

Parallel Sorting Algorithms

In this project I will be exploring the domain of sorting algorithms. My motivation for choosing this domain is to gain a deeper understanding of parallel programming techniques and to evaluate the performance of parallelizing sorting algorithms. Computation plays a significant role in sorting algorithms, as there is often an issue of sorting large amounts of data in a short amount of time. Parallelization of these sorting algorithms can be useful in speeding up these time-consuming processes. The software I will be exploring will be a variety of sorting algorithms such as quick sort, merge sort, bubble sort, etc. I will be using the HPCC to help me conduct this study as well as my own personal computer to compare the time differences. A successful outcome for this project would be a significant speedup in sorting algorithms given a large dataset.

I will be reviewing existing sorting algorithms and be parallelizing them to run faster. Some specific algorithms I will be reviewing will be quick sort, merge sort, and bubble sort. OpenMP and MPI will be very helpful with providing support for my project. Some resources I will be using are the OpenMP specifications, and the MPI specifications pages. I will have a list of instructions for running the code in a readme file on the HPC.

The code that I will be benchmarking will be code used for existing sorting algorithms. I will be implementing the algorithms in a non-parallel version and then create a parallel version after, evaluating the time required to run the code given a data set. I will be looking for ways to optimize my software through evaluating code structure, data distribution, etc. I would evaluate the performance of the code by measuring the execution time of each script and algorithm I make and comparing it to the parallel version of the algorithm. My project would be successful if the parallel version of my sorting algorithms is substantially faster than the original version. I am hoping and

expecting that through creating parallel versions of the sorting algorithms, I will create a faster execution time of the algorithms.