Count Primes Using MPI

By Congying Wang

Spring 2018 CSE633
Instructor : Dr. Russ Miller

Why did I choose this topic

MPI primarily addresses the message-passing parallel programming model：data is moved from the address space of one process to that of another process through cooperative operations on each process.

Why MPI:

Standardization, Portability, Performance (Vendor implementations), Availability, Functionality

This topic can clearly show how parallel computing has a better performance by comparing with the sequential computing.

Why we count prime number

They are a mathematical mystery

One of the most widely used applications of prime numbers in computing is the RSA encryption system

Large prime numbers are used prominently in other crypto-systems

The largest known prime is 243,112,609 – 1

The algorithm for generating the primes

trial division algorithm

Trail division divides an n-bit random number by primes up to sqrt(n)

- Accept some input integer n
- For *each integer x* from {2 ... sqrt(n)} check if *x divides n*
- If you found a divisor then *n is composite* OR ELSE *n is prime*

Use MPI to distribute workload

- If there are n workers, the i th node starts with the 2i-1th value.
- Worker then checks every 2n values from its start position until all numbers have been checked.
- Worker sends found primes to master node as soon as they are found.

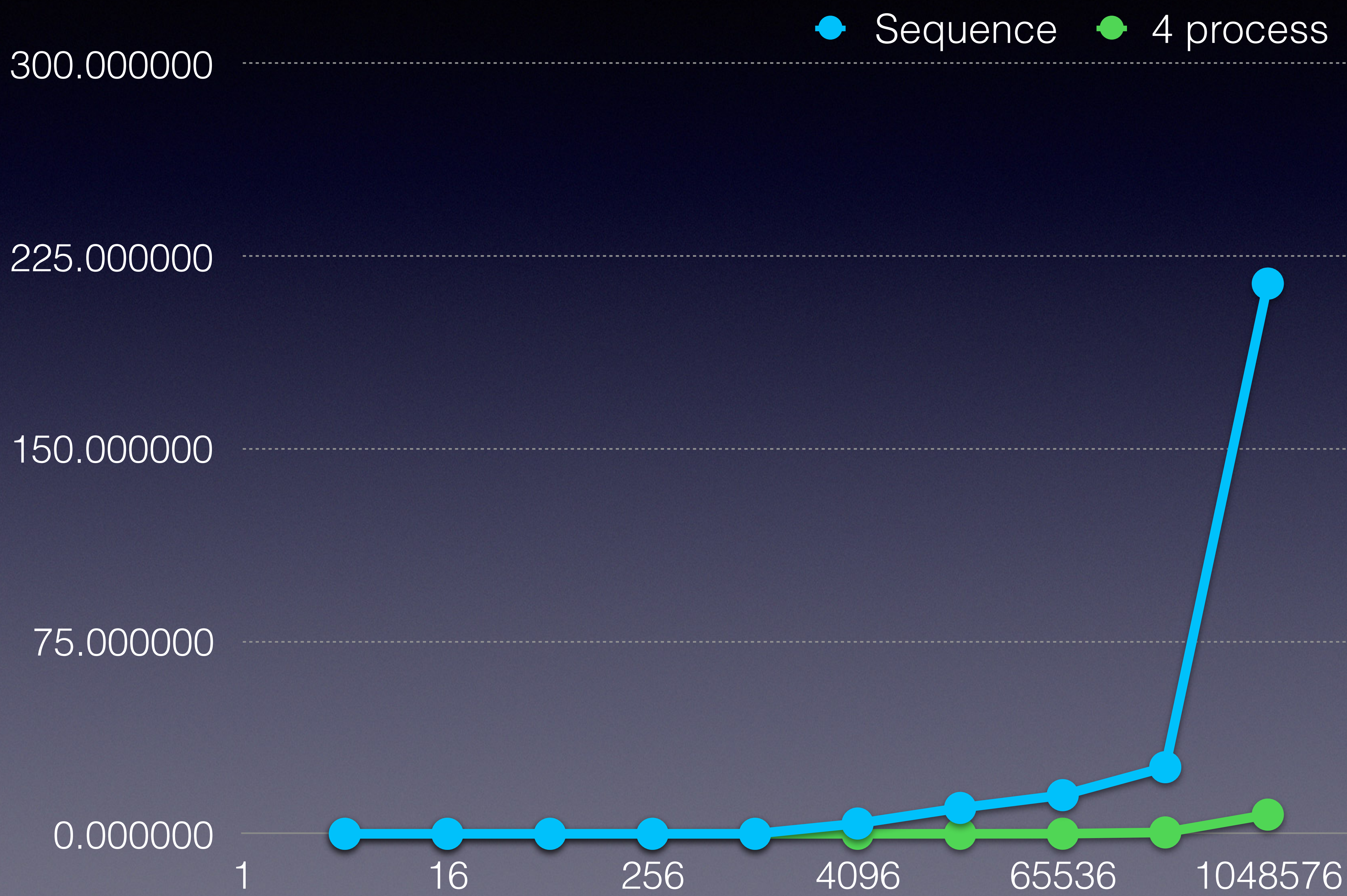
Experiment Details

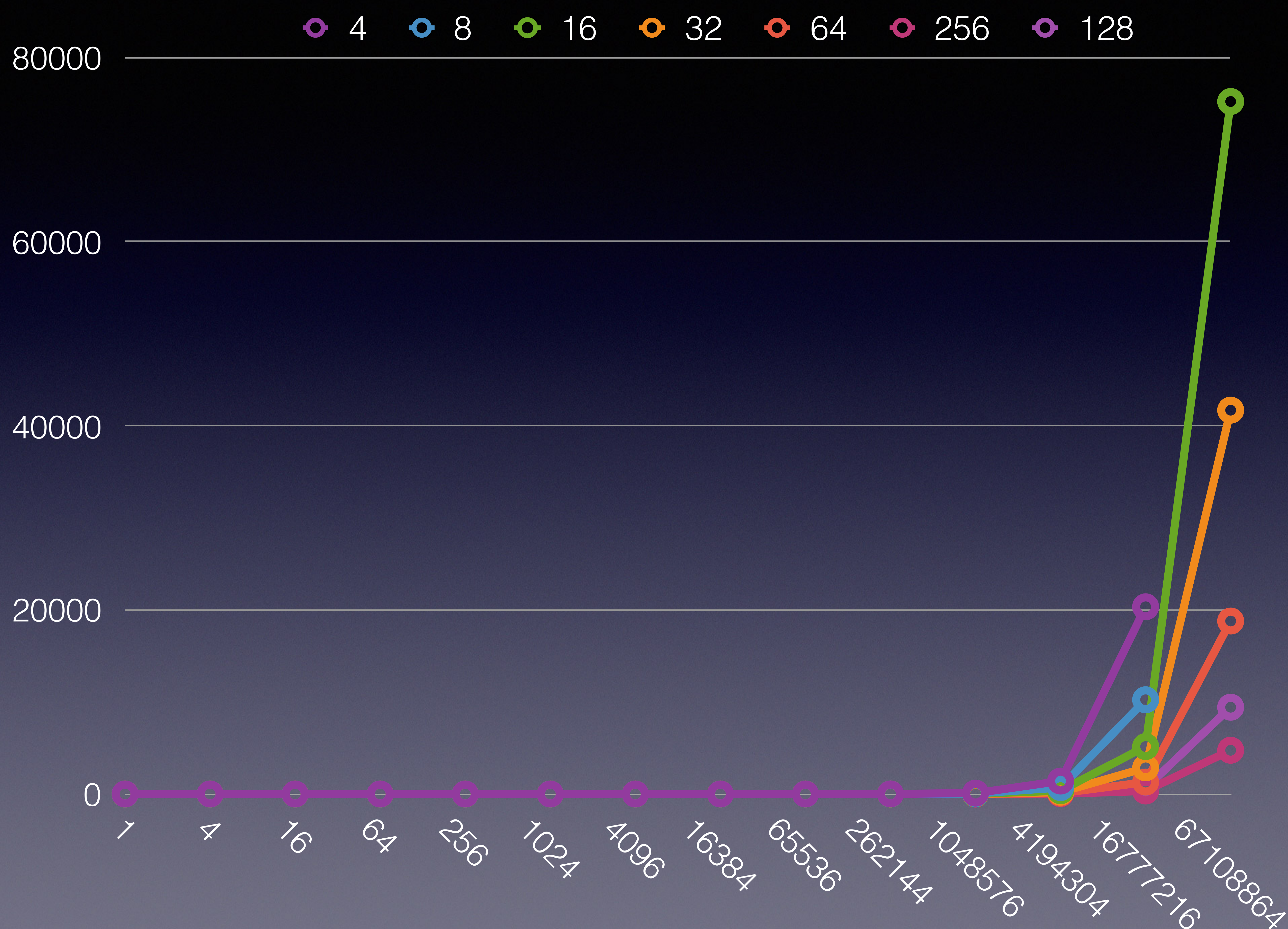
- Using Intel MPI
    - This implementation has multi-network support (TCP/IP, Infiniband, Myrinet, etc.)
    - Compiler "wrappers" around both Intel's compiler suite (mpiifort, mpiicc, mpiicpc) and the GNU compilers (mpif90, mpicc, mpicxx)
    - This implementation runs over **InfiniBand**.

MPI usage

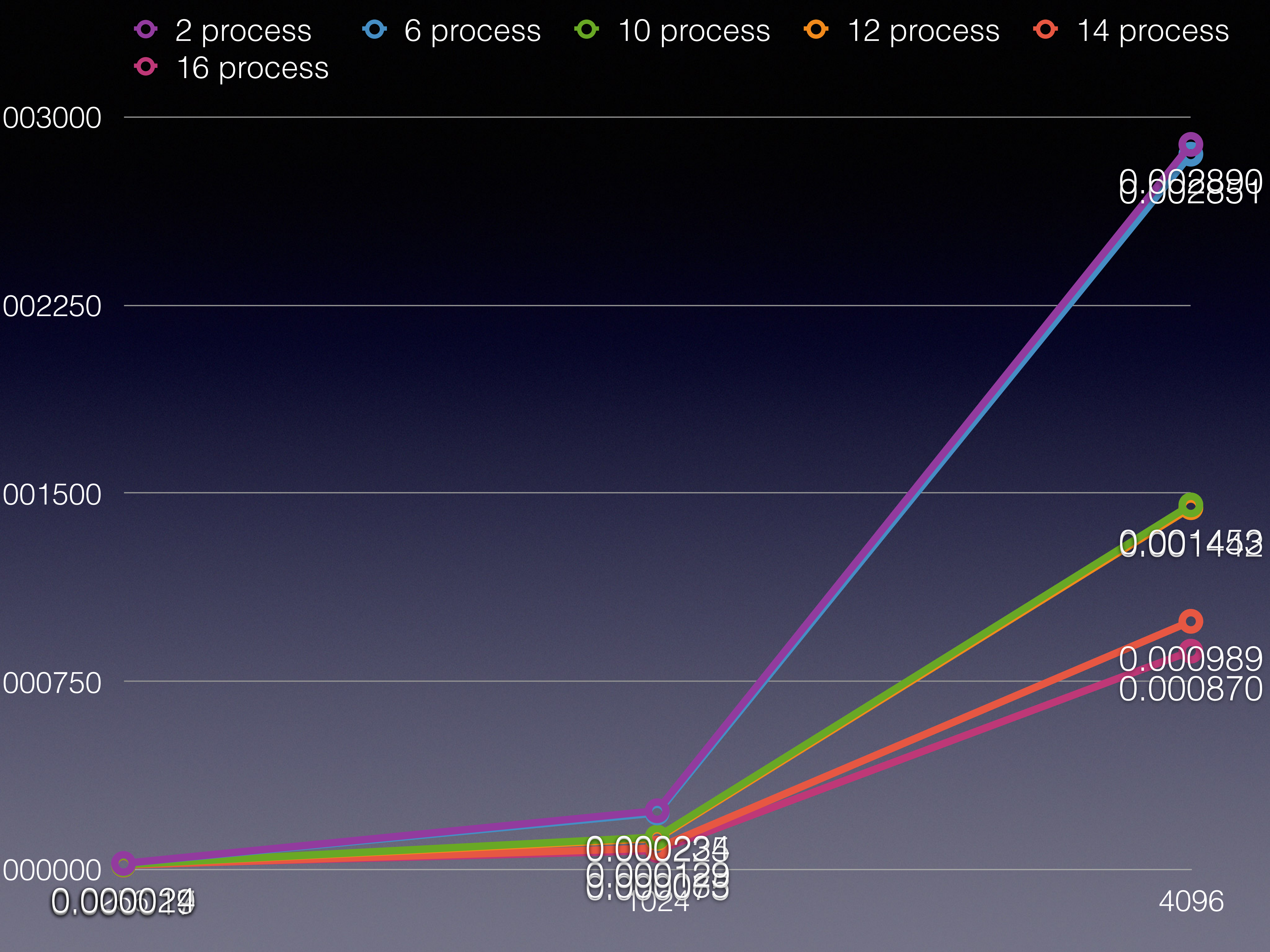
```
[user@rush mpi-stuff]$ module load intel/14.0

[user@rush mpi-stuff]$ module load intel-mpi/4.1.3

[user@rush mpi-stuff]$ mpiicc -o cpi cpi.c

[user@rush mpi-stuff]$ mpiexec.hydra -n 2 ./cpi
```

While using sequential

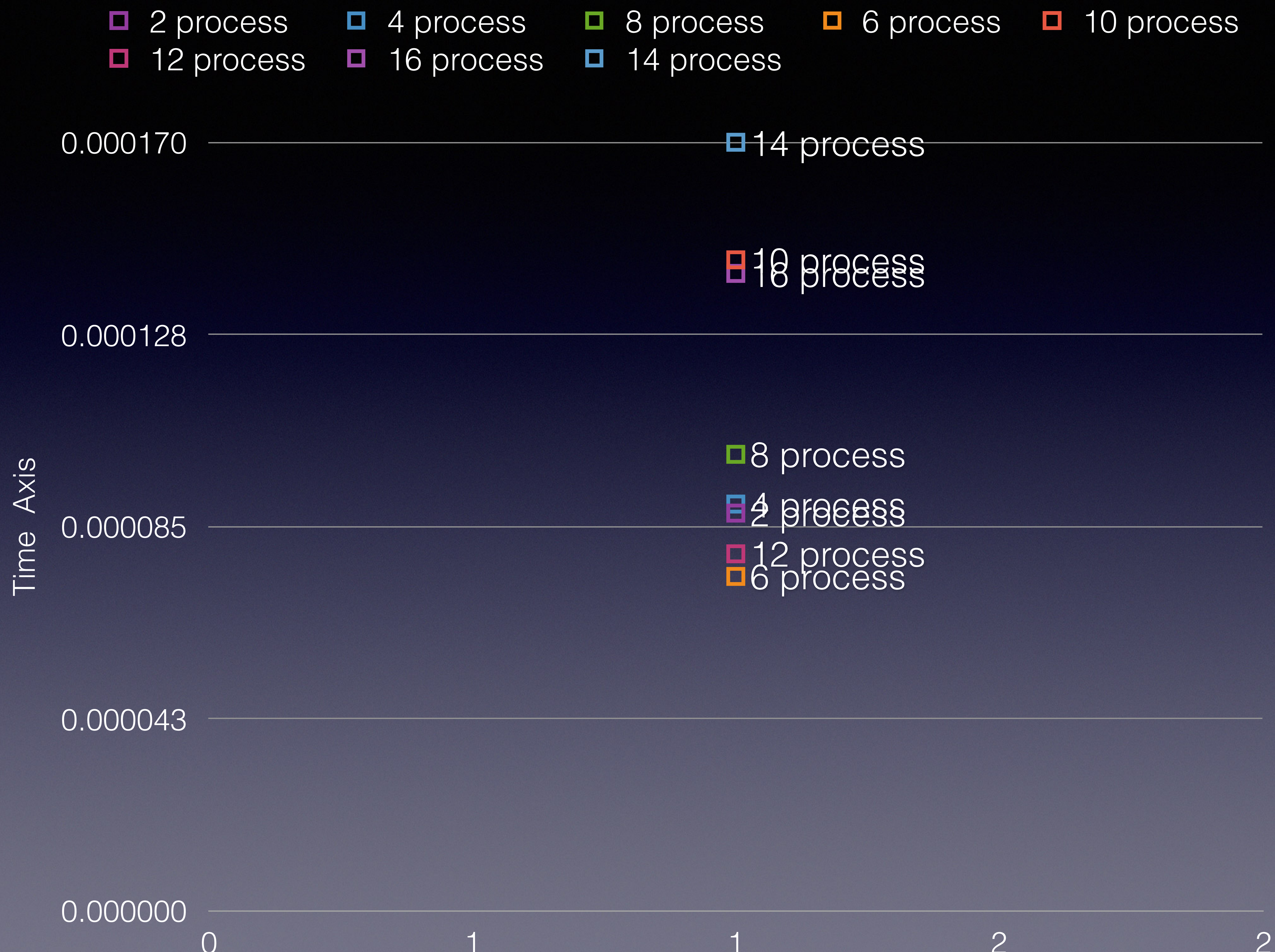



4
8
16
32
64
256
128
80000
60000
40000
20000
0
1
4
16
64
256
1024
4096
16384
65536
262144
1048576
4194304
16777216
67108864

But while Compare small number with different number of processes


2 process
6 process
10 process
12 process
14 process
16 process
003000
002250
001500
000750
000000
0.000989
0.000870
4096

Let’s see more detail


2 process
4 process
8 process
6 process
10 process
12 process
16 process
14 process
0.000170
0.000128
0.000085
0.000043
0.000000
Time Axis
14 process
10 process
16 process
8 process
4 process
2 process
12 process
6 process
0
1
1
2
2

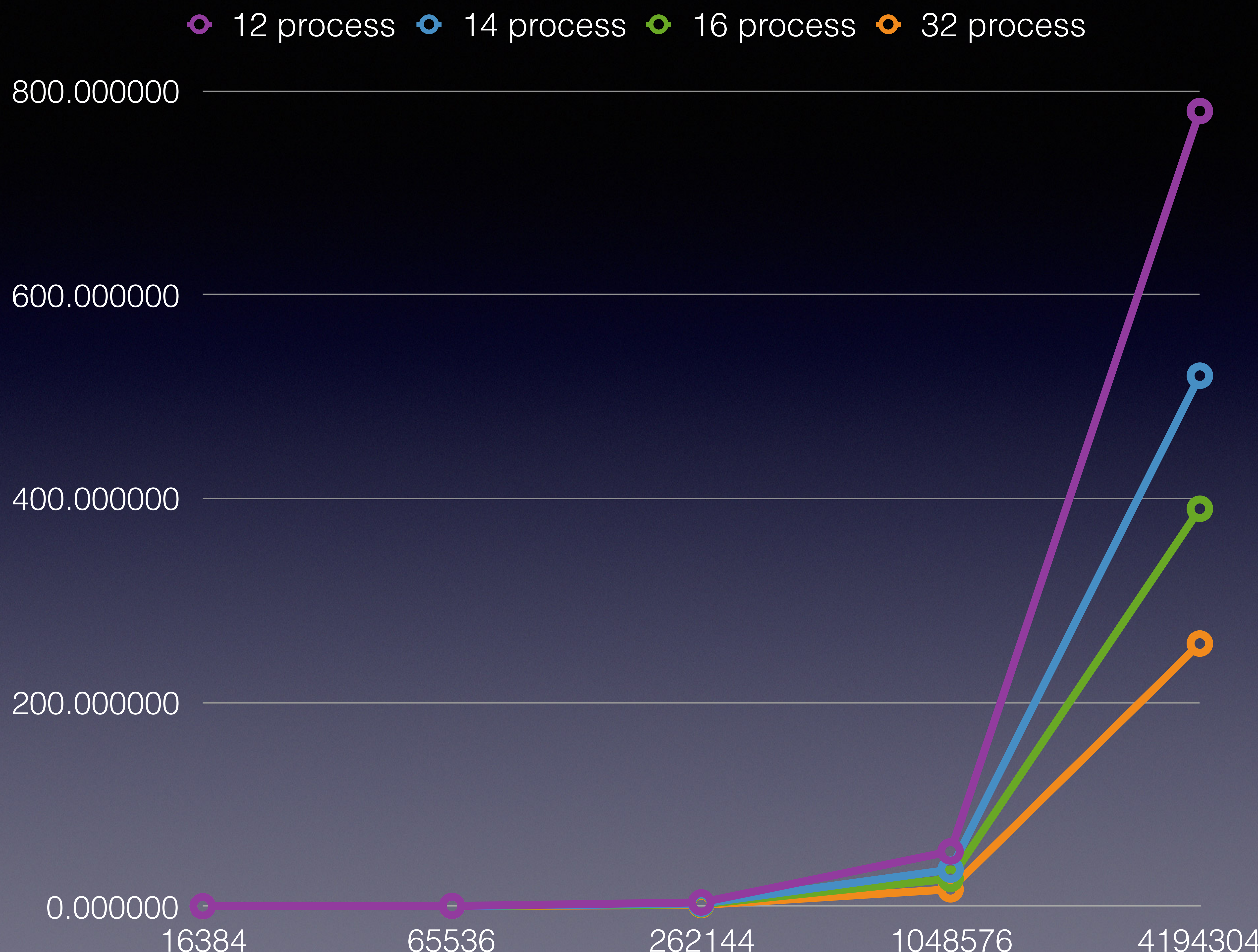
- More process doesn’t mean faster
- 12 process even slower than 4 process

Jobs with very small numbers are bound by communication time.

Since sequential runtime is so small, the time to send found primes to the head node makes the program take longer with more nodes, and makes adding processors slow down the program’s runtime.

Parallel execution of these computations is impractical.

We don’t need parallel execution for small number


12 process
14 process
16 process
32 process
800.000000
600.000000
400.000000
200.000000
0.000000
16384
65536
262144
1048576
4194304

Observation

- The difference happens when number is large.
- The counting prime algorithm also influence the running time .
- The communication time cannot be ignored. When number of processor is increasing, the efficiency of parallel algorithm drops, cost of communication increases.

Thinking

- Is there any way other than mpi can make counting prime more quick

- Other algorithm better than trial division. The paper and example I read about all using trial division

- If the finding range is limited in a small range, parallel computing is not needed.

- Is there any formula to calculate best amount of process

Something I learned

- Write C and MPI
    - C for me is hard
- Everything is not absolute
    - You can only say parallel computing is helping when your data is large enough

Reference

- Introduction of MPI
- MPI usage handbook
- programming MPI with C
- Parallelization of Prime Number Generation Using Message Passing Interface
- Why do we need to know about prime numbers with millions of digits?

Thank you for your listening !