**Project 2 - Hadoop Page Rank**

Varun Patil

Abderrezak Allalen

Indiana University

**Page Rank Algorithm:**

The web search engine is a typical distributed system on the Internet. It is designed to search for information on the World Wide Web. The search results are generally presented in a list of results and are often called hits. PageRank is a well-known web graph ranking algorithm that helps Internet users sort hits by their importance.

PageRank calculates a numerical value for each element of a hyperlinked set of webpages, which reflects the probability that a random surfer will access that page.

It is given by:

PR(A) = (1-d)/N + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

Where,

|  |  |
| --- | --- |
|  | PR(A) is the PageRank of page A, |
|  | PR(Ti) is the PageRank of pages Ti which link to page A, |
|  | C(Ti) is the number of outbound links on page Ti and |
|  | d is a damping factor which can be set between 0 and 1.  N refers to the total number of unique urls.  The PageRank theory holds that even an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability at any step, that the person will continue is a damping factor d. Various studies have tested different damping factors, but it is generally assumed that the damping factor will be around 0.85.  **Page Rank Algorithm using Hadoop:**  Due to large processing of input data, like web graphs containing more than a million webpages, we need to run the PageRank application in parallel so that it can aggregate the computing power of multiple compute nodes. By using the Map Reduce framework of Hadoop we can implement parallel version of PageRank Algorithm. |

**Figure 1:** Hadoop PageRank Dataflow (from: https://cloudmooc.appspot.com/homework?unit=4)

**PageRank MapReduce Program:**

Initially, the PageRank input data is stored in the format of adjacency matrix as a file(s) in the local file system. Then it is uploaded to the HDFS and distributed across the compute nodes. Hadoop framework reads the application records from HDFS with the InputFormat interface and generates <key, value> pair input streams. Each Map function produces zero or more intermediate (key, value) pairs by consuming one input (key, value) pair. Each reduce task aggregates all the partial values of specific webpages.

**Map Function:**

If the “targetUrlsList” is less or equal to 0, this indicates that there is no out degree for the url. Therefore its rank value is scattered to all other urls and for each url in “numUrls” write out the url and “rankValuePerUrl” pair.

On the other hand if the “targetUrlsList” greater than 0, this indicates there is out degree for the url, therefore its rank value is the rankvalue divided by the number of target urls. Then iterator is created for the ‘targetUrlsList”. While the targetUrlsList has next url emit nextUrl , rankValuePerTargetUrl> pair and finally emit the map output that consists from < sourceUrl, nextUrl> pairs (See while loop in PageRankMap.java)

**Reduce Function:**

If strArray contains only one url this means there are no out urls. Therefore rank this url using this code ‘sumOfRankValues += Double.parseDouble(strArray[0]);”. strArray has more than 1 url then for each url in strArray compute targetUrlsList using (targetUrlsList += "#" + strArray[i];). Finaly the rank values are summed using(sumOfRankValues = 0.85\*sumOfRankValues+0.15\*(1.0)/(double)numUrls;) and final output is emitted as <key, value> pair using (context.write(key, new Text(sumOfRankValues+targetUrlsList));). The value consists of sumOfRankValues+targetUrlsList.

**Output (See aallalen\_HadoopPageRank\_output.txt ):**

The output is generated after 3 iterations. The top 10 ranks are displayed using the following script

cat output/\* | sort -n -k2 -r | head -n10

########################################################

# Hadoop PageRank Job take 489.521 sec.

########################################################

Warning: $HADOOP\_HOME is deprecated.

PageRank Finished execution, see output in output/.

summer@ubuntu:/root/MoocHomeworks/HadoopPageRank$ cat output/\* | sort -n -k2 -r | head -n10

cat: output/\_logs: Is a directory

236 9.966958965951023E-4

2180 9.922394955092162E-5

1014 9.909665728044204E-5

2606 9.907236642218223E-5

1032 9.845535372997214E-5

3340 9.817576110563587E-5

176 9.792325060092003E-4

2158 9.774443867093785E-5

1682 9.766237020918693E-5

4748 9.750330711822623E-5

summer@ubuntu:/root/MoocHomeworks/HadoopPageRank$