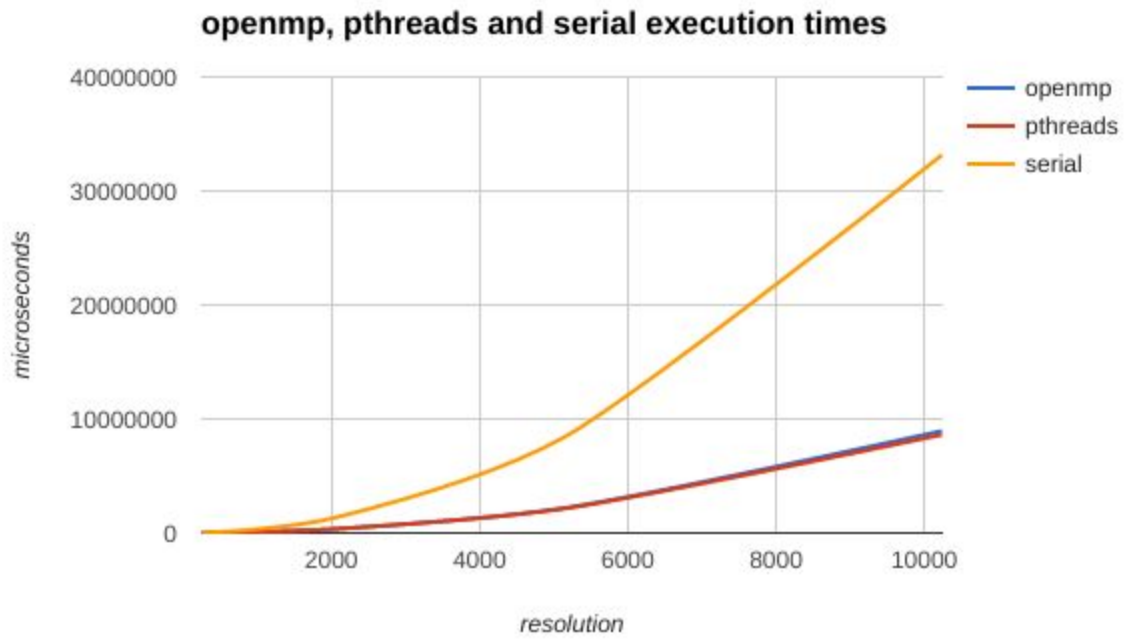
Cameron Franke
3/4/17
High Performance Computing



The chart above what produced by running performance tests on the different size “Lenna” images available on canvas. Both parallelization libraries used all available threads on the test machine, in this case; 8. OpenMP and Pthreads both exhibited significant speedup relative to the serial implementation at approximately 3.8X. At each resolution the Pthreads implementation beat the OpenMP implementation by a very small margin. This fit with my expectations of the systems. I assumed that OpenMP would incur a larger overhead due to its high level of automation. At 10240x10240 the Pthread implementation beat the OpenMP implementation by only 1.03%. While in some tasks this may be a significant enough margin to matter, in this case it is negligible. One of the most relevant metrics in the comparison between OpenMP and Pthreads in this case is speedup/programmer time. Assume that the speedup of OpenMP and Pthreads is equal, that it took me 2 hours to implement my Pthreads code and that it took me 10 minutes to implement the OpenMP code*. Under these conditions OpenMP has provided me with a speedup rate of 22.8x/programmer hour while Pthreads only offers a speedup rate of 1.9X/programmer hour. The time investment necessary to achieve very high levels of speedup with OpenMP is staggeringly low.

*In reality it took more more than 2 hours to implement the Pthreads version of this algorithm