CSCE 4013-002: Assignment 1 (7pt)

Due 11:59pm Friday, September 20, 2019

1 Setting up a stand-alone Spark instance

Download and install a stand-alone Spark instance following the instructions provided in "instructions1.pdf", and instructions from Stanford University provided in "instructions2.pdf".

2 Word Count (2pt)

Write a Spark application which outputs the number of words that start with each word. In your implementation ignore the letter case, i.e., consider all words as lower case. You can ignore all non-alphabetic characters.

Run your program over the input data "pg100.txt".

What to submit:

Submit the printout of the output file and the source code (.java or .py files).

3 Letter Count (5pt)

Write a Spark application which outputs the number of words that start with each letter. This means that for every letter we want to count the total number of (non-unique) words that start with that letter. In your implementation ignore the letter case, i.e., consider all words as lower case. You can ignore all non-alphabetic characters.

Run your program over the input data "pg100.txt".

What to submit:

Submit the printout of the output file and the source code (.java or .py files).

4 1-step Method for Matrix Multiplication (Bonus 3pt)

Consider the following MapReduce algorithm for computing $P = M \cdot N$, where $M = \{m_{i,j}\}$ is a $|I| \times |J|$ matrix, $N = \{n_{j,k}\}$ is a $|J| \times |K|$ matrix, and $P = \{p_{i,k}\}$ is a $|I| \times |K|$ matrix such that

$$p_{i,k} = \sum_{j=1}^{|J|} m_{i,j} \cdot n_{j,k},$$

Map Function:

- For each element $m_{i,j}$, the map function produces key-value pairs $\Big((i,k),(M,j,m_{i,j})\Big)$ for all $k = 1, 2, \cdots, |K|$.
- For each element $n_{j,k}$, the map function produces key-value pairs $\Big((i,k),(N,j,n_{j,k})\Big)$ for all $i = 1, 2, \cdots, |I|$.

Reduce Function:

- The reduce function receives $\Big((i,k), < \cdots, (M,j,m_{i,j}), \cdots, (N,j,n_{j,k}), \cdots > \Big)$ for each key (i,k). For each key (i,k), sort all values in the list by j, compute $m_{i,j} \cdot n_{j,k}$ for each pair of $m_{i,j}$ and $n_{j,k}$ with the same j, then sum up and output as $p_{i,k}$.

The above algorithm requires $|I| \cdot |K|$ reducers. Write a MapReduce algorithm that improves the above algorithm by decreasing the number of reducers. Analyze the communication cost of your algorithm and express it using the Big-O notation with only one parameter a (the number of reducers).

Hint: Use a hash function to hash I-values into b buckets, and use a hash function to hash K-values to c buckets, where $bc = a$.

What to submit:

Include in your report the pseudo-codes of Map Function and Reduce Function, the communication cost, and a short paragraph describing how the communication cost is analyzed.