**Introduction:**

Our aim in this project is to obtain the top rated pages from the list of pages in the given set of Wikipedia Articles. In order to implement, the **PAGERANK** Algorithm is used.

PageRank is a significant part of search engine optimization and is a link analysis algorithm applied by GOOGLE. It assigns a rank to each accessed webpage.

Evaluating PageRank involves evaluating all the links to the specific page. If a page is called or accessed by a lot of links then it will be ranked high among all searched pages.

PageRank can be calculated using the formula:



Where

* : is a constant (it is typically set to 0.85)
* N: is total number of pages ( In this case Wikipedia articles)
* : The set of all incoming links to
* : The number of outgoing links form

Initially, each page is initialized with PageRank 1/N and the sum of the PageRank is 1.

**Methodology:**

**Map reduce:**

To extract the wiki links from the given dataset, map reduce is used.

* As a first step in the Project, the wiki links are to be extracted from the xml file that contains the data records. And the map reduce alone can’t parse the xml files as it not available with inbuilt input format for xml, so Apache Mahout Libraries are used for parsing the xml files.
* In order to generate the mapper output of the format (pagetitle, pagelink), I have used java regular expressions

Pattern pat = Pattern.compile("\\[\\[([^\\[\\]|]\*)[^\\[\\]]\*\\]\\]");

* In the map reduce only mapper function is used to generate the required dataset containing the pagetitles and pagelinks.

**Pig:**

Once the output file is obtained

1. The red links have to be removed from the dataset. The red links are ones that doesn’t point to any links in the dataset. Pig programming has used to remove red links by using self-join.

Example: page\_join= join page by page\_link, new\_page by page\_title1;

1. The output of red links will be in the form of (pagetitle , pagelink). Now we need to calculate the total no of links(N).

For Example :



As the outgoing links are from A,B,C the total count is 3.

**Steps to perform 1st iteration:**

OUTgoingLink= foreach page\_grp generate group, COUNT($1);

dump OUTgoingLink;

(a,2)

(b,1) { Outgoing link for every page }

(c,1)

in\_out\_links = join OUTgoingLink by $0, page\_distinct by $0;

(a,2,a,b)

(a,2,a,c) {for each outgoing links finding the incoming links}

(b,1,b,c)

(c,1,c,a)

p = foreach in\_out\_links generate $2 as pi, $3 as pj, $1 as No\_outgoing, 1 as no;

pi -> incoming link

pj -> outgoing link

C= CROSS page\_count, p;

generate\_pages = foreach C generate $2 as pi, $3 as pj, $4 as No\_outgoing, $1 as Total\_nodes, $5/$1 as pr;

(a,b,2,3.0,0.3333333333333333)

(a,c,2,3.0,0.3333333333333333)

(b,c,1,3.0,0.3333333333333333)

(c,a,1,3.0,0.3333333333333333)

generate\_pages\_pr = foreach generate\_pages generate pi, pj, No\_outgoing, Total\_nodes, (1-0.85)/Total\_nodes as first, (double)pr/No\_outgoing as PR;

D= group generate\_pages\_pr by pj;

(a,{(c,a,3.0,0.05000000000000001,0.3333333333333333)})

(b,{(a,b,3.0,0.05000000000000001,0.16666666666666666)})

(c,{(a,c,3.0,0.05000000000000001,0.16666666666666666),(b,c,3.0,0.05000000000000001,0.3333333333333333)})

Sum\_of\_D = foreach D generate group, $1 as data, 0.85\*SUM($1.$5) as Total\_PR;

E = foreach Sum\_of\_D generate $0 as pj, FLATTEN($1), Total\_PR;

F = foreach E generate pj, $1 as pi, $3 as No\_outgoing, $5 as firstpart\_alg, (double)$5+Total\_PR as Page1\_Rank;

page2 = foreach F generate pi , pj, No\_outgoing,firstpart\_alg,Page1\_Rank;

(c,a,1,0.05000000000000001,0.3333333333333333)

(a,b,2,0.05000000000000001,0.19166666666666668)

(a,c,2,0.05000000000000001,0.475)

(b,c,1,0.05000000000000001,0.475)

* In the same way I need to run the pig script for 9 iterations
* Then by using the below script I have retrieved the page rank of top 100 universities.

extract\_university = FILTER Iteration\_10 by (pi matches 'university of.\*') OR (pi matches '.\*university');

Order\_university = order extract\_university by $4 DESC;

Top\_100 = LIMIT Order\_university 100;

**Results and Conclusion:**

For the Wikipedia Dataset the PageRank has been for all pages, then we have selected the pages whose title starts with “university of” or ends with “university” and Produced the output, the list of top 10 universities with the highest page rank.