Massive-Scale Analytics of Streaming Social Networks
David A. Bader
Exascale Streaming Data Analytics:
Real-world challenges

All involve analyzing massive streaming complex networks:

- **Health care** → disease spread, detection and prevention of epidemics/pandemics (e.g. SARS, Avian flu, H1N1 “swine” flu)
- **Massive social networks** → understanding communities, intentions, population dynamics, pandemic spread, transportation and evacuation
- **Intelligence** → business analytics, anomaly detection, security, knowledge discovery from massive data sets
- **Systems Biology** → understanding complex life systems, drug design, microbial research, unravel the mysteries of the HIV virus; understand life, disease,
- **Electric Power Grid** → communication, transportation, energy, water, food supply
- **Modeling and Simulation** → Perform full-scale economic-social-political simulations

Sample queries:

- **Allegiance switching**: identify entities that switch communities.
- **Community structure**: identify the genesis and dissipation of communities
- **Phase change**: identify significant change in the network structure

Ex: discovered minimal changes in O(billions)-size complex network that could hide or reveal top influencers in the community

REQUIRES PREDICTING / INFLUENCE CHANGE IN REAL-TIME AT SCALE
Center for Adaptive Supercomputing Software (CASS-MT)

- CASS-MT, launched July 2008
- Pacific-Northwest Lab
  - Georgia Tech, Sandia, WA State, Delaware

- The newest breed of supercomputers have hardware set up not just for speed, but also to better tackle large networks of seemingly random data. And now, a multi-institutional group of researchers has been awarded $4.0 million to develop software for these supercomputers. Applications include anywhere complex webs of information can be found: from internet security and power grid stability to complex biological networks.
Objective
To design software for the analysis of massive-scale spatio-temporal interaction networks using multithreaded architectures such as the Cray XMT. The Center launched in July 2008 and is led by Pacific-Northwest National Laboratory.

Description
We are designing and implementing advanced, scalable algorithms for static and dynamic graph analysis, including generalized $k$-betweenness centrality and dynamic clustering coefficients.

Highlights
On a 64-processor Cray XMT, $k$-betweenness centrality scales nearly linearly (58.4x) on a graph with 16M vertices and 134M edges. Initial streaming clustering coefficients handle around 200k updates/sec on a similarly sized graph.

Our research is focusing on temporal analysis, answering questions about changes in global properties (e.g. diameter) as well as local structures (communities, paths).

David A. Bader (CASS-MT Task 7 LEAD)
David Ediger, Karl Jiang, Jason Riedy
Massive Data Analytics: Protecting our Nation

US High Voltage Transmission Grid (>150,000 miles of line)

Public Health
- CDC / Nation-scale surveillance of public health
- Cancer genomics and drug design
  - computed Betweenness Centrality of Human Proteome

The New York Times
Thursday, September 4, 2008

Report on Blackout Is Said To Describe Failure to React

BY MATTHEW L. WALK
Published: November 12, 2003

A report on the Aug. 14 blackout identifies specific lapses by various parties, including FirstEnergy’s failure to react properly to the loss of a transmission line, people who have seen drafts of it say.

A working group of experts from eight states and Canada will meet in private on Wednesday to evaluate the report, people involved in the investigation said Tuesday. The report, which the Energy Department

Human Genome core protein interactions
Degree vs. Betweenness Centrality

David A. Bader
Network Analysis for Intelligence and Surveillance

- [Krebs ’04] Post 9/11 Terrorist Network Analysis from public domain information
- Plot masterminds correctly identified from interaction patterns: centrality
- A global view of entities is often more insightful
- Detect anomalous activities by exact/approximate graph matching

Massive data analytics in Informatics networks

- Graphs arising in Informatics are very different from topologies in scientific computing.

- We need new data representations and parallel algorithms that exploit topological characteristics of informatics networks.

Emerging applications: dynamic, high-dimensional data

Static networks, Euclidean topologies
This image is a visualization of my personal friendster network (circa February 2004) to 3 hops out. The network consists of 47,471 people connected by 432,430 edges. Credit: Jeffrey Heer, UC Berkeley
Limitations of Current Tools

- Graphs with millions of vertices are well beyond simple comprehension or visualization: we need tools to summarize the graphs.
- Existing tools: UCINet, Pajek, SocNetV, tnet
- Limitations:
  - Target workstations, limited in memory
  - No parallelism, limited in performance.
  - Scale only to low density graphs with a few million vertices
- We need a package that will easily accommodate graphs with several billion vertices and deliver results in a timely manner.
  - Need parallelism both for computational speed and memory!
  - The Cray XMT is a natural fit...
The Cray XMT

- **Tolerates latency** by massive multithreading
  - Hardware support for **128 threads on each processor**
  - Globally hashed address space
  - No data cache
  - Single cycle context switch
  - Multiple outstanding memory requests

- Support for fine-grained, word-level synchronization
  - Full/empty bit associated with every memory word

- Flexibly supports dynamic load balancing

- GraphCT currently tested on a 128 processor XMT: **16K threads**
  - **1 TB** of globally shared memory

Image Source: cray.com
Graph Analysis Performance:
Multithreaded (Cray XMT) vs. Cache-based multicore

- SSCA#2 network, SCALE 24 (16.77 million vertices and 134.21 million edges.)

![Graph Analysis Performance Diagram]
What is GraphCT?

- **Graph Characterization Toolkit**
- Efficiently summarizes and analyzes static graph data
- Built for large multithreaded, shared memory machines like the Cray XMT
- Increases productivity by decreasing programming complexity
- Classic metrics & state-of-the-art kernels
- Works on many types of graphs
  - directed or undirected
  - weighted or unweighted
Key Features of GraphCT

- Low-level primitives to high-level analytic kernels
- Common graph data structure
- Develop custom reports by mixing and matching functions
- Create subgraphs for more in-depth analysis
- Kernels are tuned to maximize scaling and performance (up to 128 processors) on the Cray XMT

Load the Graph Data

Find Connected Components

Run $k$-Betweenness Centrality on the largest component
## GraphCT Functions

<table>
<thead>
<tr>
<th>Name</th>
<th>Name</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMAT graph generator</td>
<td>Modularity Score</td>
<td>Included</td>
</tr>
<tr>
<td>Degree distribution statistics</td>
<td>Conductance Score</td>
<td></td>
</tr>
<tr>
<td>Graph diameter</td>
<td>st-Connectivity</td>
<td>In Progress</td>
</tr>
<tr>
<td>Maximum weight edges</td>
<td>Delta-stepping SSSP</td>
<td></td>
</tr>
<tr>
<td>Connected components</td>
<td>Bellman-Ford</td>
<td></td>
</tr>
<tr>
<td>Component distribution statistics</td>
<td>GTriad Census</td>
<td></td>
</tr>
<tr>
<td>Vertex Betweenness Centrality</td>
<td>SSCA2 Kernel 3 Subgraphs</td>
<td></td>
</tr>
<tr>
<td>Vertex k-Betweenness Centrality</td>
<td>Greedy Agglomerative Clustering</td>
<td>Proposed/Available</td>
</tr>
<tr>
<td>Multithreaded BFS</td>
<td>Minimum spanning forest</td>
<td>Included</td>
</tr>
<tr>
<td>Edge-divisive Betweenness-based Comm Detection (pBD)</td>
<td>Clustering coefficients</td>
<td></td>
</tr>
<tr>
<td>Lightweight Binary Graph I/O</td>
<td>DIMACS Text Input</td>
<td></td>
</tr>
</tbody>
</table>
GraphCT Performance

- RMAT(24) : 16.7M vertices, 134M edges
- RMAT(28) : 268M vertices, 2.1B edges
  - BC : 2800s on 64P
  - CC : 1200s on 64P
STINGER: A Data Structure for Changing Graphs

- Light-weight data structure that supports efficient iteration \textit{and} efficient updates.

Experiments with Streaming Updates to Clustering Coefficients

- Working with bulk updates, can handle almost 200k per second
STING Extensible Representation (STINGER)

Enhanced representation developed for dynamic graphs developed in consultation with David A. Bader, Johnathan Berry, Adam Amos-Binks, Daniel Chavarría-Miranda, Charles Hastings, Kamesh Madduri, and Steven C. Poulos.

Design goals:
- Be useful for the entire “large graph” community
- Portable semantics and high-level optimizations across multiple platforms & frameworks (XMT C, MTGL, etc.)
- Permit good performance: No single structure is optimal for all.
- Assume globally addressable memory access
- Support multiple, parallel readers and a single writer

Operations:
- Insert/update & delete both vertices & edges
- Aging-off: Remove old edges (by timestamp)
- Serialization to support checkpointing, etc.
STING Extensible Representation

- Semi-dense edge list blocks with free space
- Compactly stores timestamps, types, weights
- Maps from application IDs to storage IDs
- Deletion by negating IDs, separate compaction
Testbed: Clustering Coefficients

- Roughly, the ratio of actual triangles to possible triangles around a vertex.

- Defined in terms of *triplets*.
  - *i*-j-ν is a **closed triplet** (triangle).
  - *m*-ν-n is an **open triplet**.

- Clustering coefficient
  # closed triplets / # all triplets

- Locally, count those around ν.

- Globally, count across entire graph.
  - Multiple counting cancels (3/3=1)
Streaming updates to clustering coefficients

- Monitoring clustering coefficients could identify anomalies, find forming communities, etc.
- Computations stay local. A change to edge \(<u, v>\) affects only vertices \(u, v\), and their neighbors.

![Diagram of a graph with vertices labeled u and v and edges connecting them]

- Need a fast method for updating the triangle counts, degrees when an edge is inserted or deleted.
  - Dynamic data structure for edges & degrees: STINGER
  - Rapid triangle count update algorithms: exact and approximate

Updating clustering coefficients

- Using RMAT as a graph and edge stream generator.
  - Mix of insertions and deletions

- Result summary for single actions
  - Exact: from 8 to 618 actions/second
  - Approx: from 11 to 640 actions/second

- Alternative: Batch changes
  - Lose some temporal resolution within the batch
  - Median rates for batches of size B:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>B = 1</th>
<th>B = 1000</th>
<th>B = 4000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>90</td>
<td>25 100</td>
<td>50 100</td>
</tr>
<tr>
<td>Approx.</td>
<td>60</td>
<td>83 700</td>
<td>193 300</td>
</tr>
</tbody>
</table>

- STINGER overhead is minimal; most time in spent metric.
Hierarchy of Interesting Analytics

- **Extend single-shot graph queries to include time.**
  - Are there s-t paths between time $T_1$ and $T_2$?
  - What are the important vertices at time $T$?

- **Use persistent queries to monitor properties.**
  - Does the path between s and t shorten drastically?
  - Is some vertex suddenly very central?

- **Extend persistent queries to fully dynamic properties.**
  - Does a small community stay independent rather than merge with larger groups?
  - When does a vertex jump between communities?

- **New types of queries, new challenges...**
Bader, Related Recent Publications (2005-2008)

Bader, Related Recent Publications (2009-2010)


- Karl Jiang, David Ediger, and David A. Bader. “Generalizing k-Betweenness Centrality Using Short Paths and a Parallel Multithreaded Implementation.” The 38th International Conference on Parallel Processing (ICPP), Vienna, Austria, September 2009.


- Seunghwa Kang, David A. Bader. “Large Scale Complex Network Analysis using the Hybrid Combination of a MapReduce cluster and a Highly Multithreaded System,” Fourth Workshop in Multithreaded Architectures and Applications (MTAAP), Atlanta, GA, April 2010.
NSF Computing Research Infrastructure: Development of a Research Infrastructure for Multithreaded Computing Community Using Cray Eldorado Platform

- The Cray XMT system serves as an ideal platform for the research and development of algorithms, data sets, libraries, languages, tools, and simulators for applications that benefit from large numbers of threads, massively data intensive, sparse-graph problems that are difficult to parallelize using conventional message-passing on clusters.
  - A shared community resource capable of efficiently running, in experimental and production modes, complex programs with thousands of threads in shared memory;
  - Assembling software infrastructure for developing and measuring performance of programs running on the hardware; and
  - Building stronger ties between the people themselves, creating ways for researchers at the partner institutions to collaborate and communicate their findings to the broader community.

FACULTY
David A. Bader, PI (GA Tech)

Collaborators include: Univ of Notre Dame, Univ. of Delaware, UC Santa Barbara, CalTech, UC Berkeley, Sandia National Laboratories

http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0708307