**Page Rank Algorithm using Hadoop**

**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. |

Hadoop PageRank Dataflow

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.

**Hadoop – CreateGraph:**

Map function-

<key, value> pair is generated. It corresponds to <sourceUrl, Init.PageRank#targetUrls>.

Initial page rank is calculated as:

Init.PageRank=1.0/numUrls

The file generated is stored locally.

Reduce function-

Flushes the < key, value> pair to HDFS. It forwards the file generated in the Map function to Hadoop DFS.

**Hadoop – PageRank:**

Map function-

<key, value> pair is generated as before. It corresponds to <targetUrl, rankValuePerTargetUrl> or <sourceUrl, #targetUrls>.

Here, the Dangling urls pagerank value is scattered across all other urls. The pagerank of Normal urls is divided equally to its outbound urls.

Reduce function-

<key, value> pair generated corresponds to <url, sumofPageRankValues#targetUrls>. Here, we combine the key page ranks(sum it up) and append the target Urls together.

The file generated is in HDFS at both the steps.

**Hadoop – CleanupResult:**

Map function-

<key, value> pair generated corresponds to <url,pagerank>. The target urls are removed.

Reduce function-

<key, value> pair generated corresponds to <pagerank,url>. It just reverses the pair.

The file generated is in HDFS at both the steps.