Proposal for Comparison of Graph Centrality Algorithms in GraphX

J.T. Liso, Sean Whalen

***Abstract*— GraphX is a popular graph processing system developed at UC Berkeley in 2014. Although the system is widely used, it has received some criticism for its performance. In their paper, GraphX is described to be very powerful in use of PageRank, but this is only one of many graph centrality algorithms. For this project, we propose a comparison of performance of other graph centrality algorithms used in research in comparison to the PageRank algorithm that GraphX was optimized for. We hope to find that GraphX has proportional performance in other graph centrality algorithms as it does in PageRank. Centrality algorithms with varying time complexities will be used to test this. Additionally, any potential functionality problems will become apparent through the testing of varying algorithms.**

I. MOTIVATION

Graph processing has become an integral part of data analysis, especially in the analyses of social networks. GraphX [1] was proposed in 2014 as a graph processing solution built on top of Apache Spark. However, their analyses of the system were limited to only the graph algorithms of PageRank and connected components. PageRank is a graph centrality algorithm focused on finding the most used website in a search result. Although this algorithm is widely used (especially by Google), many other graph centrality algorithms exist and are necessary for other applications. We want to analyze the GraphX system through a variety of graph centrality algorithms of different time complexities and compare their performance of both large and small graphs to those of PageRank.

II. GRAPH CENTRALITY ALGORITHMS

Graph centrality is a process of finding the most important vertex or group of vertices in a graph. Different algorithms have different definitions of what the most important vertices are in a graph, and consequently have varying purposes. We aim to compare the PageRank algorithm used in the GraphX paper to three other graph centrality algorithms, namely, Degree Centrality, Closeness Centrality, and Betweenness Centrality. We will only consider simple, undirected finite graphs, with *V* vertices and *E* edges.

A. PageRank

PageRank [2] is an algorithm developed by the Google founders originally used to rank websites, but can be used to determine the ranking of vertices in a graph by importance. Simply speaking, it uses probability distributions to represent likelihood of a certain vertex (website) being visited. The PageRank, *PR(u)*, for a vertex *u* can be found by

$$PR(u) = \sum_{v \epsilon B_u} \frac{PR(v)}{L(v)}, \tag{1}$$

where B_u is the set of vertices connected to *u* and *L(v)* is the number of edges from vertex *v*. Since PageRank is a probability distribution, values range between 0 and 1. PageRank typically runs in *O(V + E)*.

B. Degree Centrality

Degree centrality is probably the most conceptually simple graph centrality measure. The degree centrality of a vertex *v* is found by finding the degree of the vertex, meaning the number of edges connected to *v*. The vertex with the highest centrality, i.e. degree, is considered to be the center of the graph. This algorithm runs in $O(V^2)$ for dense graphs and $O(E)$ for sparse graphs [3].

C. Closeness Centrality

Closeness centrality measures how close all other vertices are to a vertex. Closeness centrality is the inverse of the farness measure [4]. Consequently, closeness can be defined by the equation

$$C(v) = \frac{1}{\sum_u d(v,u)}, \tag{2}$$

where $d(v,u)$ is the distance from vertex *v* to vertex *u*. Large closeness centrality indicates vertices are closer than a smaller closeness centrality. Closeness runs in *O(E)* time.

D. Betweenness Centrality

Betweenness centrality calculates the number of times a vertex serves as a point along the shortest path between two vertices. It can be described by the equation

$$C_B(v) = \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{3}$$

where σ_{st} is the total number of shortest paths from vertex *s* to vertex *t* and $\sigma_{st}(v)$ is the number of shortest paths from vertex *s* to vertex *t* that pass through vertex *v* [5]. Betweenness centrality typically takes $O(V^3)$ time using Floyd-Warshall algorithm [6].

III. DATA

We intend to use a variety of datasets in our analyses to extensively test the robustness of GraphX on varying data sizes and data types. We plan to use not only social media data, but also some biological data that we have obtained from a prior project. The only data analyses done on GraphX in the paper was using social network and census data, so we hope to see no change in performance of different types of datasets. As stated before, all graphs will be simple, finite, and undirected.

IV. EXPECTED OUTCOMES

From this project, we hope to completely understand the robustness of GraphX under a variety of graph centrality algorithms. We hope to see that the type of algorithm used has reasonable bearing on the performance of GraphX in identifying a graph's center vertices or components in comparison to the time complexity of the varying algorithms. Since the algorithms run in different times, we should see proportional changes in the computational speed of GraphX. This means that as the time complexity increases between algorithms, the runtime in GraphX should increase by a proportional factor. We will time each of the algorithms on a variety of datasets consisting of different number of vertices and edges. We expect to see that the runtime of each algorithm is consistent with the Big-O of the algorithm. For example, we should see a quadratic relationship between the number of vertices and the runtime for dense graphs using degree centrality.

Additionally, we hope that the robustness and fault tolerance of GraphX will prevent any crashes in some of these more computationally intense and longer algorithms such as betweenness centrality. The robustness will just be a measure of how large of graphs GraphX can handle for these different algorithms. We intend to measure this through finding the maximum size graph that will avoid a crash in the system. In case of a crash, we expect that GraphX should be able handle the fault tolerance for very large graphs as it would PageRank. From these results, we will be able to better understand how GraphX can handle different applications as well as finding potential performance or functionality problems if they exist.

REFERENCES

[1] Joseph E. Gonzalez, Reynold S. Xin, Ankur Dave, Daniel Crankshaw, Michael J. Franklin, and Ion Stoica. 2014. GraphX: graph processing in a distributed dataflow framework. In Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation (OSDI'14). USENIX Association, Berkeley, CA, USA, 599-613.

[2] Page, L., Brin, S., Montwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999

[3] Freeman, Linton C. "Centrality in social networks conceptual clarification." Social networks 1.3 (1979): 215239.

[4] Sabidussi, G (1966). "The centrality index of a graph". Psychometrika. 31: 581603. doi:10.1007/bf02289527.

[5] Brandes, Ulrik (2001). "A faster algorithm for betweenness centrality" (PDF). Journal of Mathematical Sociology. 25: 163177. doi:10.1080/0022250x.2001.9990249. Retrieved October 11, 2011.

[6] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L. (1990). Introduction to Algorithms (1st ed.). MIT Press and McGraw-Hill. ISBN 0-262-03141-8. See in particular Section 26.2, "The FloydWarshall algorithm", pp. 558565 and Section 26.4, "A general framework for solving path problems in directed graphs", pp. 570576.