Cloud Programming HW2
Page Rank

Instruction

Build Graph

cd BuildGraph
sh compile.sh
sh execute.sh
cd ../

Calculate PageRank by 10 Iterations

cd PageRank
sh compile.sh
sh execute.sh
cd ../

Sort by PageRank

cd SortPageRank
sh compile.sh
sh execute.sh
cd ../

Calculate Inverted Index Table

cd InvertedIndex
sh compile.sh
sh execute.sh
cd ../

Load PageRank and Inverted Index Table to HBase

sh loadToHbase.sh

Making Queries

cd Query
sh compile.sh
sh execute.sh

Algorithm Design

- BuildGraph
  - Use multiple layers of regular expressions to extract title and its links
  - For title A with links B and C, Mapper outputs key value pairs (B,A) and (C,A)
  - For title A, Mapper outputs an additional key value pair (A,)
  - In Reducer, combine all the titles of a link together, and output the link-title pair if this link contains a white space as title. By doing this, titles that do not exist can be eliminated because they don't contain white space
  - In the second Mapper, read the link-title pair and output the pair reversely as key value pair
  - In Reducer, collect all links of a title and the initial page rank is set to 1

- PageRank
  - An object class Links is defined to contain rank of double type, and an array of link
  - Mapper outputs two kinds of key value pairs. The first key value pair is (title, Links), where Links contains all the links of the title and a rank of -1. The other key value pair is (link, Links), where Links contains empty array of link and a rank calculated as titleRank/linkNum
  - For each title, output the first kind of key value pair. For non-dangling titles, output an additional second kind of key value pair
  - Ranks of dangling titles are added up in reporter by multiplying the ranks by 10^{10}, and the number of titles is also counted in reporter
  - In Reducer for each page, add up all ranks that are not -1 as the new page rank. When encountering rank of -1, collect the links from value to complete the information of a page
  - Get the total number of titles and the total rank of dangling titles from the Mapper's reporter, and calculate the final page rank
  - Running 10 iterations by setting input path as the previous output path

- SortPageRank
  - Mapper output (rank, title) as key value pair
  - KeyComparator sort the pairs by rank in descending order
  - Reducer output the pair in reversely

- Inverted Index Table
  - Reuse HW2 but omit TF.IDF

- Load into HBase
  - Use Pig to load data from hdfs into HBase
  - 100062236_pagerank: {page}\t{rank}
  - 100062236_inverted: {word}\t{page1##page2##…##pageN}

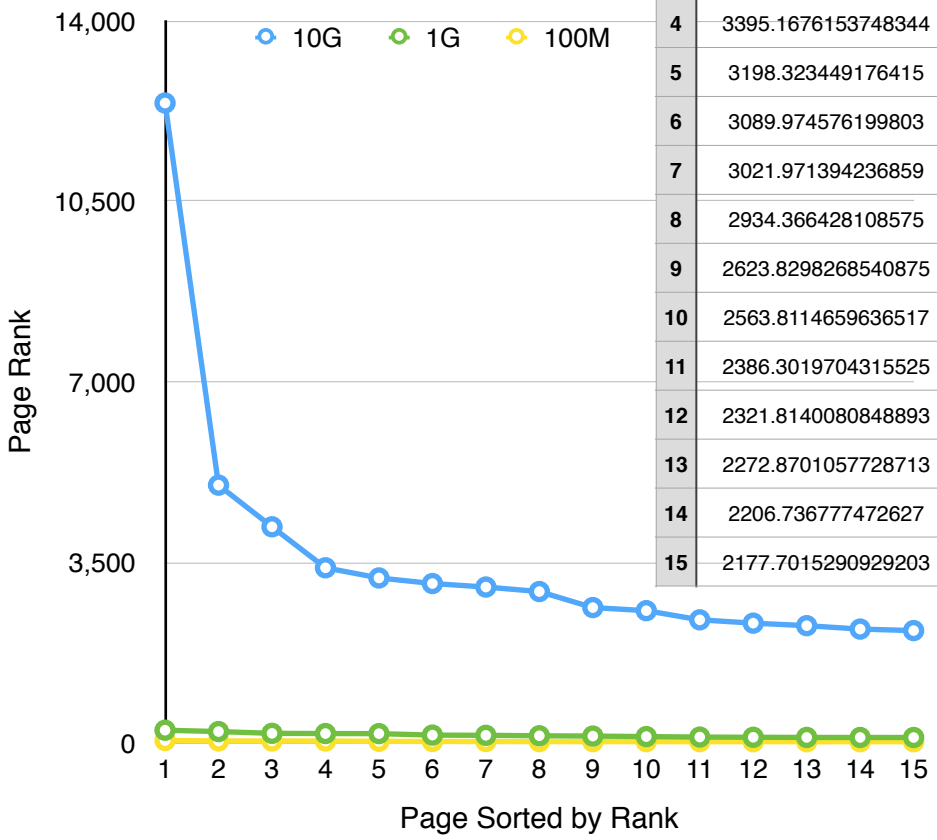
- Query
  - Check if the query string is in inverted index table. If it does, return the value; if not, output "No results found"
  - Split the value string with "##" and extract all the pages in the string. For each extracted page, retrieve its page rank, and output the top-10 highest pages
  - To eliminate the letter case sensitivity, there are four cases considered:
    1. use the original query string to find rank
    2. convert the query string to lower case to find rank
    3. convert the query string to upper case to find rank
    4. convert the first character of query string to upper case to find rank

Experiment

- PageRank Distribution

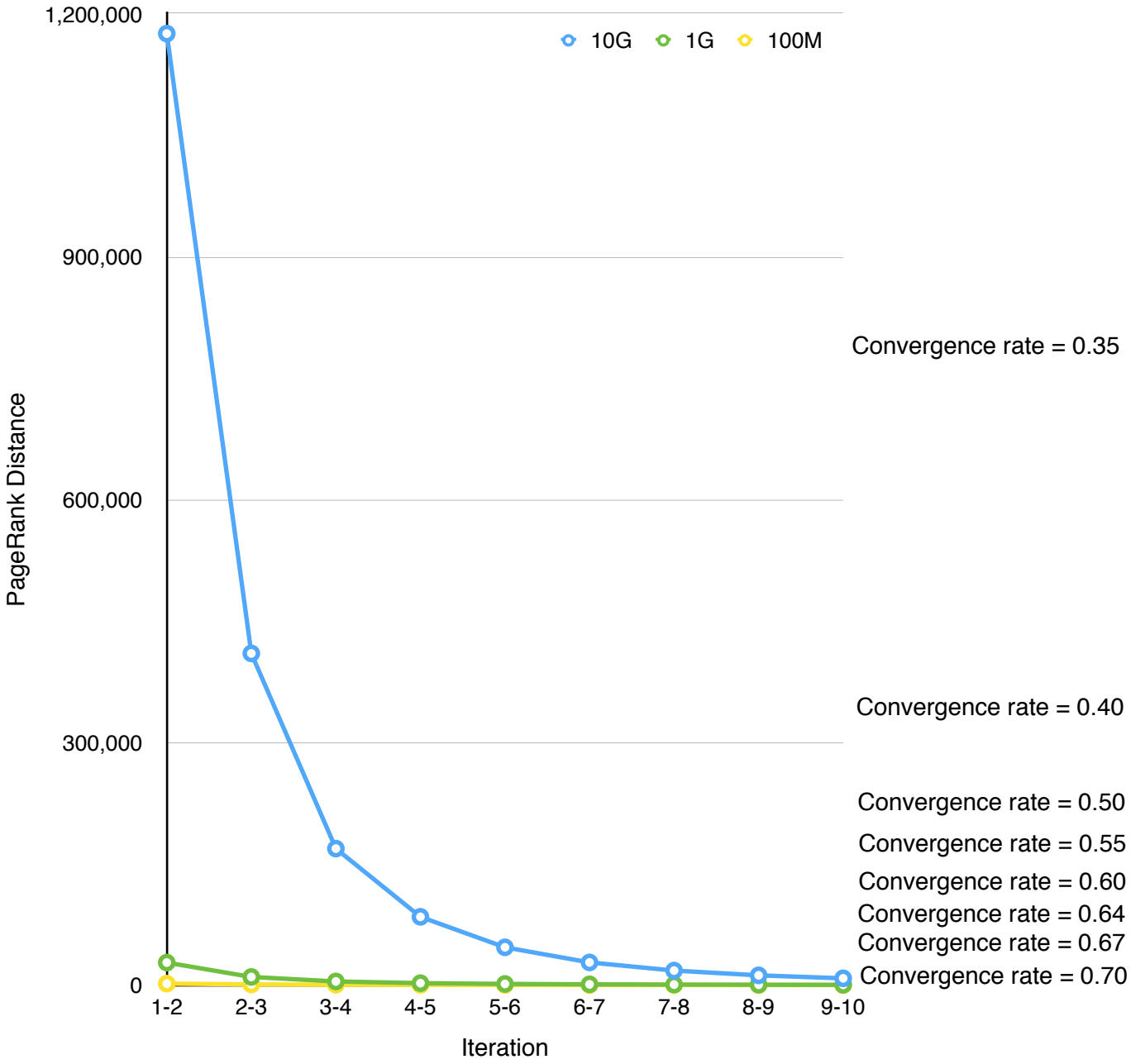
PageRank Distribution

	10G	1G	100M
1	12406.233463158027	246.31989596806926	46.58215505646997
2	4997.803157358753	219.83920880846878	38.15227630346114
3	4190.056284481879	185.48272891434692	37.16976677981223
4	3395.1676153748344	181.42136318344453	36.17073583990466
5	3198.323449176415	179.6925784940482	31.031646976950924
6	3089.974576199803	151.21364986545305	27.747056510703686
7	3021.971394236859	146.83351117932017	26.608120882362872
8	2934.366428108575	138.310076499785	25.243015656396537
9	2623.8298268540875	132.33592117007896	24.322298773954486
10	2563.8114659636517	122.29484584147792	23.713663350125383
11	2386.3019704315525	113.15980015188086	23.669773117528994
12	2321.8140080848893	109.24533500891592	22.766569471616844
13	2272.8701057728713	106.09196271680764	22.546113015030272
14	2206.736777472627	105.26007876988857	22.478799525960355
15	2177.7015290929203	104.5580437327034	22.439957334841747



- When data is large, page ranks of all pages become more distinct. As in 100M data, the highest one is only greater than the second by approximately 8, while in 10G data, the highest one is as twice as the second.

- PageRank Distance



- As shown in the figure, the total page rank distance decreases after each iteration along with increasing convergence rate, meaning that the page rank will eventually approach a steady state where distance will be significantly small to be neglected.

- Calculating page rank is like processing a transition matrix. Since the random jump factor resolves the problem of dangling nodes and self links, page rank is proved to be converging in terms of math.