George Iadze (gi40)
Johnny Polanco (jap442)
Eva Tsang (tct38)

WordCount Assignment

Most challenging part of this assignment was figuring out all the details of setting up the AWS environment properly. That took most of the time as well as installing and configuring HADOOP on local machine to compile the WordCount.java file. We spent hours upon hours trying to figure all these things out. Since MapReduce is a framework for parallel computing, we don't have to deal with parallelization, data distribution, fault tolerance or anything of that sort because of an API that is provided. We solved the problem by basically splitting the program in to two “main” functions. Namely, “Mapper” and “Reducer”. Where mapper function reads and parses the data into key value pairs. When thats done, the Reducer is called ONCE and for each unique key that was created by Mapper and given a key and a list of all values that were generated for that key as a parameter. The program takes in a file containing the data and is split. For each m worker, we want m splits. The we fork the processes with a master to pick idle workers. Then the programs maps where each map task reads from the input split. Then partition part, which is responsible for deciding which of the R reduce workers will work on a specific key. Then reduce by shuffling and finally reduce. The output is outputted in a specified output directory.