Homework 9: Implement Graph Algorithms with GridGraph

Introduction to Big Data Systems course

Due: December 4, 2022 23:59 China time. Late submission results in lower (or even no) scores.

For questions or concerns, contact TA (Huanqi Cao, Mingzhe Zhang) by WeChat. Or send an email to caohq18@mails.tsinghua.edu.cn or zmz21@mails.tsinghua.edu.cn if you could not use WeChat.

Overview

In this assignment, you need to implement **TWO** of the following algorithms with GridGraph and C++, aiming to use the graph computing system to solve a problem.

Environment

You can do this assignment on your own PC or on the server.

We have offered the GridGraph source code directory on the server at: `/data/hw9_GridGraph`

About GridGraph

GridGraph is a large-scale out-of-core graph processing framework on a single machine, you can refer to this paper for detail about GridGraph:

https://www.usenix.org/system/files/conference/atc15/atc15-paper-zhu.pdf

Source code and build instructions:

https://github.com/thu-pacman/GridGraph

It can be built directly on Ubuntu with `gcc` (by typing `make`). If you wish to build in MacOS or Windows platform, prepared to meet troubles. The system runs best with machine with SSDs, but hard disk is also fine

How to write your own application with GridGraph

You can just place your code in the subdirectory called `example`, then modify the `Makefile`. After compiled successfully, you will get the application in the subdirectory called `bin`.

Hint:

You should first read the `README.md` in the repo to learn how to compile and run application with GridGraph. And it will be helpful for you to understand how to program with GridGraph and finish this homework by reading the code of example applications (like PageRank, BFS) in the `example` directory.

Candidate Algorithms

Notice: You should implement TWO (2) of the following algorithms with GridGraph at least.

The candidate algorithms, as well as its reference sample code in GraphChi, Ligra or X-Stream, is as follows. **And maybe you should first understand these algorithms with the sample code in them or other resources.**

- **Conductance:** https://github.com/epfl-labos/x-stream/blob/master/algorithms/scan/conductance.hpp
- **PageRank-Delta:**

  https://github.com/jshun/ligra/blob/master/apps/PageRankDelta.C

  https://github.com/epfl-labos/x-stream/blob/master/algorithms/pagerank/pagerank_ddf.hpp
- **K-Cores:** https://github.com/GraphChi/graphchi-cpp/blob/master/toolkits/graph_analytics/kcores.cpp

Dataset

You just need to test your programs on the LiveJournal dataset below. The original graph data is stored in binary form.

1. **livejournal (|V|=4847571)**

   The graph data offered is directed and unweighted. The input of some algorithms is undirected or weighted graph, so you should do some transformation in advance. We have provide some tools in code directory:

   - `add_weight`: add a random weight to each edge
   - `make_undirect`: make an undirected graph from an existing directed graph

We have placed dataset on the server at: `/data/hw9_data`

If you want to do this assignment on your own computer, you can download the dataset from: https://cloud.tsinghua.edu.cn/d/985c3395496d45879825/

Hand-in

Please submit your assignment containing your report (including the implementation and the optimization, the performance) and the code using GridGraph.

Please describe your implementation in detail in your report.