

Ryan Devaney

3/29/17

CS 415

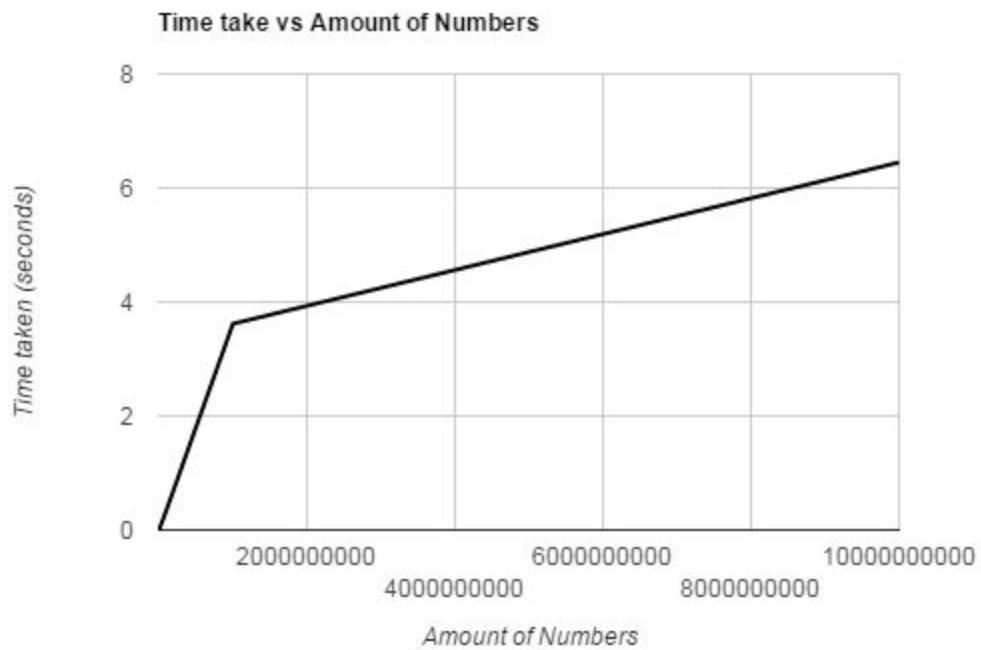
PA03 - Bucket Sort

Description of Assignment

This assignment has us parallelize bucket sort. For this project we first need to write a sequential bucket sort so that we can compare it to a parallelized version of bucket sort. For this project we need to test a variety of numbers in order to see how the two times compare as the amount of numbers gets larger and larger. Various graphs will be provided to visually aid analysis of the results of sequential bucket sort against parallel bucket sort.

Part 1 - Sequential Bucket Sort

For this part of the project we were to write a sequential bucket sort that would work in the cluster. The first thing to do was write a small program that would easily give a large amount of numbers to a data file. This is a lot better than manually writing out 1 billion numbers by hand. The program then has to read in the amount of numbers there are to sort, and then the numbers themselves. After that the numbers are moved to buckets. After that, the buckets are sorted and concatenated from smallest bucket to largest bucket which results in a sorted list of numbers. We take the time from the end of reading in the numbers from the file to when the bucket sort is complete. We do not time IO operations as they would be about the same in sequential and parallel versions of bucket sort.



This chart shows a line describing the time taken compared to the amount of numbers that there were to sort. The smallest amount of numbers used to sort was 100000 which took 0.002912522 seconds on average. The largest amount of numbers used was 10 billion which took 6.44888 seconds on average. As the graph shows as the amount of numbers goes up it increases very fast until at about 1 billion numbers. It then follows a more linear fashion.