Distributed Systems Assignment 1: PageRank

-Rohit Nair (Ronair), Abhijit Karanjkar(Aykaranj)

Google’s PageRank algorithm has several features which it considers to rank pages in the order of importance. We’ve used eqn. 2 of the algorithm, which considers the damping factor (generally taken as 0.85, but we’re accepting it as an input as requested)

Eqn.2:

Dangling Nodes: There’re 2 ways of handling these nodes. Either assign all nodes as its out links or assign all nodes but itself as its out links, we’ve used the former approach. Though testing suggested that it didn’t make a big difference (at least for an input size of 1000 pages).

Method-wise description:

parseArgs:

assign the command line values to the global variables.

loadInput:

read from the input file. Each line consists of node followed by urls that it points to. We save it in the adjacency list. Dangling nodes are assigned out links to all urls in the system.

calculatePageRank:

PageRank algorithm is memoryless. We calculate next page ranks for each iteration from the current values of page ranks and the knowledge of the system from the adjacency list. For each node, we calculate its contribution to each of the nodes it points to and sum all of them. This is very efficient especially in a distributed environment. In the end, we make adjustments in accordance with the damping factor for each page’s rank.

rankSum:

We can pass the rank list to this function to check whether the sum of ranks of all pages is 1, this is a good consistency check for the system

hasConverged:

We calculate the Euclidean distance between the current rank values and the next and stop further iterations if the distance is below a threshold.

printValues:

We write the number of iterations and page name and rank of top 10 pages in the output file. We display the time taken by the algorithm on the console