**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
  - Heterogeneous – 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

**Algorithm Design for Graph Analytics**

**Multistep approach to graph connectivity** (Slota et al. 2014)

- On average 2× faster than prior state-of-the-art

- **Key optimizations**: thread-local queues, minimize global synchronizations, direction optimizing BFS

- 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.)

- 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

- 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**

- 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