# 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
  - Heterogenous – 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** connectivity algorithms; on average 2× faster than state-of-the-art
- **FASCIA** subgraph counting program; 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
- Methodology for in-memory **distributed graph layout**; 1.5 – 5× performance improvements over naive methods
- Using techniques derived from above accomplishments, provided **first-ever complex analysis** of largest publicly available web crawl (3.5B vertices and 129B edges)

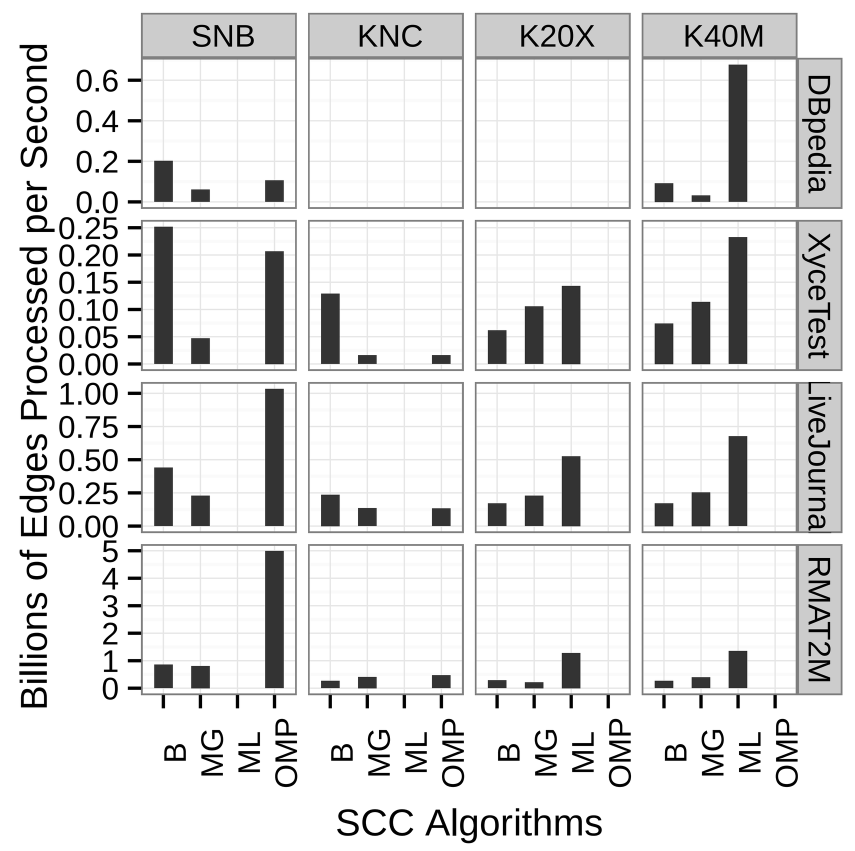
## Acknowledgements

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.

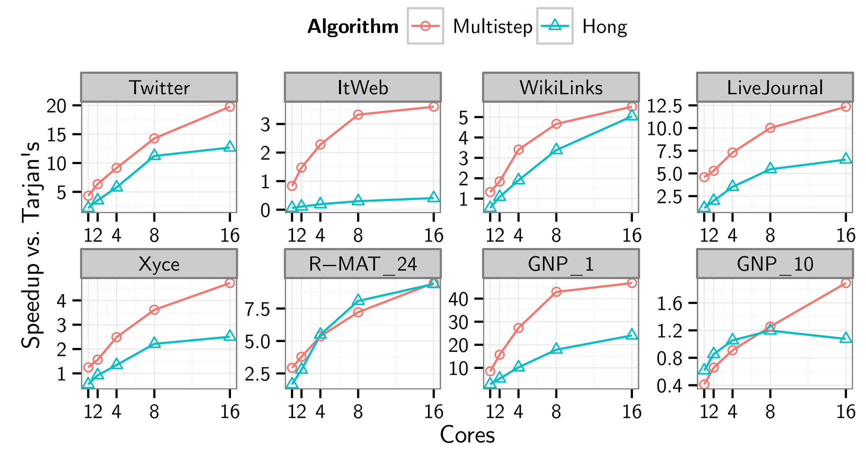
## Algorithm Design for Graph Analytics

### Graph Connectivity, Strong Connectivity, and Weak 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.*

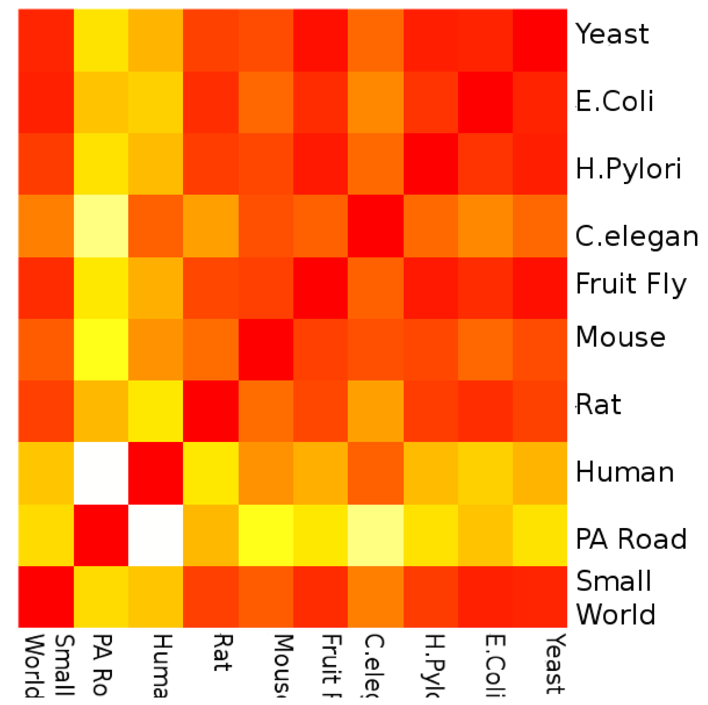



*Multistep performance scaling relative to the state-of-the-art Hong et al. 2013 SCC code.*

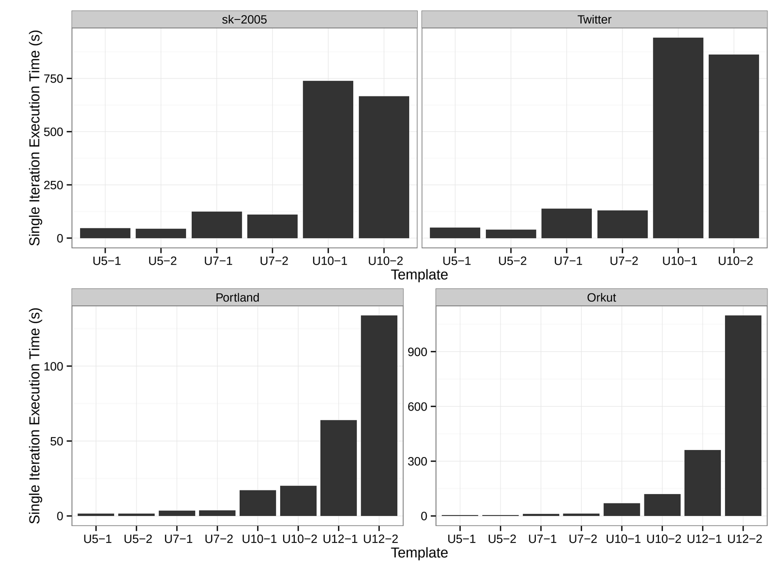
- Optimized SCC code for high performance on manycore processors such as GPUS (Slota et al. 2015)
- 3.25× performance improvement over optimized CPU code on most irregular test graphs
- **Key optimizations**: loop manipulation for higher parallelism, shared-memory and other locality considerations, warp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

### 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)




*Graphlet frequency distance between various biological and other networks. Red indicates high inter-network similarity while yellow and white indicate lower similarity.*

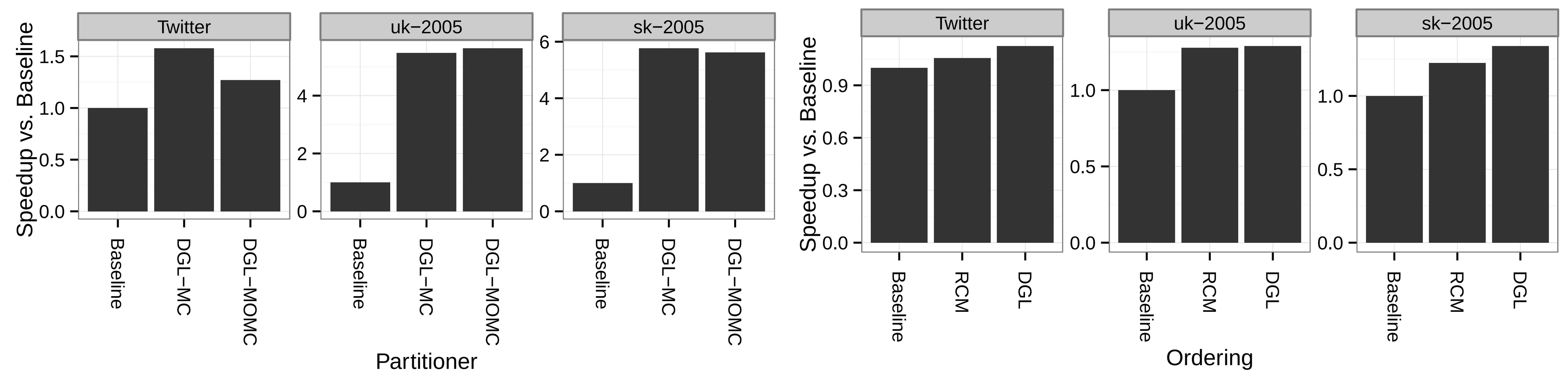



*Time to count subgraphs of varying sizes on several networks in shared (top) and distributed (bottom) memory.*

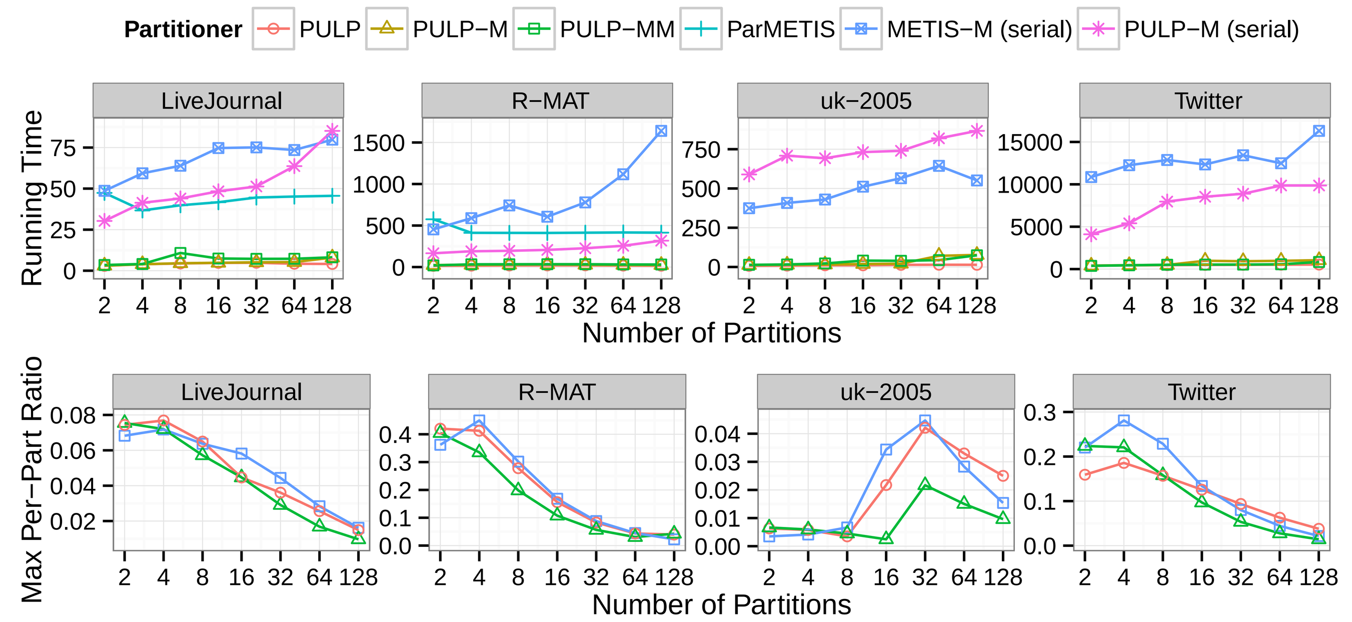
- Fast counts allow similarity comparison between large networks
- Subgraph frequency analysis allows for detection of recurring motifs on large networks
- **Key optimizations**: restructure algorithm to exploit simpler indexing methods, use of efficient communication and computation avoidance strategies

## 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 constrain and multiple objective partitioning
- Minimize edge cut and max per-part edge cut (communication) while balancing vertices and edges per-part (computation and memory)




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




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

- Analyzed distributed graph layout in terms of partitioning objectives and intra-part vertex ordering
- Partition quality and objectives can have up to a 5× speedup on execution times; vertex orderings on larger networks is also considerable, with up to 1.5× speedup against naive methods