

**Project Report – Hadoop Page Rank**

**Indiana University**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Description | Author |
| 1.0 | 11-Feb-2017 | Project Description and Main Steps | Melita Dsouza |
| 1.1 | 12-Feb-2017 | Data Flow | Karthik Vegi |
|  |  |  |  |

**PROJECT DESCRIPTION**

PageRank algorithm is used by Google search engine to measure the relative ranking of the zillions of web pages. It assigns a numerical weighting to each of the pages in the set.  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.

**1. MAIN STEPS**

The Hadoop program has 6 important classes:

**HadoopPageRank:** The driver program that starts the code and starts all the jobs

**PageRankMap**: Reads the key value pairs and calculates the page rank

**PageRankReduce**: Aggregates the links for the same key

**SortMap**/**SortReduce**: This will sort the links based on the page rank and list the top ten pages with highest page rank

**2. DATA FLOW**

**2.1) Input and Output**

**Input**: The input to the program is a text file with information about 5000 pages ie, the list of the pages to which they are linked.

**Output**: The output is the list of the pages along with a numerical page rank

**Build File**

We write the program, compile it using the **compileAndExecHadoopPageRank**.**sh** file which does the following:

* Checks if the arguments are passed properly
* Cleans and builds the code
* Starts the Hadoop daemon nodes
* Creates an input folder on hdfs and transfers the input
* Runs the program on hdfs
* Transfers the output to the output folder

**2.2) Reading Input**: The program has 4 arguments:

* input file path
* output directory
* number of urls
* number of iterations
* Each url is provided with a list of other urls to which it is connected
* If the node doesn’t have an outbound value, we treat that url as a dangling node and work on it differently in the program

**2.3) Converting Adjacency map to key value pairs**

* A new job is created which will run the class HadoopGraphMap
* **HadoopGraphMap** will read the input file in the adjacency matrix format and create key value pairs
* We manually set the number of reduce tasks to 1
* Users can define number of reduce tasks. If the value is low, all reducers can be launched immediately. If there is only one reducer, it must loop multiple times to collect results from map tasks. If it is high, then better load balancing is done

**2.4) Calculating the page rank**

* We read the key value pairs and start assigning page rank.
* We first calculate intermediate page rank value for each URL and distribute average value of dangling URLs to each web page.
* We multiply this summation by damping factor d to reduce the pagerank of predecessor.
* A page loses its page rank by some extent when it has an outbound link to another page.
* The formula we use for page rank is given below

**2.5) Iterations and scope of improvement**

We give the input of how many iterations we require to do to get final ranks for each page. However, we can make it more efficient by setting a threshold value and checking after every iteration. This way convergence will occur even before all the iterations are complete and we can save on a lot of computation time.

**2.6) Data Flow**

<0, . #>

<1, . #2>

<2, . #1>

<3, . #0#1>

<0, . #>

<1, . #2>

<2, . #1>

<3, . #0#1>

0

1 2

2 1

3 0 1

mapper

Page rank

reducer

cleanup

<0, 1/16>, <1, 1/16>

<2, 1/16>, <3, 1/16>, <0, “”>

<2, .>,<1, #2>

<1, .>, <2, #1>

<0, 1/8>, <1, 1/8>, <3, #0#1>

<0, 3/16>

<1, 7/16>

<2, 5/16>

<3, 1/16>

<1, . #2>

<2, . #1>

<3, . #0#1>

<0, . #>

<0, 3/16>

<1, 7/16 #2>

<2, 5/16 #1>

<3, 1/16 #0#1>

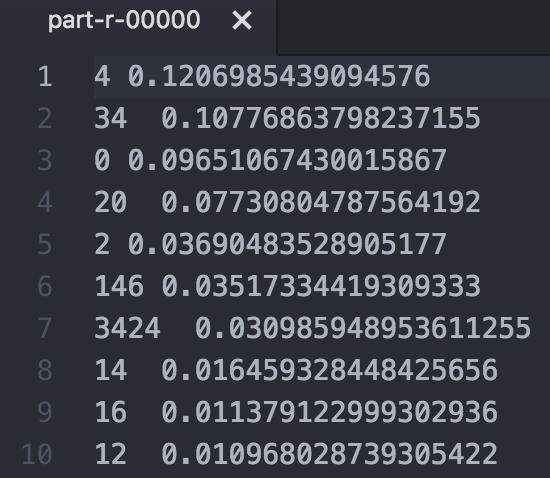
<1, . #2>

<2, . #1>

<3, . #0#1>

**2.7) Output**

* We ran for 10 iterations and the output is given below.
* The output is written to HDFS output folder on the file **part-r-00000** and it is transferred to the output directory of the code base. We take the first ten from the sorted list.



**REFERENCES**

* **Hadoop Docs**: <https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>
* **Hadoop API**: <https://hadoop.apache.org/docs/r1.2.1/api/org/apache/hadoop/mapred/Mapper.html>
* **Wiki:** https://en.wikipedia.org/wiki/PageRank
* Hadoop Definitive Guide, Tom White