CSCI4180 (Fall 2024)

Assignment 2: Amazon EC2 and Iterative MapReduce

Due on November 14 (Thur), 2024, 23:59:59

Introduction

In all parts, you only need to configure Hadoop in *pseudo-distributed mode*. For the bonus part, you will need to configure Hadoop in *fully distributed mode*.

Part 1: Configure Hadoop on Amazon EC2 (40%)

In this part, you need to demonstrate the following:

1. Configure Hadoop in pseudo-distributed mode on an Amazon EC2 instance.
2. Run the provided WordCount program on EC2.

Please show to the TAs that you can run Hadoop on Amazon EC2 during demos.

Part 2: PageRank Algorithm (60%)

In this part, you will need to write a MapReduce program to compute the PageRanks of nodes. We still use the Twitter dataset by treating each node in the dataset as a "page" for the PageRank algorithm. The Twitter dataset is obtained from the following reference:

- Haewoon Kwak, Changhyun Lee, Hosung Park, Sue Moon.
  "What is Twitter, a Social Network or a News Medium".
  19th World-Wide Web (WWW) Conference, April 2010.
  URL: http://an.kaist.ac.kr/traces/WWW2010.html

Problem: Given a graph $G = (V, E)$, our goal is to find the PageRank values of all nodes in V. In addition, we include the random jump factor α (which is a command-line parameter) and redistribute all missing PageRank mass m due to dangling nodes. For dangling nodes, we set $p = 0$ during the map phase (i.e., during the distribution of PageRank mass) of the PageRank algorithm, and its PageRank mass is considered to be missed. The missing PageRank mass m (from all dangling nodes) will be added back based on the following equation:

$$p' = \alpha(\frac{1}{|G|}) + (1 - \alpha)(\frac{m}{|G|} + p). \tag{1}$$

Input Format: Each line contains a tuple of (nodeID, nodeID, weight), separated by spaces. Each tuple indicates a directed edge from the former node to the latter node. We ignore the edge weights in this problem, as they are not needed by the PageRank algorithm that we taught in class.

Output Format: Each line contains a tuple of (nodeID, PageRank value), separated by spaces. We only output tuples for nodes whose PageRank values are above certain threshold, where the threshold value (between 0 and 1) is given as a command-line argument.

Sample Command:

```
hadoop jar [.jar file] PageRank [alpha] [iteration] [threshold] [infile] [outdir]
```

Notes:

- Your program (call it *PageRank.java*) will execute the PageRank algorithm over a fixed number of iterations. The program will take a command-line argument `Iterations` to indicate the number of MapReduce iterations needed to be executed. We assume that `Iterations` is at least one.
- You need to write a MapReduce program (call it *PRAdjust.java*) to adjust the PageRank values due to random jumps and dangling nodes after each iteration of the PageRank algorithm.
- We need to implement a class of node structure (call it *PRNodeWritable.java*) and write a MapReduce program to convert the input files into adjacency list format (call it *PRPreProcess.java*).

Bonus (5%)

(a) (2%) Configure Part 2's program to run on Hadoop in fully distributed mode.

(b) (3%) The top 3 groups whose Part 2's program have the smallest running time in fully distributed mode will receive the bonus marks. You may consider to optimize your programs or configure some parameters in Hadoop to make the programs perform better. If more than 3 groups have the best performance, we will still give out the bonus 3% to each group.

Note that the program must return the correct answer in order to be considered for the bonus mark.

Notes

- To simplify our grading, we require that all parts use only a single reducer (by default, the number of reducers is one).

Submission Guidelines

Please at least submit the following files. Additional files are allowed.

Part 2:

- PageRank.java
- PRAdjust.java

- PRNodeWritable.java
- PRPreProcess.java

Declaration form for group projects:

- (`http://www.cuhk.edu.hk/policy/academichonesty/Eng_htm_files_(2013-14)/p10.htm`)

Demo will be arranged on the following day of the deadline. Have fun! :)