

PA1 Report
Adam Montano
CS 415

The three parts of this project tested the amount of time when communicating through different tasks. The first part determined the time between two processes on one machine, the second part on two separate machines. The third part determined how the communication time differed based on the amount of data passed between two tasks.

One Box

The “one box” calculated the amount of time when sending one integer between two tasks on the same machine. MPI send and receive was used to pass a single integer and WTime calculated the time between the tasks. The project utilized loops to do this a certain amount of times in order to calculate an average.

Output from /log/one_box.log :

```
time between 0 and 1 is: 0.000012
```

Two Box

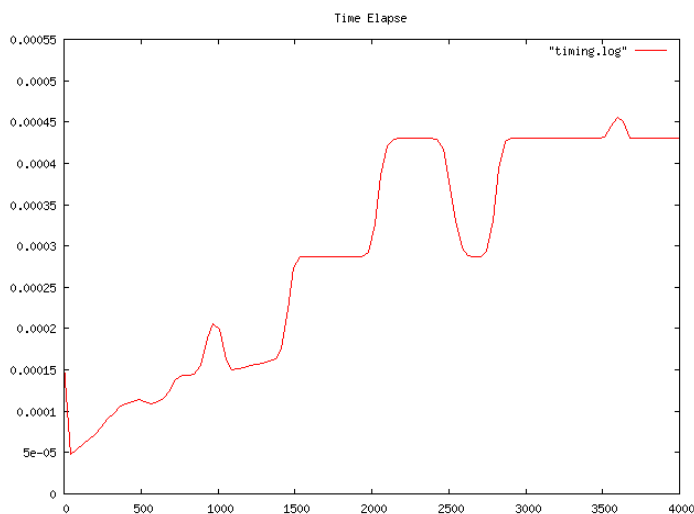
The “two box” calculated the amount of time when sending one integer between two tasks with each on separate machines. MPI send and received was used to pass a single integer and Wtime calculated the time between machines. The project utilized loops to do this a certain amount of times in order to calculate an average.

Output from /log/two_box.log :

```
time between 0 and 1 is: 0.000297
```

Timing

The “timing” calculated how the amount of time differed when increasing the amount of integers sent. MPI send and receive was used to calculate this and incremented the second parameter “count” to send a greater number of integers each time. By summing the duration calculated with Wtime, an average was taken from this and calculated. The below graph represents how the time increases while incrementing the amount of integers being passed.



The time clearly increases as the amount of integers increases. However, there are times when the graph “flattens”. This must be because a different protocol is chosen for the amount of integers being passed. The dips and valleys could be communication congestion and slurm is optimizing the connections as best it can. The first peak must be from over estimating the size of the packet by slurm in order to prevent loss of data.

Issues

The largest issue was determining how to specify one machine or two machines. I had to get accustomed to the sbatch documentation even though I was focused on the MPI code. The other issue was understanding how the MPI can send an increasing amount of data. Originally I just had “send” in a for loop and “recv” in another so that the tasks essentially just “spammed” each other repeatedly. However, the proper way was to increase the count parameter of each.

