Overview

During the last years, a wide range of huge networks has been made available to researchers. In computer science, engineering, and related fields, graph partitioning is a common technique. For example, in parallel computing good partitionings of unstructured graphs are very valuable. In this area, graph partitioning is mostly used to partition the underlying graph model of computation and communication. Roughly speaking, nodes in this graph denote computation units, and edges represent communication. This graph needs to be partitioned such that there are few edges between the blocks (pieces). In particular, if we want to use \( k \) processors we want to partition the graph into \( k \) blocks of about equal size while minimizing the number of edges cut or some other objective function that models communication of the application.

The current models used are very useful for scientific simulations in which networks do not feature vertices with very large degree. With a shifting focus towards partitioning social networks, this is no longer the case. In social networks there are typically lots of high degree vertices. As these vertices yield a bottleneck in computation, they need special treatment. For example, they could be replicated on all processors and hence, the information associated with them can be communicated more efficiently using broadcast and/or reduce operations. However, this implies that we need a new model for partitioning graphs and also methods to solve the model.

The main task of this thesis is to define a clear model and implement an algorithm to solve it. The includes the implementation of a number of heuristic algorithms. At the same time, we want to look at applications and provide empirical evidence that this approach makes a difference. This project has the chance to have a large impact in the scientific community.

Requirements

- Interest in algorithms and data structures
- Excellent programming skills in C++
- Ability to think for yourself

INFO: More topics available such as algorithm configuration.