Andrii Lunin

Professor Mohamed Zahran

Parallel Processing

Lab 1

November 11, 2020

# Lab 1

1) Once the module is loaded with `module load mpi/openmpi-x86_64`, you can run my program.

2) I have provided a handy Makefile, for compiling.

   1) On $make it complies the genprimes.c file into the executive genprimes.
   2) On $make clean, it **removes** the *executive* file **AND** all of the *.txt* files. Be mindful of that if you run it in some other folder apart from the folder with my code.

3) For executing the program run $mpiexec -n <P> genprimes <N>, where P stands for the number of processes, and N stands for the upper bound on the range of primes.

4) I have to note, that I tailored my program to be as *quick* and as *efficient* as possible for large numbers. I have explained my algorithm in detail in comments in the C file, but I wanted to make sure to assert that the best efficiency comes with $P > 10M$. Once you consider $> 1B$, the speedup for a $100$ processes is massive (I have provided the times for $p = 1B$ for $10$ and $100$ processes in the table below).

## Time

| #P vs N | 1,000 | 10,000 | 100,000 | 1,000,000 | 10,000,000 | Extra: 1B |
|---|---|---|---|---|---|---|
| 1 | 0.249s | 0.250s | 0.273s | 0.701s | 10.946s | X |
| 2 | 0.265s | 0.265s | 0.273s | 0.457s | 4.485s | X |
| 5 | 0.298s | 0.296s | 0.301s | 0.380s | 1.645s | X |
| 10 | 0.370s | 0.372s | 0.377s | 0.417s | 1.023s | 289.819s |
| 100 | 3.721s | 3.766s | 3.703s | 3.678s | 4.072s | 42.161s |

## Speedup

| #P vs N | 1,000 | 10,000 | 100,000 | 1,000,000 | 10,000,000 |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0.939 | 0.943 | 1 | 1.534 | 2.44 |
| 5 | 0.836 | 0.845 | 0.907 | 1.845 | 6.654 |
| 10 | 0.839 | 0.672 | 0.724 | 1.681 | 10.7 |
| 100 | 0.067 | 0.066 | 0.074 | 0.190 | 2.688 |

## Efficiency

| #P vs N | 1,000 | 10,000 | 100,000 | 1,000,000 | 10,000,000 |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0.4695 | 0.4715 | 0.5 | 0.767 | 1.22 |
| 5 | 0.1672 | 0.169 | 0.1814 | 0.369 | 1.3308 |
| 10 | 0.0839 | 0.0672 | 0.0724 | 0.1681 | 1.07 |
| 100 | 0.00067 | 0.00066 | 0.00074 | 0.0019 | 0.02688 |

*Note: all times provided are averaged over 10 consecutive executions.