Irregular Graph Algorithms on Parallel Processing 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: 50 B sites, 1 T+ links; Brain graph: 100 B neurons, 1,000 T synaptic connections
- Modern computational systems are also big and complex
  - Multiple levels of parallelism, memory hierarchy, configurations
  - Heterogenous – host, GPU, coprocessors (Xeon Phi MIC)
  - Optimization – account for thread, socket, node, and system level scale

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

Summary of Contributions

- **Multistep** connectivity algorithms; on average 2× faster than prior state-of-the-art
- **FASCIA** and **FastPath** subgraph counting and min-weight path finding programs; up to orders-of-magnitude execution time improvement over prior art
- **PuLP** multi-objective multi-constraint partitioner; order of magnitude faster, order of magnitude less memory, comparable or better partition quality than state-of-the-art utilities for irregular small-world networks
- **Distributed Graph Layout** methodology for distributed memory; up to 12× performance improvements over naive methods
- **Complex Analysis** of largest publicly available web crawl using techniques derived from above work (3.5B vertices and 129B edges), analytic suite completes in only minutes

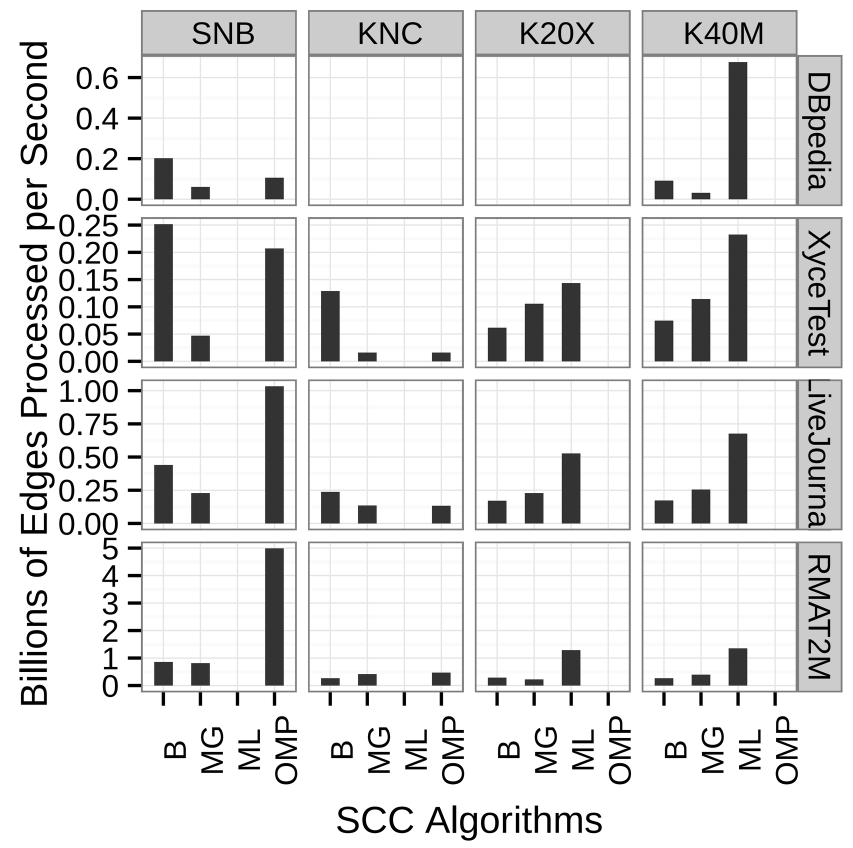
Acknowledgements

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070, ACI-1238993, and ACI-1444747) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. This work is also supported by NSF grants ACI-1253881, CCF-1439057, and the DOE Office of Science through the FASTMath SciDAC Institute. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

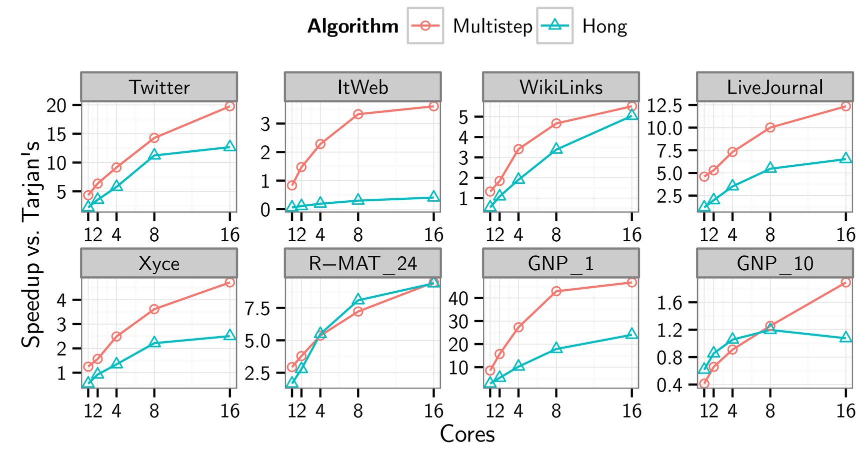
Algorithm Design for Graph Analytics

Graph Connectivity, Weak Connectivity, and Strong Connectivity

- **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



Cross-platform comparison of optimized manycore code.

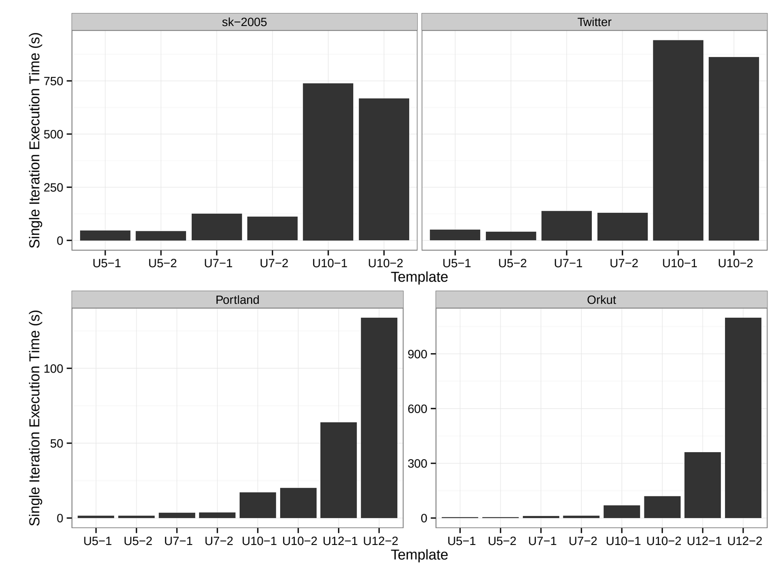


Multistep performance scaling relative to the state-of-the-art Hong et al. 2013 SCC code.

- Optimized SCC code for high performance on manycore processors such as GPUs (Slota et al. 2015)
- Up to 3.25× performance improvement over optimized CPU code on irregular test graphs
- **Key optimizations**: loop manipulation for higher parallelism, shared-memory and other locality considerations, warp and team-based atomics and operations (team scan, team reduce, etc.)

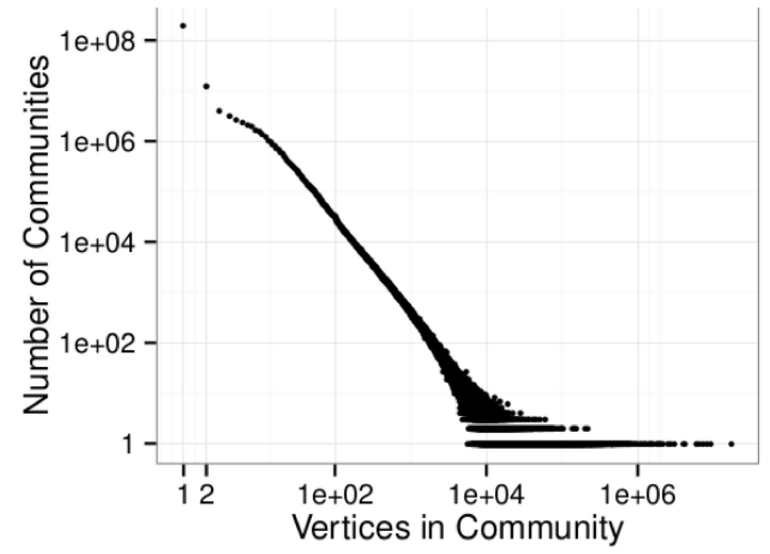
Fast Approximate Subgraph Counting using Color-coding

- **FASCIA**: implementation of color-coding subgraph counting algorithm (Alon et al. 1995)
- *Orders-of-magnitude improvement* over prior art (Slota and Madduri 2013, 2014, 2015)
- **Key optimizations**: abstracting dynamic programming phase, use of communication and computation avoidance and memory reduction strategies



Time to count subgraphs of varying sizes on several networks in shared (top) and distributed (bottom) memory.

Analysis of 2012 WDC Hyperlink Graph on Blue Waters

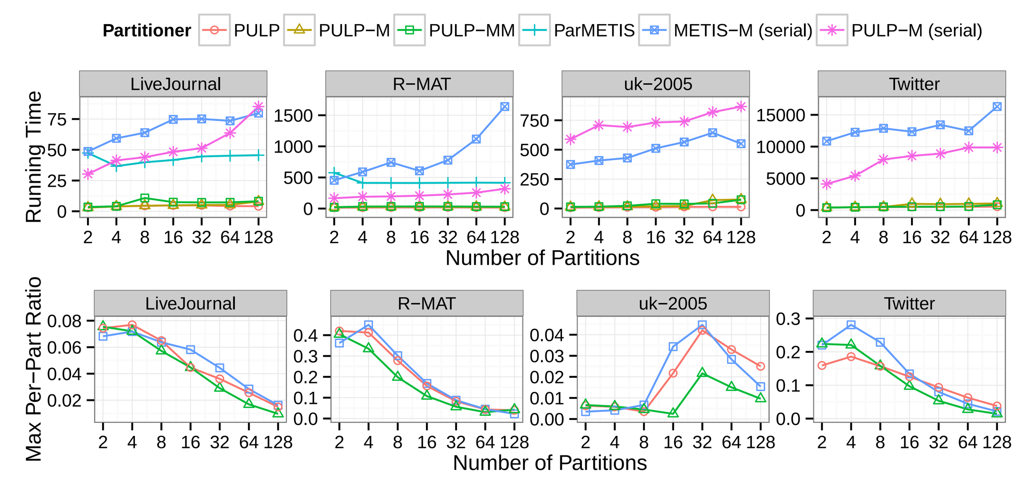


Community frequency distribution in the 2012 Web Data Commons hyperlink graph derived from Common Crawl corpus.

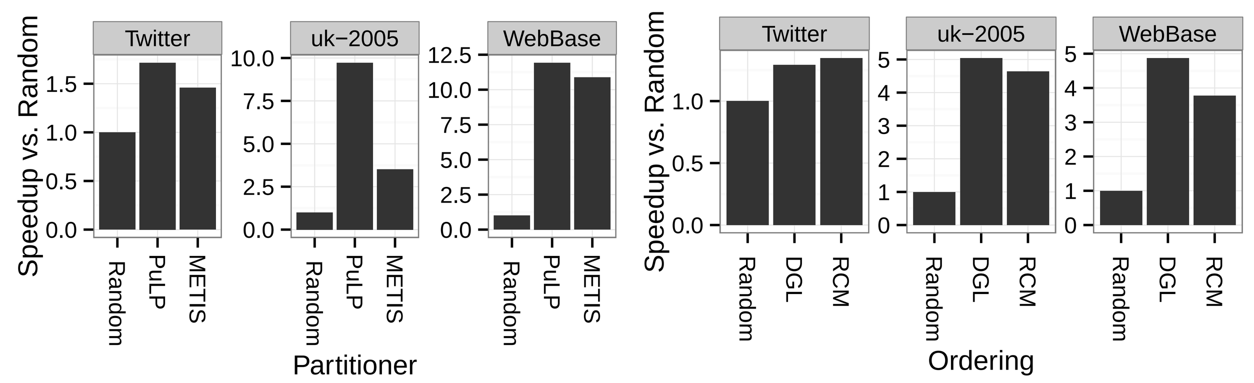
- Utilized multiple optimization strategies to implement several complex analytics (connectivity, k-core, centrality measures, and **community detection**)
- Community structure of Internet follows apparent power law with heavy tail distribution, mirroring multiple other structural frequency plots (Meusel et al. 2014)

Partitioning and In-memory Layout for Irregular Graphs

- Developed the **PuLP** partitioner for real-world irregular graphs (Slota et al. 2014)
- **Order-of-magnitude faster execution times and memory consumption** than state-of-the-art with comparable partition quality
- Allows for concurrent multiple constrain and multiple objective partitioning



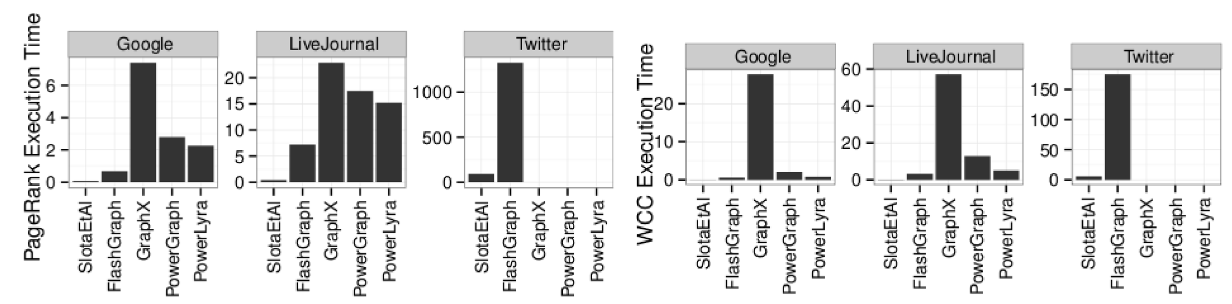
Execution times for PuLP relative to METIS and ParMETIS (top) and partition quality in terms of edge cut for PuLP and ParMETIS (bottom).



Impact of partitioning quality (left) and intra-task vertex orderings (right) on execution times of graph analytics.

- Developed new fast ordering method, and analyzed graph analytic performance impacts of partitioning and intra-part vertex ordering
- PuLP partitioning can have up to a **12× speedup** on communication times; our vertex ordering can have up to **5× speedup** on computation times against naive methods

Distributed Graph Analysis



Comparison of graph analysis frameworks for PageRank (left) and WCC (right).

- Developed implementation strategy for large scale distributed graph analytics
- **Can process graphs with hundreds of billions of edges in minutes** on only 4 K cores
- Figures above compare PageRank and WCC implementations to several popular frameworks, demonstrate 97× and 37× average speedup across test set
- Implementations only require a few hundred lines of C code at most
- **Takeaway**: HPC systems can be effectively utilized for large-scale graph analysis