Parallelizing Community Detection Algorithms in Large Networks

Suraj Ketan Samal

Department of Computer Science, University of Nebraska,Lincoln

The Problem

Given a network, is it possible to uncover its modules naturally without specifying their number or their size? Does such a solution scale well for large graphs ?

Figure 1: Possible communities in a flock of birds.

Two parallel approaches

Parallel Label Propagation(PLP)

- Each node is initialized with a unique label and iteratively adopts a label that most of its neighbors have, ties broken randomly.
- Each iteration takes linear time in the number of edges $O(E)$.
- Algorithm stops when there is no more change in labels of the nodes.

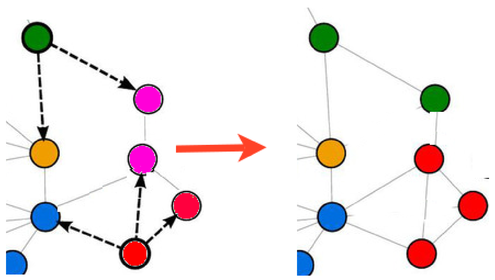

Figure 2: An iteration of Parallel Label Propagation(PLP) Algorithm [2]

- Parallel implementation is achieved by processing the *nodes* simultaneously using multiple threads.
- Each thread owns a set of nodes and updates their labels based on it's neighbours.
- All threads synchronize after each iteration.

Another approach

Parallel Modularity Optimization(PMO)

- Modularity Q = fraction of edges within communities *minus* expected fraction of such edges.
- Nodes are ordered, initialized with it's own label and traversed in order and adopt a label of its neighbours such that the modularity gain is maximum.
- Algorithm works in linear time in number of edges $O(E)$.

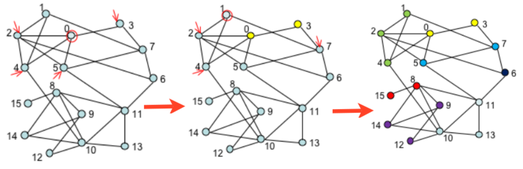

Figure 3: Modularity Optimization Algorithm(MO) [3]).

- Iteration continues till a local maxima is reached, after which coarsening is done by making the communities as the nodes.
- Parallel implementation is achieved by processing the nodes in each iteration using multiple threads.

Analysis and Observations

DataSet	Vertices(v)	Edges(e)
jazz	198	2742
cond-mat-2005	16726	47594
fe_ocean	143437	409593
144	144649	1074393
wave	156317	1059331
m14b	214765	1679018
auto	448695	3314611
in-2004	1382908	13591473
asia	11950757	12711603

Figure 5: Datasets chosen for our experiments [5].

- Nine datasets from small to large were chosen for our experimentsfrom DIMACS archive.
- Using OpenMP, linear speedup was only observed till *four* processors after which there was no change.
- PLP outperformed PMO for small datasets because of simpler approach. However for very large datasets, they both perform similar.

Implementation using NetworKit

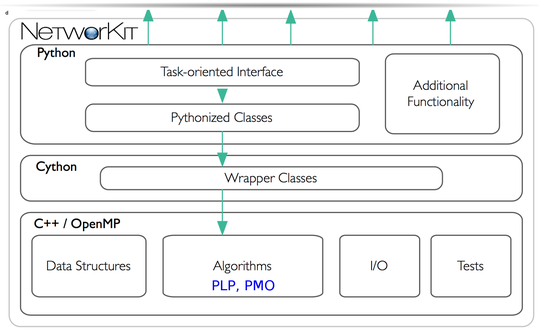

Figure 4: Architecture of networkit tool used for our implementation.

- NetworKit is a high-performance network analysis tool consisting of C++ core and uses shared momory (OpenMP) for parallelism.
- We decoupled the C++ code and attempted to execute the two algorithms on various parallel architecture (shared memory, GPU and PGA) models.
- Scaling experiments(1-16 processors) and comparison between the two algorithms was done for small and large datasets.
- Experiments performed on *crane* cluster (2CPU/16 cores and 64GBRAM per node) at UNL.

Preliminary Results

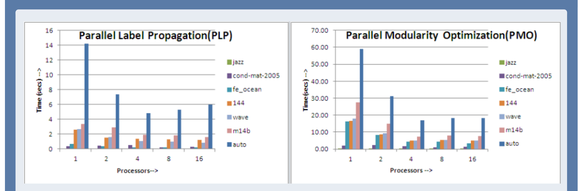

Figure 6: Strong scaling using existing (OpenMP) model.

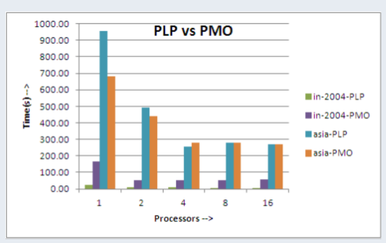

Figure 7: PLP versus PMO on larger datasets (*asia* and *in-2004*).

Conclusions and Outlook

- Parallelism seen to be used only with node traversals and hence a good amount of future scope for performance improvement.
- Inability to port and run using UPC (PGAS Model) or CUDA (GPU) due to underlying complex architecture.
- Advanced Graph storage techniques and processing algorithms (approximation & heuristics) need to be explored based on the underlying architecture to minimize communication between processors.
- Analysis of other high performance computing tools may give better insights.

References

[1] Staudt, C., and Henning Meyerhenke. Engineering Parallel Algorithms for Community Detection in Massive Networks.

[2] Raghavan, Usha Nandini, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks." Physical Review E 76.3 (2007): 036106.

[3] Blondel, Vincent D.,"Fast unfolding of communities in large networks." Journal of Statistical Mechanics: Theory and Experiment 2008.10 (2008): P10008.

[4] NETWORKIT (High Performance Network Analysis), https://networkit.iti.kit.edu/

[5] David A. Bader, Andrea Kappes, Henning Meyerhenke, Peter Sanders, Christian Schulz and Dorothea Wagner. Benchmarking for Graph Clustering and Partitioning. In Encyclopedia of Social Network Analysis and Mining, pages 73-82. Springer, 2014

University of Nebraska Lincoln