PDC Proposal

Members : Bilal Ali (21k-3153), Ehaab Tariq Ahmed (21k-3372)

Problem:

Examining the effect parallelism has on time complexity of bubble sort, insertion sort and bucket sort when they are used to sort a large dataset.

Insertion sort is a very quick algorithm while bubble sort is usually discarded because of its O(N^2) complexity. We are aiming to see if insertion sort can be made even faster and if bubble sort can be optimized to compete with more popular sorting algorithms when it is parallelized.

Testing:

We would get a large dataset of numbers, approximately 100K to 200K.

OpenMP and MPICH code would be written to optimize each algorithm. The results would be tested on multiple threads and processes (eg 10 threads/processes, 20, 30 and so on).

We would note down the time taken to sort the dataset with each amount of threads/processes and graph the results.

Proposed Techniques:

Bucket Sort = Its main logic is to spare different parts of array into different buckets, then sort them at the same time with another algorithm and put it back together in one big result array.

Insertion Sort = split data into chunks, finer chunks than already available implementation, each thread cycles between chunks

Bubble Sort = Use nested parallelism (turned off by default in Open MP) and split data into chunks so each thread can parallelize 2 for loops.