Parallel Graph Algorithms on Modern Systems

George M. Slota (gms5016@psu.edu, www.gmslota.com), Penn State University
Kamesh Madduri, Penn State (advisor), Siva Rajamanickam, Sandia National Labs (mentor)

Research Motivation

Graph analysis is key for the study of biological, chemical, social, and other networks.
- Real-world graphs are big, irregular, complex
  - Graph analytics is one of DARPA’s 23 toughest mathematical challenges
  - Web graph: 50B sites, 1T+ links; Brain graph: 100B neurons, 1,000T synaptic connections
- Modern computational systems are also big and complex
  - Multiple levels of parallelism, memory hierarchy, configurations
  - Heterogeneous – host, GPU, coprocessors (Xeon Phi MIC)
  - Optimization – account for socket-level, node-level, and system-level

How can we design graph algorithms to be performant on large modern systems?

Summary of Contributions

- Multistep approach to graph connectivity
  On average 2× faster than prior state-of-the-art for SCC
  Key optimizations: thread-local queues, minimize global atomics and synchronization, direction optimizing BFS

- Multistep approach to graph connectivity (Slota et al. 2014)
- On average 2× faster than prior state-of-the-art for SCC
- Key optimizations: thread-local queues, minimize global atomics and synchronization, direction optimizing BFS

Algorithm Design for Graph Analytics

Graph Connectivity, Strong Connectivity, and Weak Connectivity

- Multistep approach to graph connectivity
- On average 2× faster than prior state-of-the-art for SCC
- Key optimizations: thread-local queues, minimize global atomics and synchronization, direction optimizing BFS

- Multistep approach to graph connectivity
- On average 2× faster than prior state-of-the-art for SCC
- Key optimizations: thread-local queues, minimize global atomics and synchronization, direction optimizing BFS

Fast Approximate Subgraph Counting using Color-coding

- Implemented the Alon et al. 1995 color-coding approach for counting tree-structured subgraphs
- Shared and distributed memory implementations (FASCIA) give real-time count estimates of up to 7 vertex subgraphs on billion edge networks (order-of-magnitude improvement over prior art) (Slota and Madduri 2013, 2014, 2015)

Partitioning and In-memory Layout for Large Irregular Graphs

- Developed the PuLP partitioner for real-world irregular graphs (Slota et al. 2014)
- Exploits the label propagation community detection algorithm
- Order-of-magnitude faster execution times and memory consumption than state-of-the-art with comparable partition quality
- Allows for concurrent multiple constraint and multiple objective partitioning
- Minimize edge cut and max per-part edge cut (communication) while balancing vertices and edges per-part (computation and memory)

Acknowledgements

This research is part of the Blue Waters sustained-petascale computing project, which is supported by National Science Foundation awards OCI-0725070 and ACI-1133894, and by the U.S. Department of Energy’s Office of Advanced Scientific Computing Research under contract DE-AC02-06CH11357.