# Assignment C

Network analytics using Giraph and GoFFish

*Weightage: 150 points (15%), with extra credit of up to 25 points*

*Posted date: Thu 16 Mar, 2017*

*Due date: Fri 31 Mar, 2017 by 11:59PM IST*

## Introduction

In this assignment, you will learn to implement vertex centric graph algorithms using Giraph, design new subgraph centric algorithms and implement them using GoFFish, and evaluate the weak and strong scaling of these platform. Usual guidelines for using the shared turing cluster hold.

## 1. Implement Vertex-centric Algorithms *[30+ points]*

In the Table 1 below, you are given a list of vertex centric algorithms that have appeared in literature, along with the references. You should implement 2 or more of these algorithms in Giraph using a vertex-centric model. The goal here is to translate a given distributed algorithm to a working implementation using the Giraph Big Data platform. You can use any feature of Giraph for the same. You must validate this algorithm for at least 2 graphs from the Table 2 below, and report the results. Separate out the loading and superstep compute times. You may implement as many algorithms as you wish, and the points are cumulative.

NOTE: If you are using any code base that already exists on the web, in full or in part, you must cite such source and clearly indicate the extent of reuse and your own contribution. Points awarded will be scaled down proportional to the extent of reuse.

## 2. Design and Implement Subgraph-centric Algorithms *[30+ points]*

For one of the algorithms in Task 1 above, design and implement its subgraph-centric algorithm using GoFFish (Hama flavor). These algorithms carry twice the points as the vertex-centric ones since you're both designing and implementing the algorithm. You must validate this algorithm for at least 2 graphs from the Table 2 below, and report the results. Separate out the loading and superstep compute times. You may implement as many algorithms as you wish, and the points are cumulative.

NOTE: If the subgraph-centric algorithm or implementation is already available in literature, you must cite such source and clearly indicate the extent your own contribution. Points awarded will be proportional to your own contribution to the algorithm and implementation.

## 3. Weak and Strong Scaling *[20+ points]*

For the algorithms in Tasks 1 and 2 above, evaluate their weak and strong scaling. For weak scaling, you will run the algorithms on all 7 graphs with the number of worker slots proportional to (|V|+|E|) and see if the total superstep time remains constant. For strong scaling, you will observe if increasing the number of worker slots for a single graph proportionally reduces the total superstep time. Each pair of weak and

strong scaling experiments for one (vertex or subgraph centric) algorithm carries 20 points. You may report scalability experiments for as many algorithms as you wish, and the points are cumulative.

## 4. Extra Credit *[25 points]*

You may implement additional algorithms and experiments for Tasks 1, 2 and 3 above and get up to an additional 25 points of extra credit. However, extra credits will be given only if at least 1 vertex and 1 subgraph centric algorithm along with their weak and strong scaling results is submitted.

**Table 1: Algorithms, literature and points**

| Algorithm | Reference | Points |
|---|---|---|
| PageRank w/ convergence | SIGMOD 2010 | 15 |
| SubgraphRank w/ convergence *(compare to vertex-centric PageRank)* | COMAD 2012 | 30 |
| Bipartite Matching | SIGMOD 2010 | 15 |
| Semi-clustering | SIGMOD 2010 | 25 |
| Strongly Connected Component | VLDB 2014 A | 25 |
| Weakly Connected Component[1] | VLDB 2014 A | 15 |
| Graph Coloring | VLDB 2014 A | 40 |
| Euler Tour | VLDB 2014 B | 25 |
| Spanning Tree | VLDB 2014 B | 25 |
| K-Truss | ASONAM 2012 | 25 |
| Conductance | TR 2012 | 30 |

*References*

| | |
|---|---|
| SIGMOD 2010 | Pregel: A System for Large-Scale Graph Processing, Grzegorz Malewicz, et al. |
| COMAD 2014 | Subgraph Rank: PageRank for SubgraphCentric Distributed Graph Processing, Nitin Badam, et al. |
| VLDB 2014 A | Optimizing Graph Algorithms on Pregellike Systems, Semih Salihoglu |
| VLDB 2014 B | Pregel Algorithms for Graph Connectivity Problems with Performance Guarantees, Da Yan |
| ASONAM 2012 | Using Pregel-like Large Scale Graph Processing Frameworks for Social Network Analysis, Louise Quick, et al. |
| TR 2012 | Tech Report: Compiling GreenMarl into GPS, Sungpack Hong et al |

**Table 2: Graphs for Testing and Scaling Experiments, hosted on turing**

| Graph | \|V\| x$10^6$ | \|E\| x$10^6$ |
|---|---|---|
| WEBG | 0.87 | 5.1 |
| CITP | 3.77 | 33.04 |
| LIVJ | 4.85 | 86.22 |
| ORKT | 3.07 | 234.37 |
| CARN | 1.96 | 2.76 |
| USRN | 23.95 | 58.33 |
| EURN | 50.91 | 108.1 |

[1] Not eligible if Strongly Connected Component done.