Havish Chennamraj
903201642
Porgramming Assignment 2 Analysis

How scalable is my map reduce program?

The first thing to consider is that I've ran my Map-Reduce program on my local host, and not on a cluster. This obviously inflicts restrictions on the number of mappers/reducers that I can parallely run. Also considering that I've ran my map reduce program on a VM to which I've allocated only 4 cores (the best I could do because of some weird issue) further tightens the restrictions. Having said that, the Map-Reduce example that I've ran which basically entails calculating the max edge weight for every node in dataset of ~4.84 million nodes and ~69 million edges and ~1.2 GB ran successfully in reasonable time (under 5 mins). Currently my Web Link graph has around 2700 nodes and 4500 edges, but if I have to scale my Web Link graph to the same order as that of my Map-Reduce example, it should be able to run fine considering both time and space constraints of my VM. However, entering into the realms of 100 Gbs worth of data or more is when some kind of horizontally scalable infrastructure would be absolutely needed to run the Map-Reduce Job for it to complete within a reasonable time limit and also save the input, intermediate and output files.

Input and Output Graphs

Both Input and Output graphs are created from the directed edge files. The input edge file is obtained my scraping my CS 7646: Machine Learning for Trading & output edge file is produced by the Map-Reduce job on the input edge file. In both the graphs, only nodes with degree greater than 115 have been labelled not to clutter the graph. Also please note that distance between two nodes in either of graphs signifies nothing and the appearance of the graphs is basically a result of different visualization layouts in Gephi.