**INTRODUCTION**

The ACM Citation Network database is accessed and analyzed by the company. Our record contains indexes and references for the papers displayed in the ACM Library as the name suggests. Initially, the ACM citation file is reduced to two columns representing the main paper in the first column and cited in the second column. Second task enables the graph to be visualized with the ACM Citation network's in-degree distribution. The third task is to implement Weighted Page-Rank Algorithm, based on the data generated in the previous step. The goal is to return the top 10 papers most cited in the dataset. The final step of our project is to calculate the average clustering coefficient .

**Methodology**

The source data for our project was the ACM Citation Network file called ACM-Citation-Network-v8.txt found. (<https://aminer.org/billboard/citation>)

**Step 1:** **Building the Citation network using MapReduce**

* Configure MapReduce Driver class to a customizable record delimiter.

Configuration conf = getConf();

conf.set(“textinputformat.record.delimeter”,”#\*”);

Job job = new Job(getConf());

* It enables us to create input split between “#\*”s.
* We create two regular expressions to filter indexes and citations.
* After successful match emit index as key and citations as value.
* Filtering and shuffling will be done in Spark .
* Only returns those papers which have citation.

**Step 2: Visualizing In-Degree Distribution**

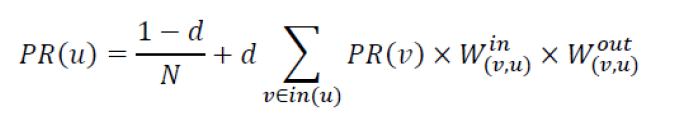
In our second step, our main goal is to produce a graph of in-degree distribution from the first step results. The number of other papers represented by in-degree for our paper that cited our paper.

* For our data, the in-degree for a particular paper would be the number of papers that have cited it .
* The in-degree distribution is number of papers with same in-degree divided by total number of papers.
* We use the file generated in the previous step to generate the graph using python.
* Calculate the in-degree of each paper. First, we initialize each indegree with 1 and sum all the same indegree to calculate indegree distribution.

**Step 3: Implementing Weighted Page Rank Algorithm**

Weighted Page-Rank Algorithm Implementation returns the top 10 papers in our dataset with the highest page rank. Every node is assigned weight to show which node is relevant by weight.

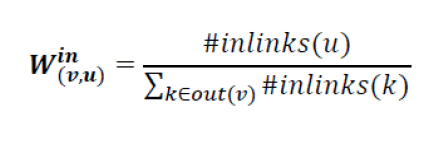
The Weighted page rank:



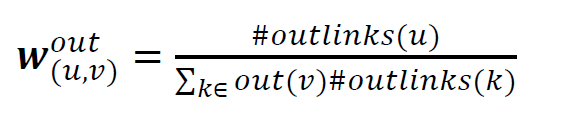
N=Total number of nodes

In(u) is indegree calculated for node u.

Win is an in-weight of link(v, u). It is calculated as the ratio of the number of in-links (incoming links) to node u over the number of in-links to all references of node v .



Wout is out-weight of link(v, u) . It is calculated as the ratio of the number of out-links (outgoing links) from node u and the number of out-links from all references of node v :



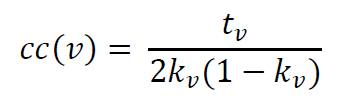
**Step 4:** **Finding Average Clustering Coefficient and Average Path Length of the network and compare it to a random graph**

Many real-world networks including social networks of a type of graph called Small-World networks. Small-world networks have two main properties:

1. Many nodes in the network have a small degree that means many nodes are not each other's neighbours, but their neighbors are also each other's neighbors for each given node.

2. Despite the small degree and dense clustering of most nodes in the graph, any node in the network can be reached relatively quickly from any other network by crossing a small number of edges determined by the network's Average Shortest Path Distance.

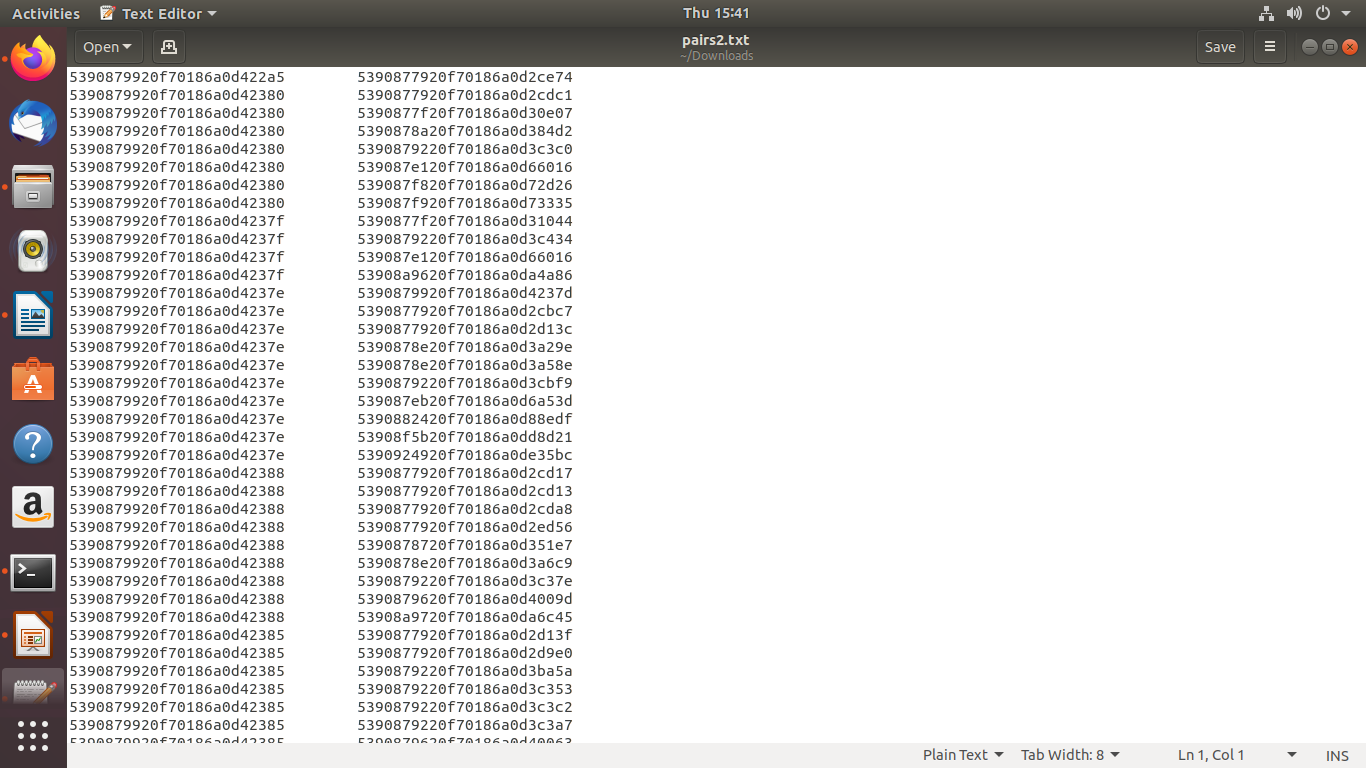
The local clustering coefficient of a node v in a graph is defined as follows:

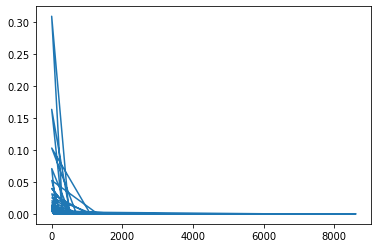


* Used GraphX API to calculate triangle count and degree for each vertex in graph.
* Further input the calculated values in the formula to calculate coefficients for each vertex.
* After, we calculate the average of the coefficients.

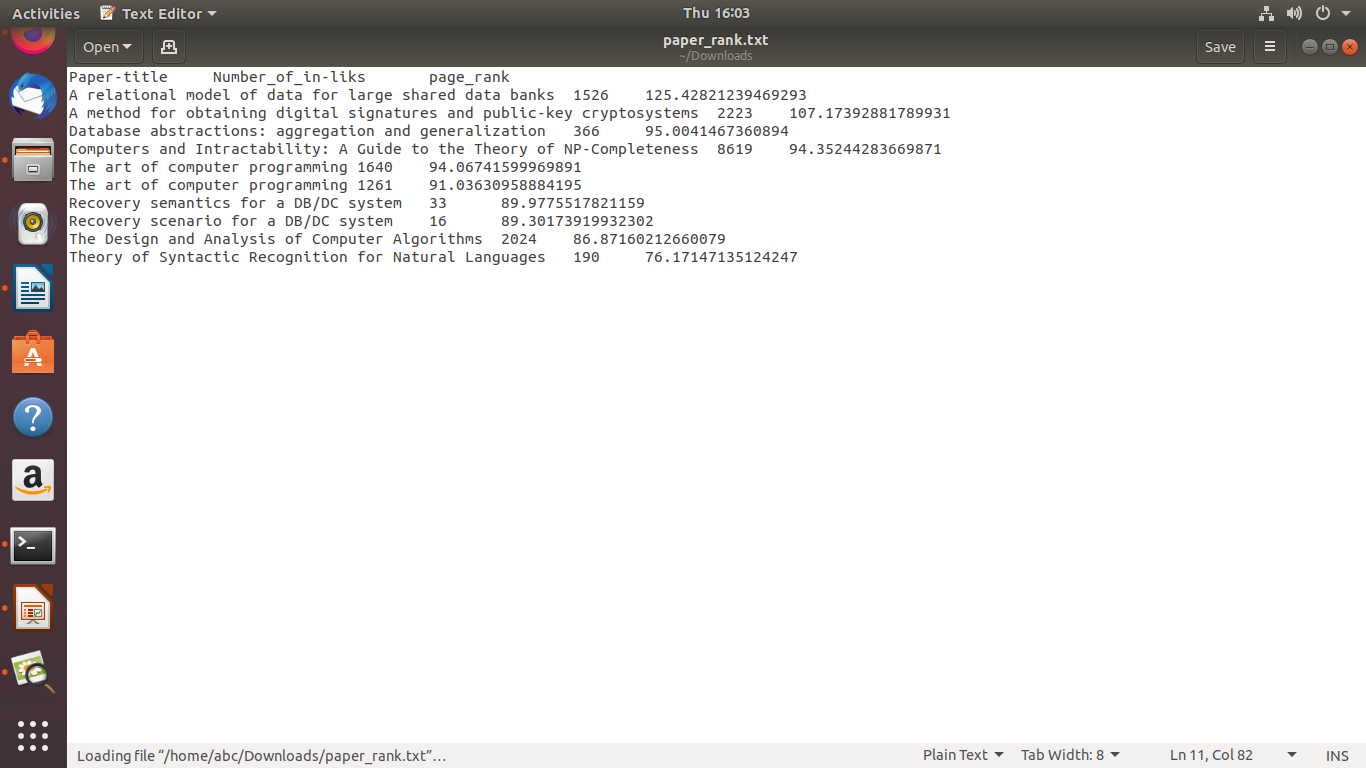
**Results**

**Step 1: Index and paper cited are separated by tab**

****

**Step2: Visualizing the in-degree distribution of graph.**

**Step 3: The Weighted Page Rank algorithm provides with the top ten most influential papers.**



**Step 4:**

Comparing coefficients and shortest path for random graphs and graph generated in the previous steps

Avg Clustering coefficient:

0.0017568950402310996

**Conclusion**

1. Spark in spylon-kernel used to filter index and citations.
2. Using Python, plotted the graph for indegree distribution.
3. The Weighted Page Rank algorithm provides with the top ten most influential papers.
4. In degree distribution graph and random graph, the clustering coefficient and shortest path allows the ACM citation graph to show small worldwide network properties.