Electrical and Computer Engineering / Software Engineering

CPrE/SE 419: SOFTWARE TOOLS FOR LARGE-SCALE DATA ANALYSIS, SPRING 2019

Purpose

In this lab, you will use Hadoop MapReduce for analyzing large graphs. A graph (sometimes called a network) is a fundamental structure used for modeling relationships between entities, for example, hyperlinks between webpages, or friendship between people in a social network. The lab extends the algorithms for graph processing covered in class. At the end of this lab, you will know how to:

- Process a large graph that is presented as a set of edges
- Compute local properties of graphs, such as edge neighborhoods and triangles

Submission

Create a zip (or tar) archive with the following and upload it on the Canvas.

- A write-up answering questions for each experiment in the lab. For output, take a snap shot of results from your terminal and summarize when necessary.
- Analysis of the communication complexity of your program, and provide your derivation of your analysis.
- Commented Code for your program. Include all source files needed for compilation.

Experiments

Our dataset is the U.S. patent citation data, which is maintained by the National Bureau of Economic Research. In the graph that is considered, the vertex set is the set of all patents issued between 1975 and 1999, for a total of nearly 4 million patents. For each citation, say, from patent A to patent B, there is an edge from vertex representing A to the vertex representing B, in the citation graph. Hence this citation graph is a directed graph.

The above graph has been uploaded to HDFS at the server at the location:

"***/cpre419/patents.txt***". More information about the data can be obtained from its source: http://snap.stanford.edu/data/cit-Patents.html Feel free to check out what these patents do at http://patft.uspto.gov/netahtml/PTO/srchnum.htm

The graph is in the form of an edge list. Every line of the file has information about a single edge. A line contains information in the format <from vertex> <to vertex>, which means that patent <from vertex> has a citation to patent <to vertex>.

Experiment 1 (50 pts)

The first task is to find significant patents, defined as follows. We say that there is a one-hop citation from patent X to patent Y if X cites Y directly, and we say that there is a two-hop citation from X to Y if there is a patent Z such that X cites a patent Z and Z cites Y. For the purpose of this experiment, we define the **significance of a patent X as the number of distinct patents Y such that there is either a one-hop citation or a two-hop citation from Y to X**. It is possible that a patent X has a direct citation to X itself. There might also be a two-hop citation from X to X. In your code ignore such self-citations, either 1-hop or 2-hop.

Your task is the following: *Write a MapReduce program to extract the ten patents with the largest significance.* If there is a tie in choosing the winners, then they can be broken arbitrarily.

You should output the top ten patents and their significance in the directory

"/user/<User ID>/lab3/exp1/output"

Hint: Good convention for temp files is to put them in a /user/<User ID>/lab3/temp. Make sure that both the output location and your temp directory are empty **before** running your job in the cluster. You can delete them in your job implementation before running a new job.

In order to avoid cluster issues and offload cluster nodes, you should kill your previous jobs if they are running for a long time. You can find job number via

http://hpc-class.its.iastate.edu:8088/cluster/apps/RUNNING

The command to kill a job is the following: hadoop job –kill job_number_extension

Experiment 2 (50 pts)

For the next experiment consider the same patent graph as the input, but convert it into an undirected graph by ignoring the direction on an edge. Thus each edge in the input file represents an undirected edge between the two vertices.

A triangle is a set of three vertices such that all three pairs of vertices are connected to each other. For example a triplet of vertices {4, 7, 9} form a triangle in a graph if and only if the graph has the following edges: {4, 7}, {4, 9}, {7, 9}. The number of triangles in a graph is an important metric of a graph that has applications in several domains including social network analysis. Note that a single vertex can participate in multiple triangles in the graph.

The global clustering coefficient (GCC), which is a measure of "connectedness" in the network, is based on the numbers of different types of triplets of nodes. A triplet of nodes is called an open triplet if the three nodes are connected with two links. A triplet is called "connected" if it is either open or is a triangle. The global clustering coefficient is calculated as:

$$GCC = \frac{3 * Number\ of\ triangles}{Number\ of\ connected\ triplets}$$

In the network shown on the right, the set of connected triplets is {D-A-B, D-A-C, A-B-C, B-C-A, C-A-B}. There is one triangle. The GCC of this network is $\frac{3}{5}$.

Write a MapReduce program to compute the global clustering coefficient of the input undirected graph. You should print your output (a single number) to: "*/user/<User ID>/lab3/exp2/output*"