Erik Swanson
Ezra Bartlett

Design Document
Homework #5

Design Process

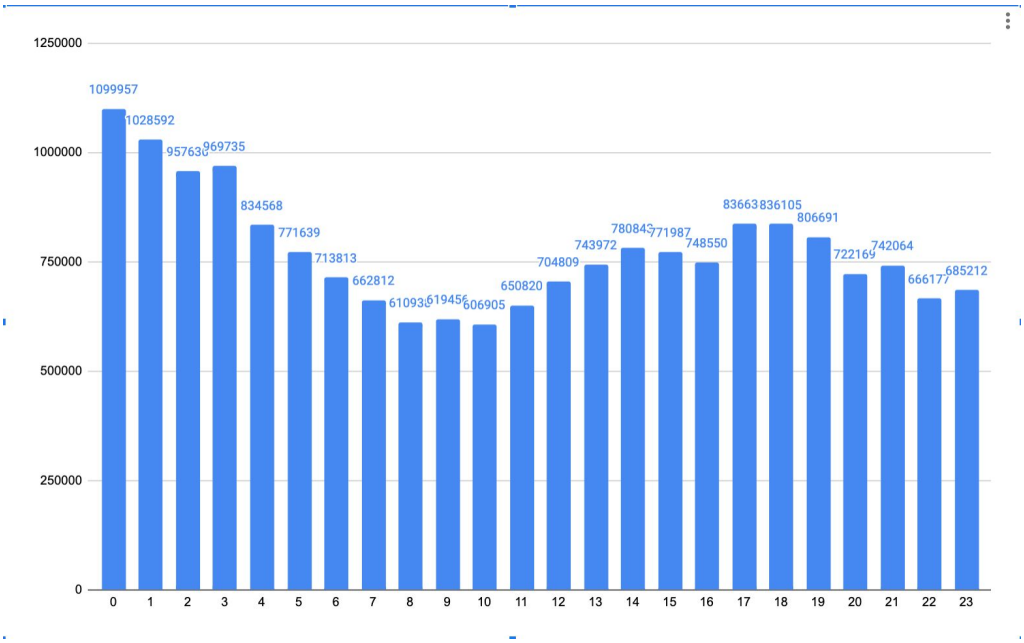
Setup

This whole framework was new to us, and difficult to learn. It took us a long time to get everything working and move to the actual programming assignments. We then had to relearn Java, which neither of us had used in years. Much of this work was done closely following the tutorial found here:
https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

Problem 1

For problem 1, we wrote a mapper class that was pretty straightforward. It checks the first character of the line it receives for a 'T', indicating that the line contains information about the time of the tweet. The mapper then checks characters 13-15 for the hour that the tweet was posted and write it to the output file.

The reducer simply loops through all keys in the output from the mapper and counts the instances. It then writes those to the output file specified when the program was run. The result is a list of hours with the number of tweets found within each hour. You can find a sample in Problem1Output.txt



Output of Problem 1

Problem 2

This problem proved more difficult than problem 1, as it involved writing a custom RecordReader class. We tried several approaches, but our understanding of the MapReduce API ended up not being good enough to come up with a solution. We tried making a custom reader and also using the NLineInputFormat, but were unable to get more than one line at a time to the mapper. If we had been able to, our solution could have checked the whole tweet data unit for the word sleep, then printed the hour to the output file only if the word was found. The reducer could then just count the number of instances of each hour as it did in problem 1.

Running the program

Instructions on running the program can be found in the README.txt file