CS262 Project Proposal: Distributed Computing on Large Datasets

Divya Amirtharaj, Siona Prasad, and Katherine Zhang

April 2, 2023

1 Main Problem

In this project, we hope to implement a distributed data processing algorithm, in a similar form to MapReduce, which distributes some processing task across multiple machines and then combines the result to create the desired output. This project would involve implementing a distributed system that can take in a dataset, distribute a time-expensive processing task such as filtering or sorting across multiple machines, return the correct result, and be tolerant to some machine failure, as well as comparing this distributed system to an equivalent centralized system.

2 Motivation

With large datasets becoming more prevalent, we find that the issue of processing said data in an efficient manner is necessary for tasks such as machine learning, recommendation algorithms, fraud detection, etc. Although individual computers with large amounts of compute power may exist, it is still more time- and cost-effective for applications from small businesses to government programs to distribute their data processing across multiple machines. Additionally, processing data in parallel comes with the same advantages that any distributed system does – it is more scalable, reliable, and harbors no individual point of failure. We feel that implementing such an algorithm will involve making interesting design and implementation choices, as well as applying the principles of scalability, efficiency, and fault-tolerance that have been discussed throughout this course.

3 Desired Result

We want to show that our distributed data processing methodology will accomplish the desired task in significantly less time than a centralized system would. Consequently, there should be more computations performed within a unit of time than there would be with a

centralized system. Finally, we will show that our system is fault tolerant for one or more machines. To show these three things, we will measure the latency and throughput, and demonstrate that the computation is performed even if one or more nodes fails, and compare these results to those of a the same computation on a centralized system.

Our deliverable will be code that implements distributed sorting, as well as a regular sorting algorithm. It will include tests that measure the latency, throughput, and reaction to failures of nodes for both algorithms. In order to best test our code we will run experiments with different sized datasets, as well as different levels of prepartitioning. Finally, we will do a short written analysis as part of our engineering notebook that demonstrates the difference in performance of each system.