Group Assignment 2
CS 5990 Computational Social Systems

Grade: 50 points.

- This is group work. You will work with your team members and make one submission.

Computational resource: You may use the cluster resource provided for this class and/or personal computer.

PROJECT DESCRIPTION

1. **Design and implement a parallel algorithm for closeness centrality. You may also implement betweenness centrality for an extra credit of 25 points.**

- **Problem statement**: Design a parallel algorithm that will process a graph of size *n* and calculates the closeness centrality and/or **betweenness centrality for each node**. You will implement this algorithm using Message Passage Interface (MPI) libraries for Python[1].
- Note that the betweenness centrality for a node v_i is defined as:

$$C_b(v_i) = \sum_{s \neq t \neq v_i} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$$

Where,

σ_{st} : number of shortest paths from vertex s to t

$\sigma_{st}(v_i)$: number of shortest paths from s to t that pass through vi

[1] https://mpi4py.readthedocs.io/en/stable/

Input: You will use the following two social networks datasets for your performance analysis (from the SNAP[2] website). They are:
1. Facebook dataset. The dataset is provided on this page:
https://snap.stanford.edu/data/ego-Facebook.html
Specifically, use the file facebook_combined.txt.gz
https://snap.stanford.edu/data/facebook_combined.txt.gz

2. Twitter data. The twitter dataset is provided on this page:
https://snap.stanford.edu/data/ego-Twitter.html (Note the statistics about this dataset provided on this page) Specifically, you will use the file titled "twitter_combined.txt.gz":
https://snap.stanford.edu/data/twitter_combined.txt.gz

- **Output:** The first processor (ID=0), will print out to a local file called "output.txt", the centrality measures for all the vertices. The processor will also print five nodes with the top centrality values (if there are more than five nodes with the centrality values, then print any five) and also the average of the centrality values of all nodes on screen.
- **Program design:** You may use any APSPs algorithm to calculate the shortest paths required in centrality calculation. The key aspect of the project is to come up with a good parallel design. Points will be given for efficient design.
- **Performance study:** Tabulate the running times of your implementation with varying number of processors = 2, 4,8, 16,…... perform your study using your chosen social networks. Prepare plots showing the trends in the run time. Answer questions, such as
  - what is the parallel speedup for your implementation? Plot the speedup with P=2,4,8,16…
  - What is the cost of the implementation? Plot the cost with various values of P.
- **Report:** You will prepare a brief report that will describe your design, pseudocode, data structures, performance study results, and any other details required to evaluate your algorithm and implementation. You will also provide a time complexity and space complexity analysis of your algorithm.
- You will use the **MPI library for python** to implement the algorithm.
- You will conduct performance comparison of the serial and parallel versions of the centrality calculation algorithm.

2. Deliverables:

Deadline for submission: See canvas page

[2] https://snap.stanford.edu/data/

- A PowerPoint file that provides a description of your design, analysis. You may use flowcharts, UML etc. to provide details of your implementation. You will also provide the tasks performed by each team member.
- The source code files.
- The compiled files
- Each member of the team will submit the slides (same copy)

Each team will schedule a presentation with the instructor . Your presentation will last about 15 minutes. The instructor will email you to set up the presentations.

Anyone who misses the final presentation will not receive a grade for the project.

Late/re-submission

- Team will have the option to resubmit or make a late submission with a penalty of 20%. Team who want to make use of this option will make their presentation during the finals week of the semester.