

Project 1

This project had us send and receive messages from different processes. There were three different parts to this assignment. The first was sending a message from one processor to another processor in the same box. Then sending a message back to the original processor. The second part was the same as the first, but sending the message to a different box. The third part was sending more and more integers until we could find a jump in the time it takes for a processor on a different box to receive them.

Part One-

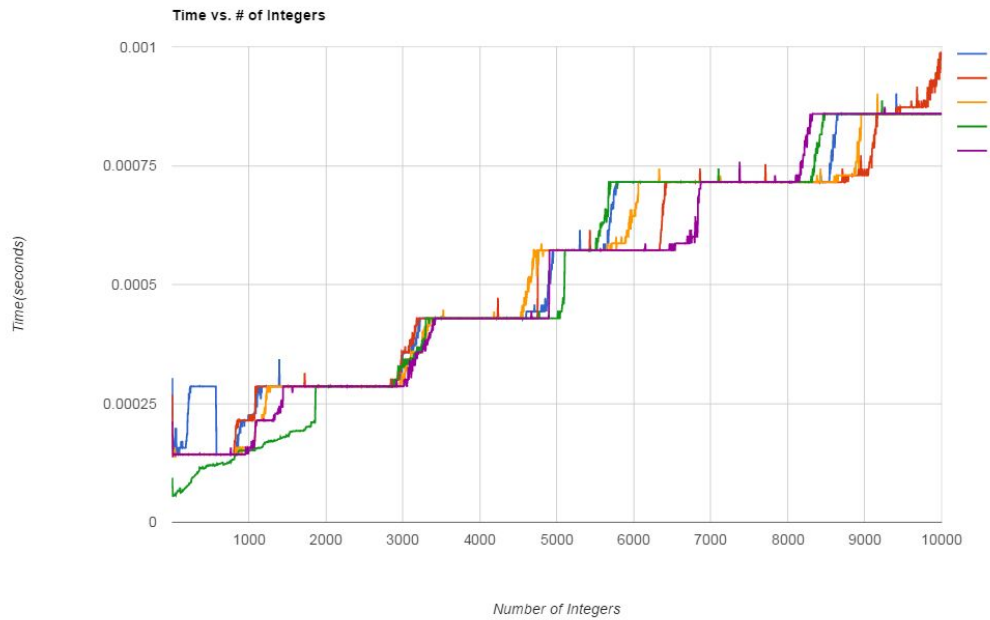
For the first part of this project, I sent a message using `MPI_SEND` and `MPI_RECEIVE` between two processor on the same box. I sent a message 1000 times and calculated the average time it took to send and receive a message.

Part Two-

The second part of the project used the exact same code as part one. The only difference was using `-N2` in the `srun` statement. This causes `SLURM` to use two processor on different boxes.

Part Three-

The third part of this project was seeing how many integers could be passed into the buffer before it could not fit anymore into the packet to the other box. The code I wrote increased the amount in the buffer by five every iteration. It went until it reached 10000 ints. For the timing I sent ten integers with the same size and averaged the time it took to go to the other box and back. This allowed for a much more accurate graph since the numbers would be normalized. The numbers were then placed into excel in order to get a graph shown below.



This graph shows the amount of time taken for five different trails. The first 2000 numbers is always a little bit messy. This could be due to interference between the network as many other people were testing out their code. However, the rest of the graph shows clear points where the time jumps up. These times are around 3000, 5000, 6500, and 8500. Averaging the numbers, I would say the jumps occur around 1800 integers.