Sunil Kumar
94018098

Lab 7 - Report

Page Rank Score Calculation:

Page Rank Initialization: Assign an initial rank of 1/N to all the pages, where N is the total number of pages. This is completed by the first stage of Map-Reduce, that is RankCalculateMapperStage1 and RankCalculateReducerStage1. The Reducer output is of the form : <title> <initial rank>\t<out-links>. It is stored in [output]/temp/iter0.out file.

Page Rank Convergence: The mapper and reducer for this job are RankCalculateMapper and RankCalculateReducer. This operation is performed 8 times, that is 8 iterations of convergence. The values are split into the title, the rank and the out-links. We count the number of outlinks for that page, say K, and calculate the rank-vote of the current page for all its outlinks by the formula , rankVote = rank/K. This value is emitted to all the outlinks on this page. The Reducer adds all the rank-votes from the outlinks and calculate the new page-rank for that page, using the formula :

PR(p) = (1-d)/N + (PR(p1)/L(p1) + PR(p2)/L(p2) +…..), where,
d = damping factor,
PR(p) = page rank of page p,
N= total pages
L(p)=number of out-links on page p.
This page rank value are emitted by the reducer.

Challenges:
1. The rank values after the 8th iteration did not exactly match with the instructor's sample output. But they are only off by less than or equal to 0.01%.

2. Programming in Java language: Understanding the syntax and sequence of operations to be performed was time-consuming.

3. Using HDFS-API instead of the Java files-API was challenging. Also, understanding the Hadoop specific datatypes like Text, was interesting.

Regrade:

I ran the script and got full marks. The file sizes also match. No paths are hard coded and output directories are correctly coded in the source files.