Project 4 Report – MPI Hotplate

## Kyle Pontius

# Introduction

Parallel processing is a powerful tool that requires careful planning. If we want to tackle large problems, with lots of data, we have two general approaches that we’ve touched on this class. First is multithreaded applications. This consists of many threads running simultaneously on a single machine, preferably with a multi-core processor. An alternative we’ve studied is networked distributed processing using MPI. This distributes pieces of a problem to multiple machines, which then return a piece of the result to be merged.

For me, this problem posed several interesting questions, some of which were: How will the communication time stack up against OMP’s nearly instant communication time? Do separate nodes with a given problem size perform better than separate cores (working on a thread) with that same problem size? Which type of parallelism’s performance degrades more quickly as problem size significantly increases? While not each of these questions will specifically be answered, there were many that were. This was a fun and meaningful learning activity for me.

# Approach

Please note, as I haven’t finished the p-threads project my results will focus on the performance results for OpenMP and MPI. For my tests, I’ve decided to run both the OMP code and MPI code on the FSL Supercomputer.