CSCI-GA.3033-016. Big Data and ML Systems - Fall 2017

Assignment 1a

September 19, 2017

Submission: Please submit in electronic format (LaTeXor MS Word or scanned handwritten solutions). Please include all supporting files along with the solutions in a compressed archive (tar, zip, etc.). Ensure that your responses are adequate and clear.

1. **[No grade]** Attempt the exercises in the NYU HPC Wiki tutorials on Hadoop and Spark.
   a) Link for Hadoop tutorials. Attempt exercises 1 and 3 to familiarize yourself with Map-Reduce methodology and Hadoop library syntax. Also, for exercise 1, look at the source code (WordCount.java, WordCountMapper.java and WordCountReducer.java), and follow Step 4 to compile the program from the sources.
   b) Link for Spark tutorials. Attempt exercises 1 through 6.
2. **[16 points]** Write MapReduce programs (using both Hadoop and Spark) in each of the following.
   a) Compute word frequencies across several text documents. You can download large text documents from here – `https://www.gutenberg.org`. Choose any number you like, a minimum of 10. You can start by modifying the WordCountMapper.java that you worked with in exercise 1 of Hadoop above. (2 points)
   b) Exercise 2.3.1 in the textbook. Given a large file of integers, compute i) largest integer, ii) average of all the integers, iii) set of unique integers, iv) count of distinct integers. (4 points)
   c) Matrix-vector multiplication as described in Section 2.3.1 in the textbook. Test your program on a small matrix of size 10×10 and then scale up to $10^6 \times 10^6$. (4 points)
   d) A modified matrix-vector multiplication as described in Section 2.3.2, in which the vector is assumed to be too large to fit into main memory and hence both the matrix and the vector are divided into equal number of *stripes*. Use the same matrices from the previous problem and repeat. (6 points)
3. **[4 points]** Experiment with varying number of Map tasks and Reduce tasks. The number of Map and Reduce tasks that you choose for your implementation affects the speed. For this exercise, take the simple Word Count application and vary the number of Map and Reduce tasks – Map tasks from 1 to 100 and Reduce tasks from 1 to 100 – and plot a graph of the times taken to execute.