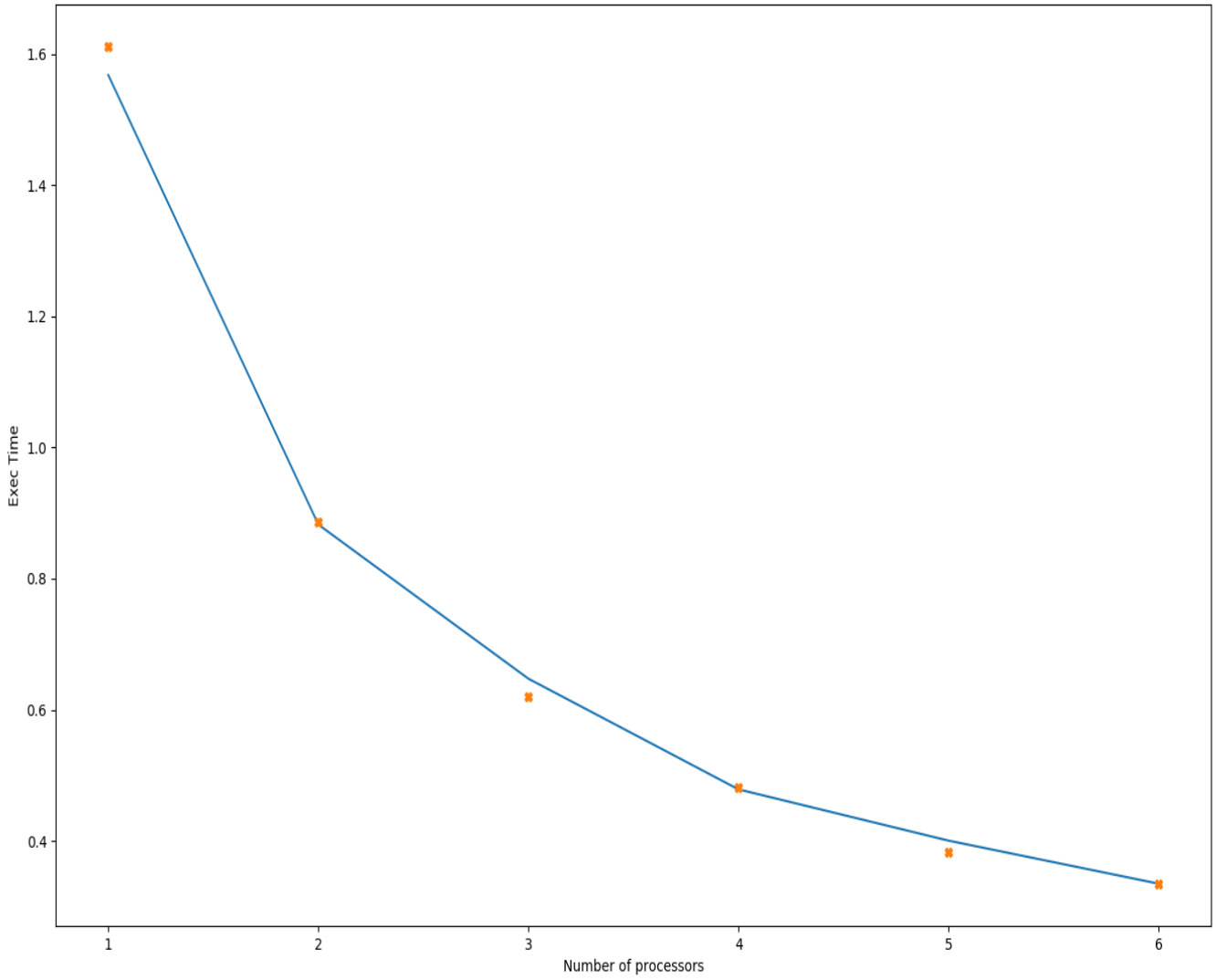
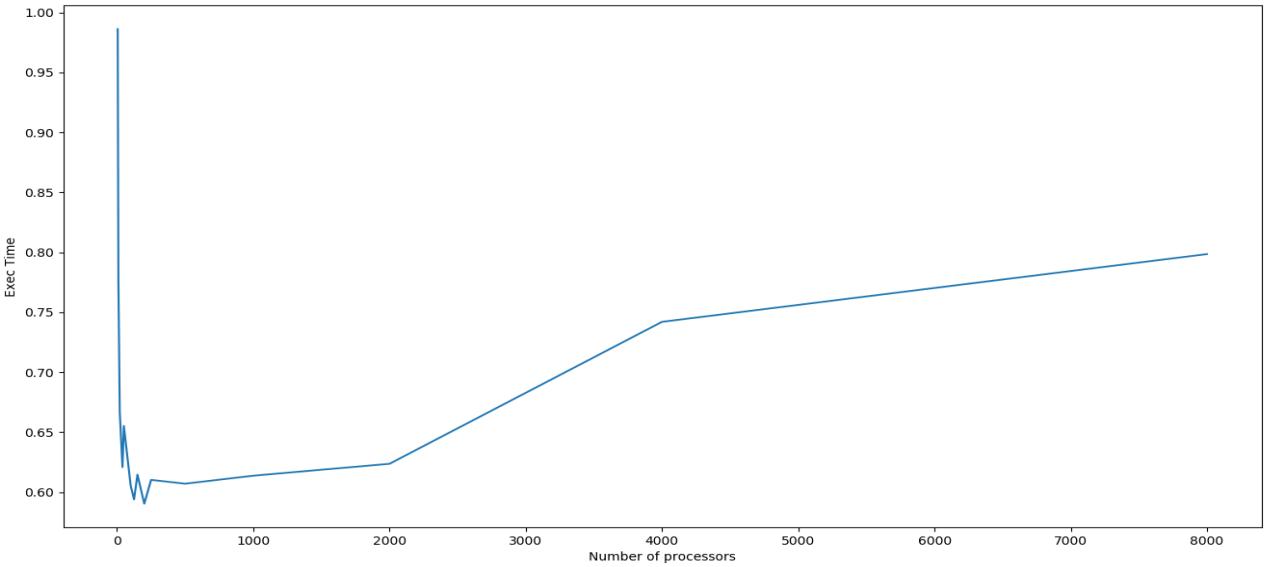
I wrote the MPI version of this stencil algorithm, and compared it to the OMP version where I parallelized the for loop. In the MPI version, I did x iterations where x was the buffer size to deplete the buffers, the did MPI communication to refill them. The sends and recieves form a chain that goes from the process 0 to p, then backwards to 0, filling the buffers on each side of each processors array of elements.

Here is a graph comparing OMP and MPI. The red X marks represent the OMP execution times and the blue line.



Both methods seem to work just as well as each other, for these input sizes and amount of processors.

I also tracked how the buffersize affects the execution time for a size of 480000 elements with 12000 iterations, running on 12 cores between two machines.



There seems to be a wide area with similar execution times between 30 and 500, that all produce good results, and there was enough variance in the data that it was hard to compare results from buffer sizes in this range to each other, because the execution time could be somewhat volatile run to run.