10:00–10:20 Me: *High-Performance Analysis of Streaming Graphs*

10:25–10:45 A. Erdem Sariyuce and Ali Pinar, *Dense Subgraphs in Temporal Networks: Algorithms and Analysis*

10:50–11:10 Anand Iyer and Ion Stoica, *Time-Evolving Graph Processing on Commodity Clusters*

11:15–11:35 Srikanta Tirthapura, *et al.*, *Parallel and Streaming Methods for Real-Time Analysis of Dense Structures from Graphs*

Continued in MS226 this afternoon, 2:15pm–3:50pm.

Continuation of MS200:

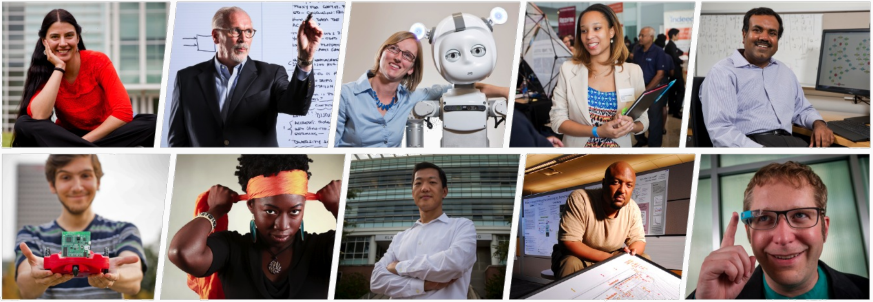
2:15–2:35 Elisabetta Bergamini and Henning Meyerhenke, *On Betweenness Centrality Problems in Dynamic Graphs*

2:40–3:00 Sriram Srinivasan and Sanjukta Bhowmick, *Predicting Movement of Vertices Across Communities in Dynamic Networks*

3:05–3:25 Keita Iwabuchi, *et al.*, *Large-Scale Dynamic Graph Processing on HPC Systems*

3:30–3:50 Anita Zakrzewska, *Creating Dynamic Graphs from Temporal Data*

Some slides to be posted at `http://graphanalysis.org`.



High-Performance Analysis of Streaming Graphs

E. Jason Riedy

School of Computational Science and Engineering
Georgia Institute of Technology

SIAM CSE, 2 March 2017

Outline

Motivation and Applications

Current and Future STINGER Models

Extracting Interesting Subgraphs

GPUs for Streaming Graphs?

Closing

Motivation and Applications

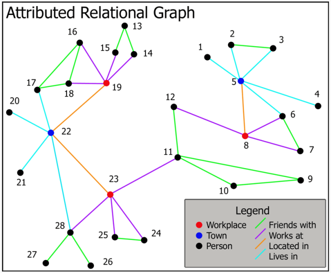
(insert prefix here)-scale data analysis

Cyber-security	Identify anomalies, malicious actors
Health care	Finding outbreaks, population epidemiology
Social networks	Advertising, searching, grouping
Intelligence	Decisions at scale, regulating markets, smart & sustainable cities
Systems biology	Understanding interactions, drug design
Power grid	Disruptions, conservation
Simulation	Discrete events, cracking meshes

Changes are important. Cannot stop the world...

Why Graphs?

Another tool, like dense and sparse linear algebra.



- Combine *things* with *pairwise relationships*
- Smaller, more generic than raw data.
- Taught (roughly) to all CS students...
- Semantic attributions can capture essential *relationships*.
- Traversals can be faster than filtering DB joins.
- Provide clear phrasing for queries about *relationships*.

Potential Applications

- Social Networks
  - Identify *communities*, influences, bridges, trends, anomalies (trends *before* they happen)...
  - Potential to help social sciences, city planning, and others with large-scale data.
- Cybersecurity
  - Determine if new connections can access a device or represent new threat in < 5ms...
  - Is the transfer by a virus / persistent threat?
- Bioinformatics, health
  - Construct gene sequences, analyze protein interactions, map brain interactions
- Credit fraud forensics $\Rightarrow$ detection $\Rightarrow$ monitoring
  - Real-time integration of all the customer's data

Streaming graph data

Network data rates:

- Gigabit ethernet: 81k – 1.5M packets per second
- Over 130 000 flows per second on 10 GigE ($< 7.7\ \mu$s)

Person-level data rates:

- 500M posts per day on Twitter (6k / sec)[1]
- 3M posts per minute on Facebook (50k / sec)[2]

But often analyze only **changes** and not *entire* graph.

Throughput & latency trade off and expose different levels of concurrency.

---

1 www.internetlivestats.com/twitter-statistics/

2 www.jeffbullas.com/2015/04/17/21-awesome-facebook-facts-and-statistics-you-need-to-check-out/

Streaming graph *analysis*

Terminology, will go into more details:

- *Streaming* changes into a massive, evolving graph
- Will compare models later...
- Need to handle *deletions* as well as insertions

Previous STINGER performance results (x86-64):

Data ingest $>$2M upd/sec [Ediger, McColl, Poovey, Campbell, & Bader 2014]

Clustering coefficients $>$100K upd/sec [R, Meyerhenke, B, E, & Mattson 2012]

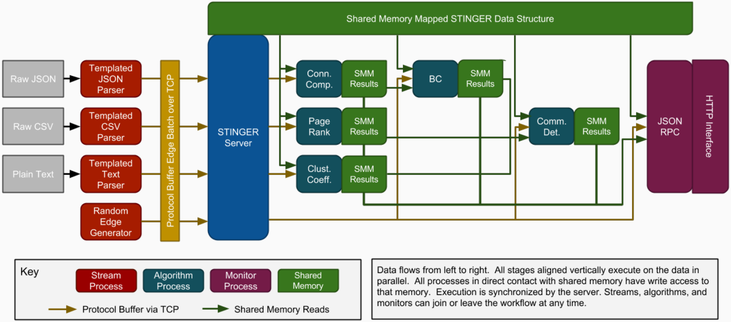
Connected comp. $>$1M upd/sec [McColl, Green, & B 2013]

Community clustering $>$100K upd/sec* [R & B 2013]

PageRank Up to 40$\times$ latency improvement [R 2016]

Current and Future STINGER Models

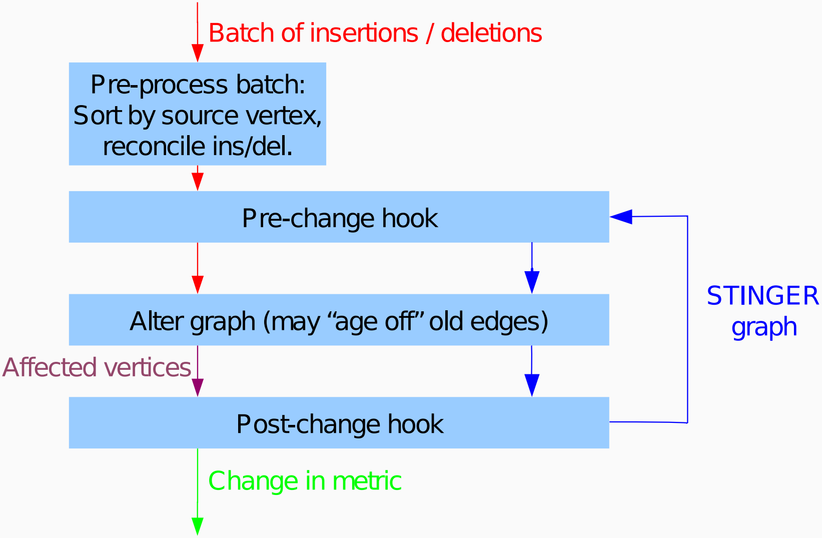
STINGER: Framework for streaming graphs



Slide credit: Rob McColl and David Ediger

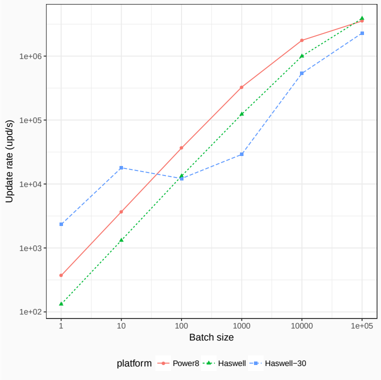
- OpenMP + sufficiently POSIX-ish
- Multiple processes for resilience

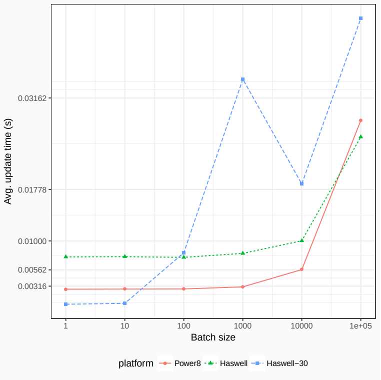
Current STINGER model



Is STINGER's current model good enough?

Data ingest rates, R-MAT into R-MAT, scales 24 & 30





Want to add analysis clients **without slowing data ingest!**

Note that scale 30 starts with 1.1B vertices, 17B edges...

(Different STINGER internal parameters.)

What if we don't hold up changes?

Additional STINGER model

Analyze concurrently with the graph changes, and produce a result correct for the starting graph and **some subset** of concurrent changes.[3]

Sample of other models

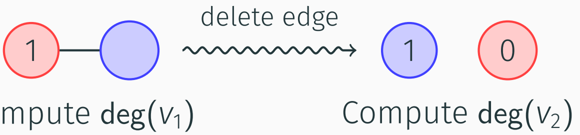
- Put in a query, wait for sufficient data [Phillips, *et al.*]
- Evolving: Sample, accurate w/ high-prob.
- Classical: dynamic algorithms, versioned data

[3] Chunxing Yin, Riedy, Bader. "Validity of Graph Algorithms on Streaming Data." January 2017. (in submission)

Algorithm validity in our model: Example.

Can you compute degrees in an undirected graph (no self loops) concurrently with changes?

Algorithm: Iterate over vertices, count the number of neighbors.



Cannot correspond to an undirected graph plus any subset of concurrent changes.

Valid for our model? No!

Not *incorrect*, just not valid for our model.

Algorithm validity in our model

- What is valid?
  - Typical BFS and follow-ons (betweenness centrality)
  - Shiloach-Vishkin connected components
  - PageRank? (hm.)
  - Saved decisions...
- What is invalid?
  - Making a decision twice in implementations.
    - Δ-stepping SSSP: Decrease a weight below Δ
    - Degree optimization: Cross threshold, miss vertex
  - Applying old information.
    - Labeling in S. Kahan's components alg.

Fun properties

Due to Chunxing Yin, under sensible assumptions:

- You can produce a single-change stream to demonstrate invalidity.
- Algorithms that produce a subgraph of their input *cannot be guaranteed* to run concurrently with changes and always produce snapshot outputs.

In progress:

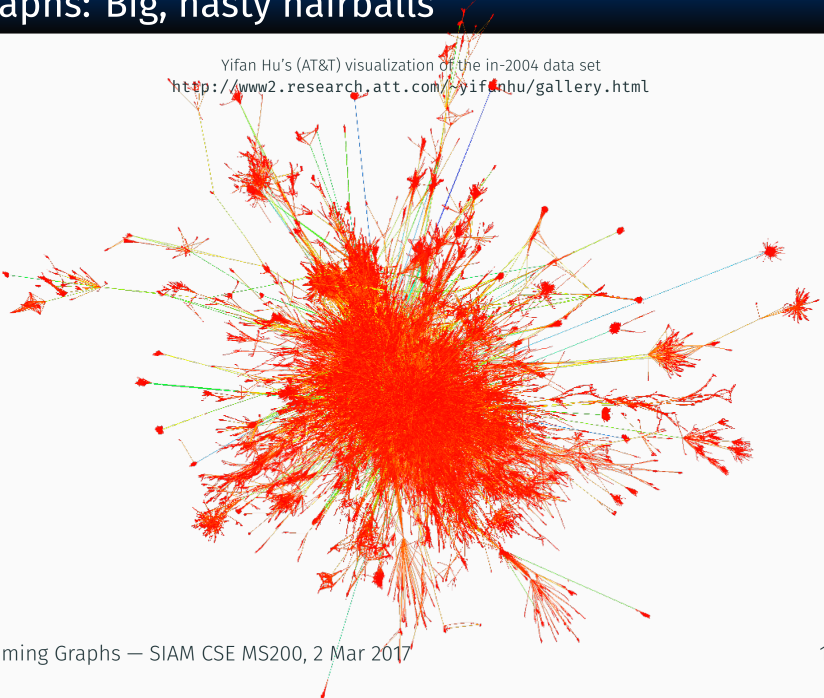
- Validity for **streaming**! Apply a algorithm valid for our model. Also collect the changes during execution. Now *update* the result for those changes while more changes accumulate. Repeat.
- Algorithms like PageRank... Actually nearby to graph + subset?
- Verification for debugging, *etc*.

Extracting Interesting Subgraphs

Graphs: Big, nasty hairballs

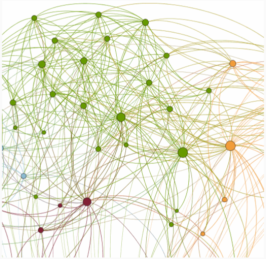
Yifan Hu's (AT&T) visualization of the in-2004 data set

`http://www2.research.att.com/~yifanhu/gallery.html`



But no shortage of structure...

in-2004, matrix format from Davis, Florida Sparse Matrix Collection

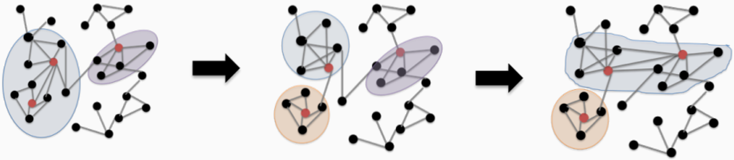


Jason's network via LinkedIn Labs

- Locally, there are clusters or *communities*.
- There are methods for *global* community detection.
- Also need *local* communities around *seeds* for queries and targeted analysis.

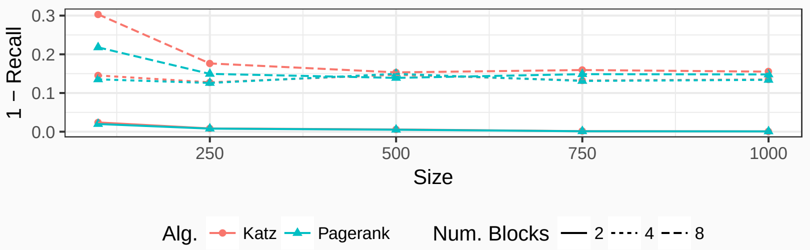
Seed set expansion

- Seed set expansion finds the *"best"* subgraph or communities for a set of vertices of interest
  - Many quality criteria: Modularity, conductance-ish, *etc.*
- Want to produce smaller expansions for viz. as well as larger communities for deeper analysis.
- Dynamic agglomerative / modularity algorithms update larger communities faster than recomputation [Zakrzewska & Bader]



PageRank and Katz centrality

Both PageRank and Katz centrality recover blocks in artificial stochastic block model graphs.



Working on *updating* the expanded sets using incremental iterations:

Updating PageRank [R]:

$$\Delta x^{(k+1)} = \alpha A_\Delta^T D_\Delta^{-1} \Delta x^{(k)} + \alpha (A_\Delta^T D_\Delta^{-1} - A^T D^{-1}) x + r_{|\Delta x^{(k+1)}}$$

Updating Katz:

$$\Delta x^{(k+1)} = \alpha A_\Delta \Delta x^{(k)} + (r - \alpha \Delta A x)_{|\Delta x^{(k+1)}}$$

Streaming seed set expansion

- Work in progress!
- Which seed set expansion methods provide subgraphs useful for further analysis? How do the results compare to global analysis?
- We do not want to maintain the *entire* $|V|$ PR or Katz vector, only around $|S|$ where S is the output.
- Can we continue applying earlier stopping criteria[4] for top-K separation?

[4] Eisha Nathan, Geoffrey Sanders, James Fairbanks, Van Emden Henson, David A. Bader. "Graph Ranking Guarantees for Numerical Approximations to Katz Centrality," Jan 2017. (in submission, Wed. CSE poster)

GPUs for Streaming Graphs?

So... Now what?

- Maintain these communities / subgraphs on or near *accelerators*!
- Sending *changes* may help with the connection bandwidth problem.
- cuSTINGER [Green & Bader]
  - A variant of STINGER for NVIDIA GPUs
  - Ingest at rates over 10^7 updates / sec
  - Ingest & triangle count updates at up to 2×10^6 upd/s (*higher* in prep!)
  - Amenable to existing high-performance static analysis kernels like betweenness centrality.
  - `https://github.com/cuStinger`

So... Now what?

- Maintain these communities / subgraphs on or near *accelerators*!
- Sending *changes* may help with the connection bandwidth problem.
- Micron Automata (in progress with Aluru, Roy, and Srivatsava)
  - Hardware implementation of non-deterministic finite automata
  - Can be adapted to tackle graph problems!

So... Now what?

- Maintain these communities / subgraphs on or near *accelerators*!
- Sending *changes* may help with the connection bandwidth problem.
- Others?
  - Examining FPGA + HMC combinations to move closer to memory (with Young).
  - Interest in others?

Closing

Future directions

- Of course, continue developing streaming / dynamic / incremental algorithms.
    - For massive graphs, computing small changes is always a win.
    - Improving approximations or replacing expensive metrics like betweenness centrality would be great.
- Include more external and semantic data.
    - If vertices are documents or data records, many more measures of *similarity*.
    - Only now being exploited in concert with static graph algorithms.

STINGER represents only some approaches! There are others.

HPC Lab People

Faculty:

- David A. Bader
- Jason Riedy
- Oded Green*

Included here:

- Chunxing Lin
- Eisha Nathan
- Anita Zakrzewska

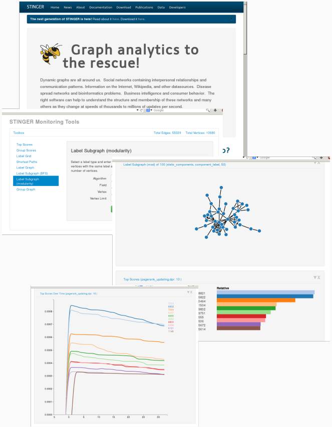
STINGER:

- Robert McColl,
- James Fairbanks* (GTRI),
- Adam McLaughlin*,
- David Ediger* (GTRI),
- Jason Poovey (GTRI),
- Daniel Henderson†,
- Karl Jiang†, and
- *feedback from users in industry, government, academia*

Support: DoD, DoE, NSF, Intel, IBM, Oracle, NVIDIA

∗ Ph.D. related to STINGER. † Other previous students.

STINGER: Where do you get it?



Home: www.cc.gatech.edu/stinger/

Code: git.cc.gatech.edu/git/project/stinger.git/

Gateway to

- code,
- development,
- documentation,
- presentations...

Remember: Academic code, but maturing with contributions.

Users / contributors / questioners: Georgia Tech, PNNL, CMU, Berkeley, Intel, Cray, NVIDIA, IBM, Federal Government, Ionic Security, Citi, Accenture, ...