45 )( 40 )
( 36 )( 32 )( 0 )( 37 )( 30 )( 43 )( 42 )( 35 )( 30 )
( 43 )( 35 )( 40 )( 0 )( 51 )( 47 )( 44 )( 29 )( 30 )
( 39 )( 17 )( 46 )( 43 )( 0 )( 37 )( 35 )( 30 )( 37 )
( 52 )( 50 )( 33 )( 44 )( 33 )( 0 )( 37 )( 50 )( 30 )
( 35 )( 30 )( 20 )( 30 )( 37 )( 10 )( 0 )( 17 )( 27 )
( 18 )( 44 )( 40 )( 13 )( 50 )( 42 )( 30 )( 0 )( 47 )
( 22 )( 29 )( 40 )( 34 )( 34 )( 49 )( 42 )( 37 )( 0 )

-----expected output-----
( 0 )( 33 )( 30 )( 33 )( 32 )( 51 )( 37 )( 20 )( 35 )
( 44 )( 0 )( 45 )( 39 )( 36 )( 44 )( 30 )( 45 )( 40 )
( 36 )( 32 )( 0 )( 37 )( 30 )( 43 )( 42 )( 35 )( 30 )
( 43 )( 35 )( 40 )( 0 )( 51 )( 47 )( 44 )( 29 )( 30 )
( 39 )( 17 )( 46 )( 43 )( 0 )( 37 )( 35 )( 30 )( 37 )
( 52 )( 50 )( 33 )( 44 )( 33 )( 0 )( 37 )( 50 )( 30 )
( 35 )( 30 )( 20 )( 30 )( 37 )( 10 )( 0 )( 17 )( 27 )
( 18 )( 44 )( 40 )( 13 )( 50 )( 42 )( 30 )( 0 )( 47 )
( 22 )( 29 )( 40 )( 34 )( 34 )( 49 )( 42 )( 37 )( 0 )
```

3 threads:

```
(base) petegp@pete-Alienware-Area-51-R6:~/Desktop/parallel-floyd-warshall-dgarciaherk$ mpirun -n 4 python parallel.py

time for thread 2 : 0.1925762694891797

time for thread 1 : 0.2824235725482832

time for thread 3 : 0.28213556289672852

time for thread 0 : 0.28277829386838273
-----Running program with 4 threads-----
-----

Successful execution!

Total program time: 0.22362351417541584

-----Subarray-----
( 0 )( 33 )( 30 )( 33 )( 32 )( 51 )( 37 )( 20 )( 35 )
( 44 )( 0 )( 45 )( 39 )( 36 )( 44 )( 30 )( 45 )( 40 )
( 36 )( 32 )( 0 )( 37 )( 30 )( 43 )( 42 )( 35 )( 30 )
( 43 )( 35 )( 40 )( 0 )( 51 )( 47 )( 44 )( 29 )( 30 )
( 39 )( 17 )( 46 )( 43 )( 0 )( 37 )( 35 )( 30 )( 37 )
( 52 )( 50 )( 33 )( 44 )( 33 )( 0 )( 37 )( 50 )( 30 )
( 35 )( 30 )( 20 )( 30 )( 37 )( 10 )( 0 )( 17 )( 27 )
( 18 )( 44 )( 40 )( 13 )( 50 )( 42 )( 30 )( 0 )( 47 )
( 22 )( 29 )( 40 )( 34 )( 34 )( 49 )( 42 )( 37 )( 0 )

-----expected output-----
( 0 )( 33 )( 30 )( 33 )( 32 )( 51 )( 37 )( 20 )( 35 )
( 44 )( 0 )( 45 )( 39 )( 36 )( 44 )( 30 )( 45 )( 40 )
( 36 )( 32 )( 0 )( 37 )( 30 )( 43 )( 42 )( 35 )( 30 )
( 43 )( 35 )( 40 )( 0 )( 51 )( 47 )( 44 )( 29 )( 30 )
( 39 )( 17 )( 46 )( 43 )( 0 )( 37 )( 35 )( 30 )( 37 )
( 52 )( 50 )( 33 )( 44 )( 33 )( 0 )( 37 )( 50 )( 30 )
( 35 )( 30 )( 20 )( 30 )( 37 )( 10 )( 0 )( 17 )( 27 )
( 18 )( 44 )( 40 )( 13 )( 50 )( 42 )( 30 )( 0 )( 47 )
( 22 )( 29 )( 40 )( 34 )( 34 )( 49 )( 42 )( 37 )( 0 )
```

Analysis of times

The time of execution gets smaller as you add more threads. This makes sense as they all work on a piece at the same time so the work should be taking less time when it is split between more processes.

Observations

I found MPI a bit more intuitive to use than the library we had been using for the previous 2 labs. I have been having a hard time testing these labs since my computer can only run 4 threads and I have been

asking a friend to run it on his computer for me. I am not sure if this makes a difference, but he ran it on his computer rather than the VM we were given.

dumpCPUInfo.sh output

AMD Ryzen 8 core i7-2700X CPU @ 3.7GHz