# COMP5112/MSBD5009 Parallel Programming

# Assignment 1: MPI Programming

# Due on 17:00 pm on Mar. 17, 2021

## Instructions

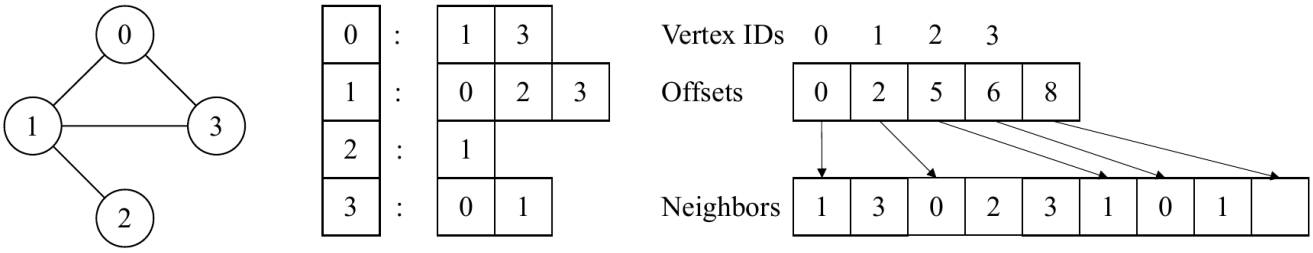
1. This assignment counts for 15points.
2. This is an individual assignment. You can discuss with others and search online resources but your submission should be your own code.
3. Add your name, student id and email as the first line of comments in your submission.
4. Submit your assignment through Canvas before the deadline.
5. No late submissions will be accepted!

## Assignment Description

Graph structural clustering is a common data analysis task to cluster vertices by their edge connections in the graph. SCAN (Structural Clustering Algorithm for Networks) [1] is such an algorithm that clusters vertices based on a structural similarity measure. The algorithm is efficient in both computation and memory.

In this assignment, you will implement an MPI program to perform clustering on multiple graphs by calling our provided sequential SCAN function in each MPI process.

In the SCAN implementation, each graph is stored in a CSR structure, illustrated as follows:




(A) Original Graph. (B) Neighbors. (C) CSR.

where one array is used to store the neighborhood information in sequential order, and the other array stores the offset for corresponding vertex.

Note: The length of array for storing offsets is the number of vertices plus one, as the first element of vertices is zero and points to the first element of the array of neighbors. Accordingly, the length of array for storing neighbors is the number of edges plus one.

The code skeleton `clustering_mpi_skeleton.cpp` is provided. Your task is to complete the missing code in the main function. Specifically, you need to:

1. Scatter the information of the graphs to each process.
2. Perform clustering in parallel.
3. Gather the results to the master process, which contain both number of clusters and cluster index of each vertex.
4. Output the clustering results to both screen and files (see the provided sequential version for details).

# Package list

```
assignment1-package/
  ├── datasets
  |   ├── test1
  |   |   ├── meta
  |   |   ├── 1.txt
  |   |   ├── ...
  |   |   └── 12.txt
  |   ├── test2
  |   |   ├── meta
  |   |   ├── 1.txt
  |   |   ├── ...
  |   |   └── 12.txt
  ├── results
  ├── clustering_h.cpp
  ├── clustering_mpi_skeleton.cpp
  ├── clustering_impl.cpp
  ├── clustering_sequential.cpp
  └── assignment1.pdf
```

1. datasets: sample input files of two test cases.
   The input files of a test case are in the following format:
   1. The number of graphs in the test case is stored in file **meta**.
   2. Each graph is stored in a text file. The first line is "`# nv ne`", where `nv` is the number of vertices and `ne` is the number of edges of the graph, respectively. Each of the following lines is a pair of integers "`v0 v1`", which represents an edge `(v0, v1)` in the graph.
2. clustering.h, clustering_impl.cpp: the source files of the SCAN implementation and some utilities.
3. clustering_sequential.cpp: the sequential implementation.
4. results: folder for saving the output results.

5. clustering_mpi_skeleton.cpp: MPI skeleton.
6. assignment1.pdf: the assignment description.

Note.1 The number of input graphs is assumed to be divisible by the number of processes. Each of the provided test cases contains 12 graphs.

Note.2 The clustering results of your MPI program should be the same as that of the provided sequential version.

Please complete the `clustering_mpi_skeleton.cpp` after the comment `// ADD YOUR CODE HERE` and make sure you only modify this file.

Please only submit `clustering_mpi_skeleton.cpp` to Canvas.

# Compile and run

## *Sequential*

```
g++ clustering_sequential.cpp clustering_impl.cpp -o clustering_sequential -std=c++11 -g
-Wall
./clustering_sequential <test_folder> <result_folder>
```

`<test_folder>` is the path of the input files and `<result_folder>` is the path of the output results. E.g.,

```
./clustering_sequential ./dataset/test1/ ./results/
```

## *MPI version*

```
mpic++ -std=c++11 clustering_mpi_skeleton.cpp clustering_impl.cpp -o clustering
mpiexec -n <num_processes> ./clustering <test_folder>  <result_folder>
```

`<test_folder>` is the directory of the test files and `<result_folder>` is the path of the output results. `<num_process>` is the number of total processes. E.g.,

```
mpiexec -n 4 ./clustering ./dataset/test1 ./results/
```

An example output is here for your reference:

```
num cluster in graph 0 : 3
num cluster in graph 1 : 2
num cluster in graph 2 : 3
num cluster in graph 3 : 2
num cluster in graph 4 : 2
num cluster in graph 5 : 2
num cluster in graph 6 : 3
num cluster in graph 7 : 3
num cluster in graph 8 : 2
num cluster in graph 9 : 3
num cluster in graph 10 : 3
num cluster in graph 11 : 3
Elapsed Time: 0.335364000 ms
```

And an example result file `results/0.txt` is output as follows:

```
3
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 -1 2 2 2 2 2 2
```

where `3` is the number of total clusters and `-1 0 1 2` is the cluster index for each vertex (`-1` for not in any cluster).

Note: mpiexec may output warning about “unable to find network interfaces”. You can run the following command and re-login the VM to suppress the warning.

```
echo "alias mpiexec='mpiexec -mca btl_base_warn_component_unused 0'" >> ~/.bashrc
```

# References


1. Xu, Xiaowei, et al. "Scan: a structural clustering algorithm for networks." Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 2007.