Towards Efficient Query Processing on Massive Time-Evolving Graphs

Arash Fard, Amir Abdolrashidi, Lakshmish Ramaswamy, John A. Miller

Department of Computer Science

The University of Georgia

International Workshop on
Collaborative Big Data (C-Big 2012)

Introduction

- Dynamic number of nodes and edges in many emerging applications, for example:
  - Hyperlink structure of the World Wide Web
  - Relationship structures in online social networks
  - Connectivity structures of the Internet and overlays
  - Communication flow networks among individuals
- Time-Evolving Graph or TEG:
  - A sequence of snapshots of a graph as it evolves over the time

Need New Approaches for TEG

In contrast to middle size and static graphs:

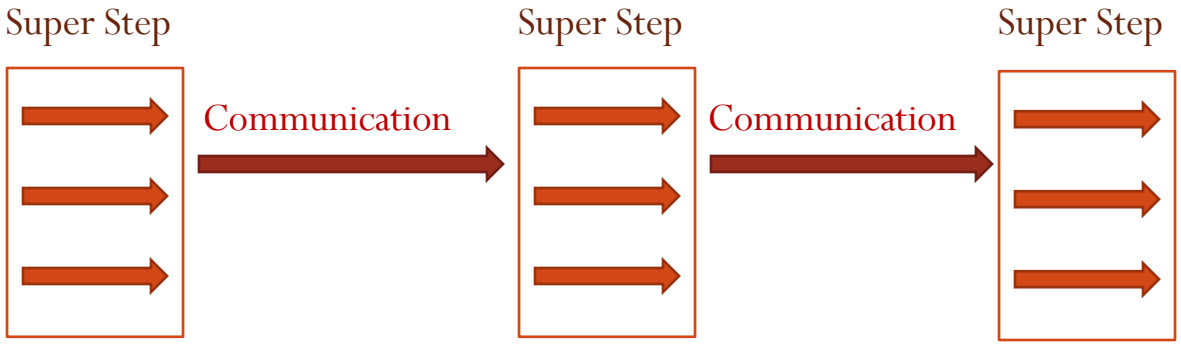
1. An additional dimension, namely *time*
2. Huge size in many modern domains
   - Facebook has about 800 million vertices and 104 billion edges
3. The additional temporal dimension causes the data size to increase by multiple orders of magnitude.

We study three important problems about TEGs:

- Distribution on Cluster Computers
- Reachability Query
- Pattern Matching

BSP model and Vertex-centric graph processing

- BSP (Bulk Synchronous Parallel) model

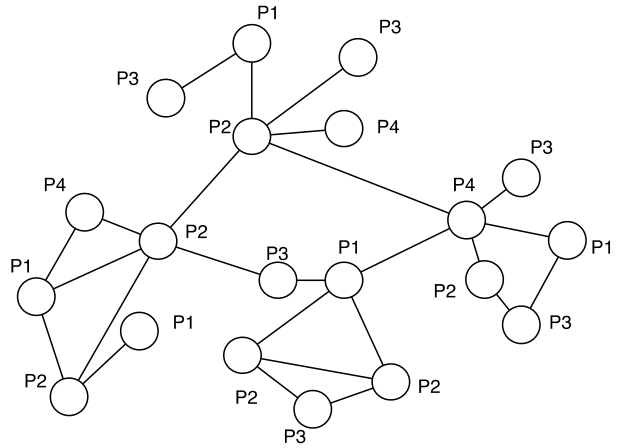


- Vertex-centric graph processing
  - Each vertex of the data graph is a computing unit.
  - Each vertex initially just knows its own label and its outgoing edges.
  - Pregel, Giraph Apache, GPS

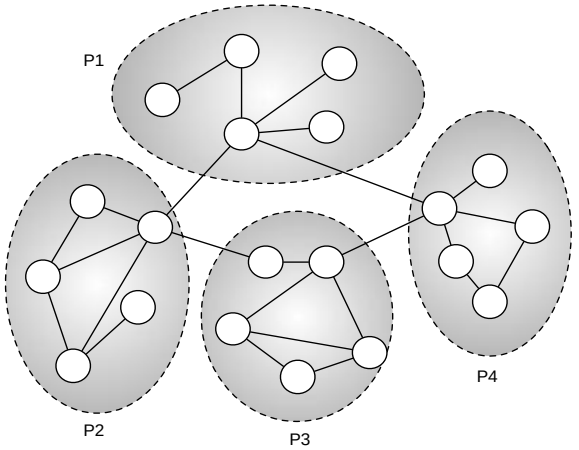
M. Felice Pace, *BSP vs MapReduce*. Proceedings of the 12th International Conference on Computational Science (ICCS '12)

TEG distribution on Clusters

- two contradictory goals:
  - Minimizing communication cost among the nodes of the cluster.
  - Maximizing node utilization.
- A trade-off between two extremes:
  - Assigning the vertices randomly
  - Partitioning the graph into connected components



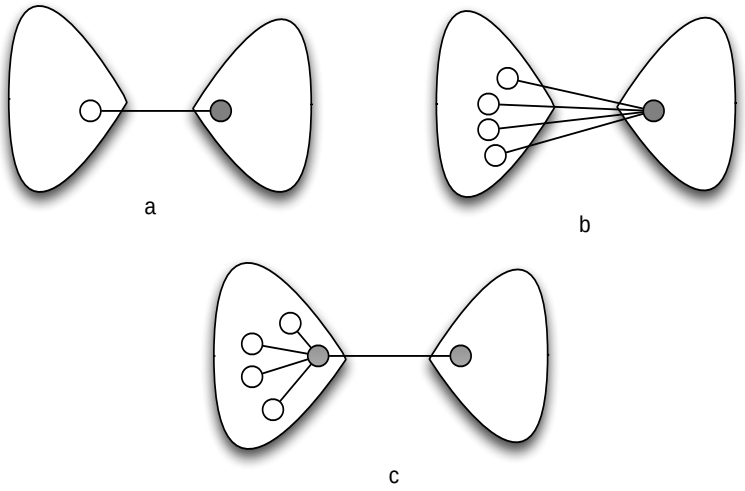
Random assignment of graph vertices



Assignment of vertices based on a partitioning pattern

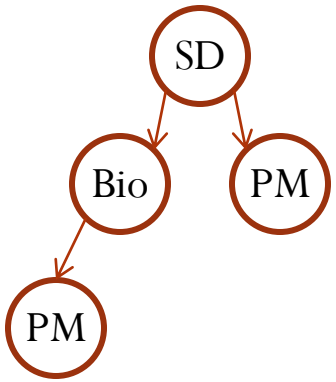
TEG distribution on Clusters

- More partitions than the number of the compute nodes
- Dynamic repartitioning of sub-graphs when changes pass a certain threshold related to the connectivity and structure of the sub-graphs
- Incremental reallocation of a node in order to reduce the communication cost

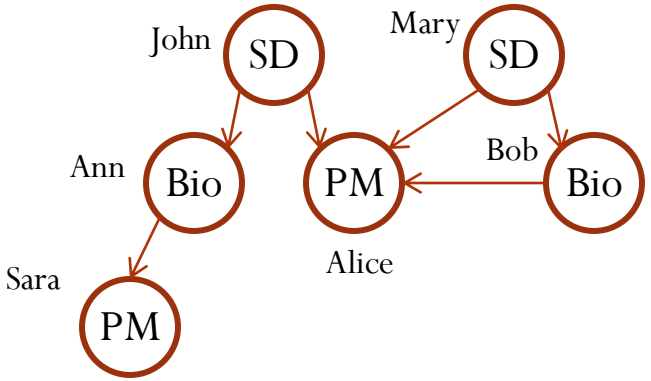


Pattern Matching

- There are different paradigms for pattern matching:
  - Sub-graph Isomorphism (NP-Complete)
  - Graph Simulation (Quadratic)
  - Dual Simulation (Cubic)
  - Strong Simulation (Cubic)



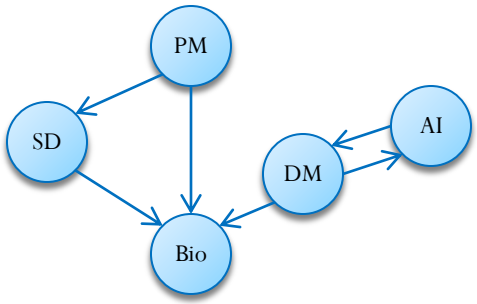
Pattern Graph



Data Graph

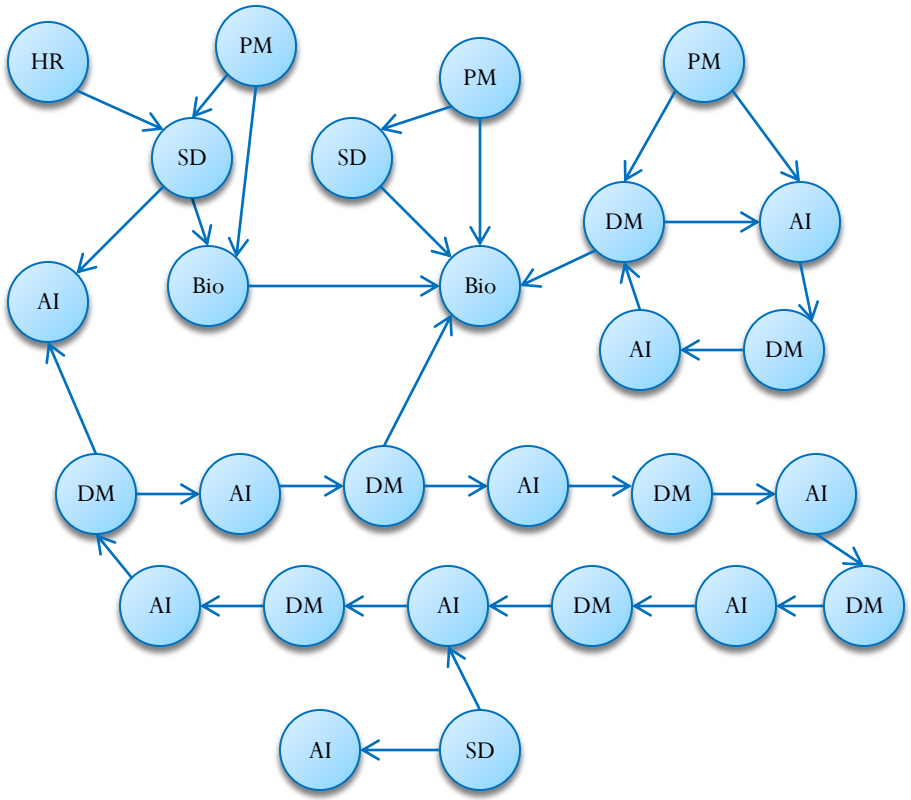
PM: Product Manager
SD: Software Developer
Bio: Biologist

Graph Simulation



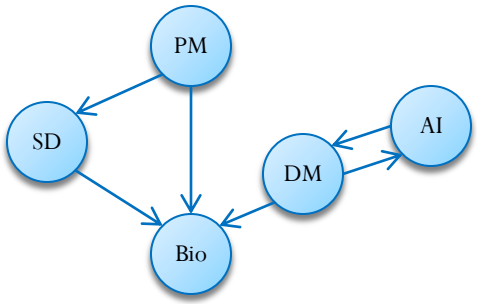
Pattern Graph

PM: Product Manager
SD: Software Developer
Bio: Biologist
DM: Data Mining specialist
AI: Artificial Intelligent specialist
HR: Human Resource



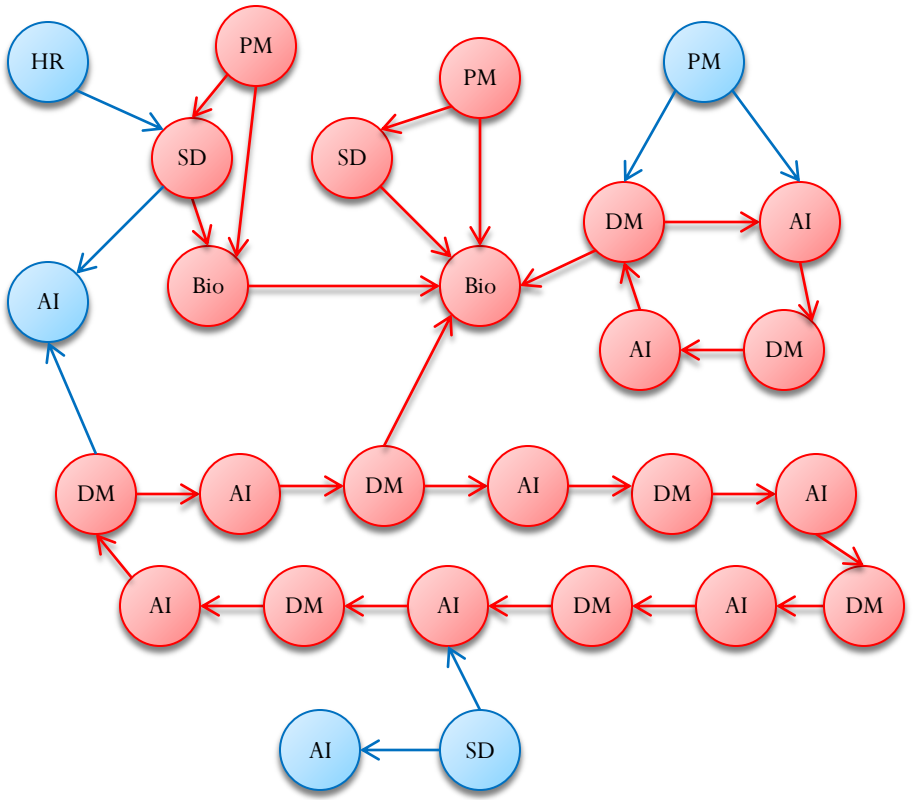
Data Graph

Graph Simulation



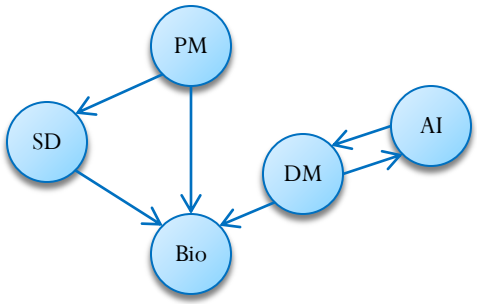
Pattern Graph

PM: Product Manager
SD: Software Developer
Bio: Biologist
DM: Data Mining specialist
AI: Artificial Intelligent specialist
HR: Human Resource



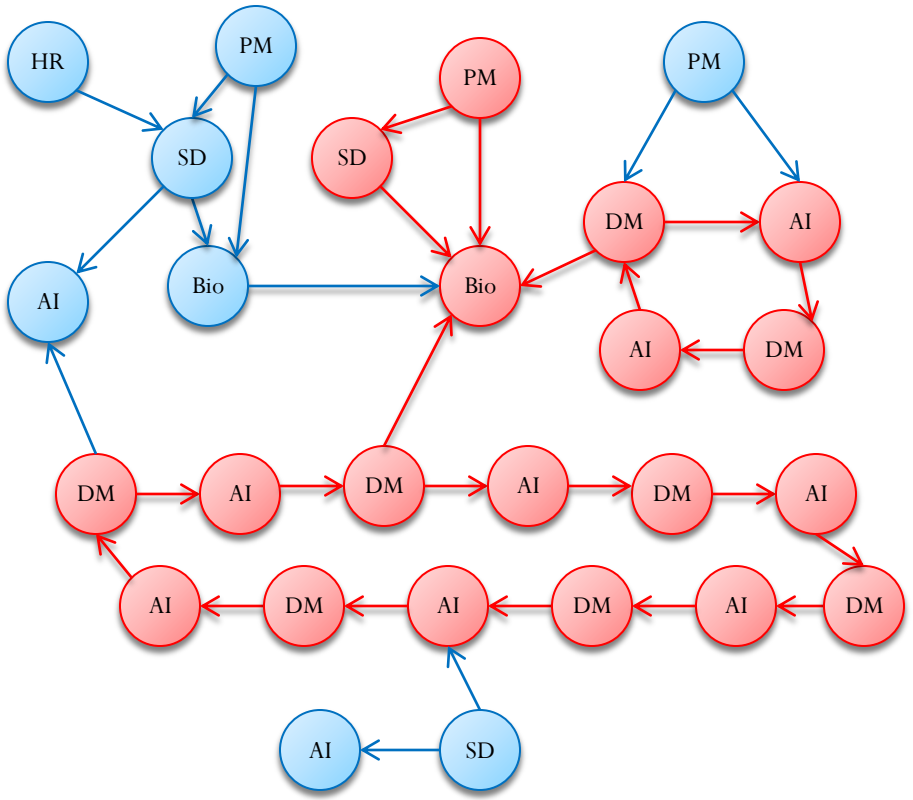
Data Graph

Graph Dual Simulation



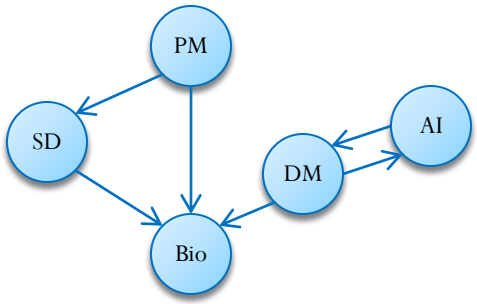
Pattern Graph

PM: Product Manager
SD: Software Developer
Bio: Biologist
DM: Data Mining specialist
AI: Artificial Intelligent specialist
HR: Human Resource



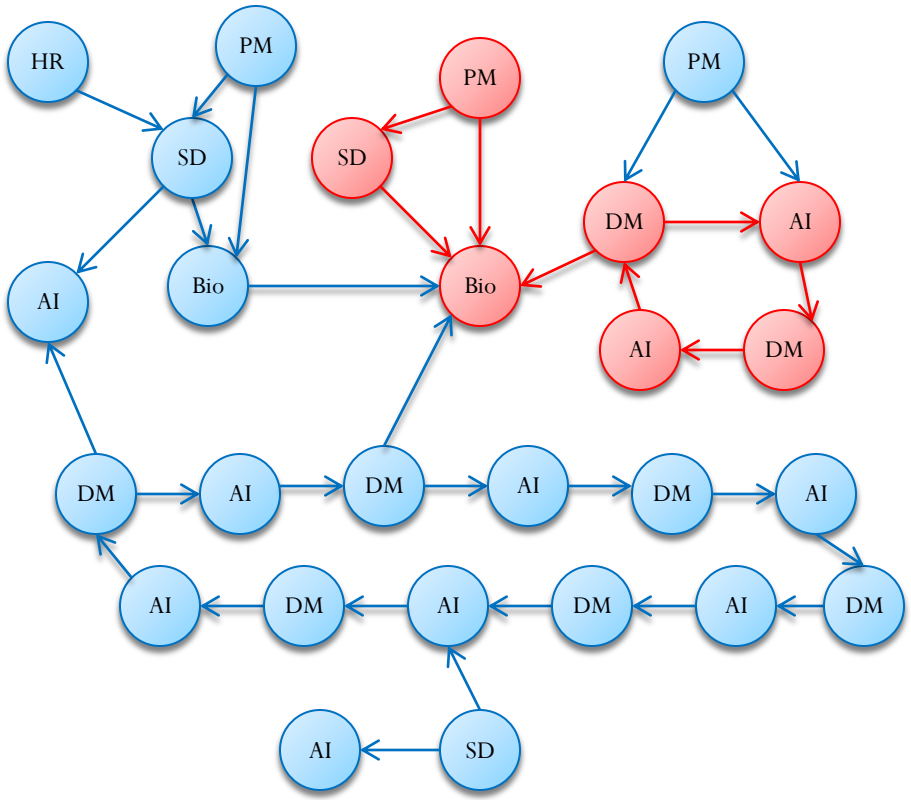
Data Graph

Graph Strong Simulation



Pattern Graph

PM: Product Manager
SD: Software Developer
Bio: Biologist
DM: Data Mining specialist
AI: Artificial Intelligent specialist
HR: Human Resource

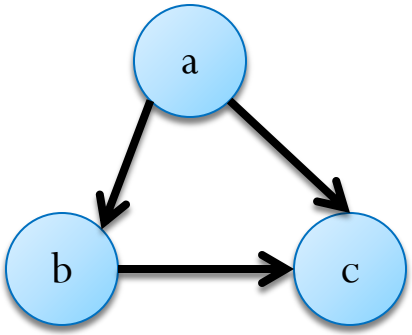


Data Graph

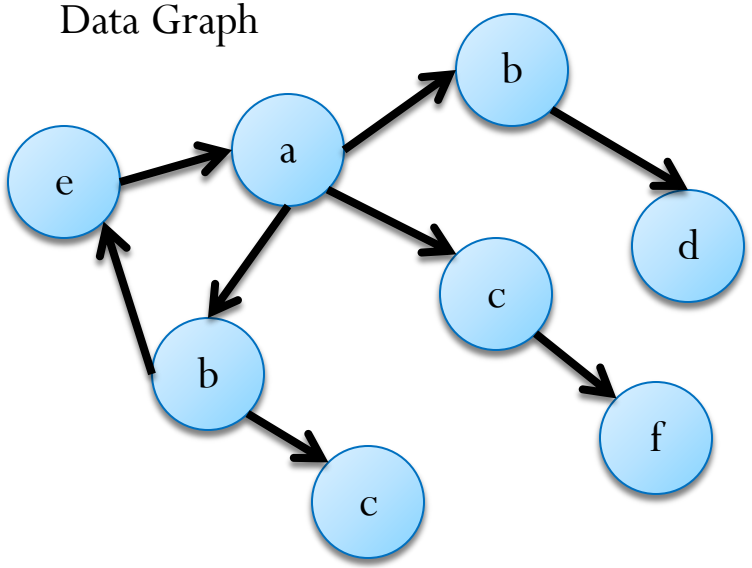
Distributed Graph simulation

Initial Distributed Graph

Query Graph



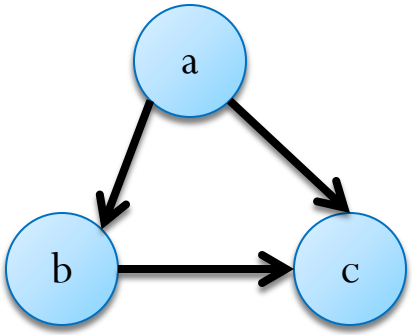
Data Graph



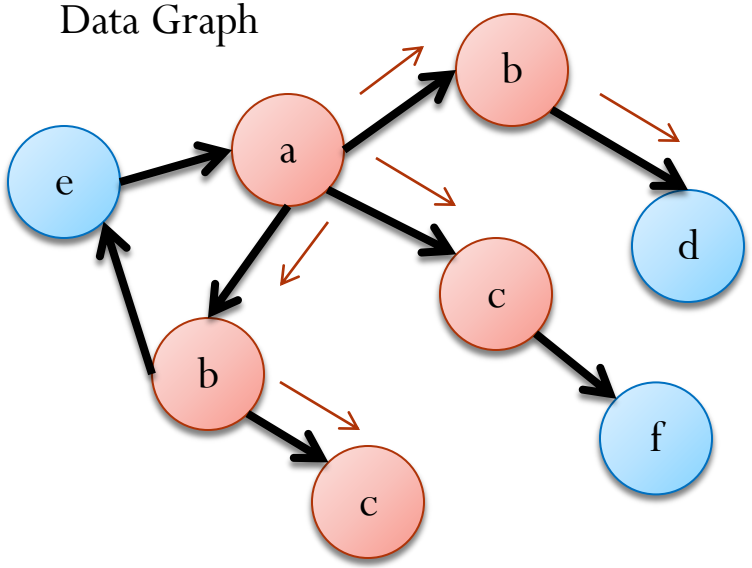
Distributed Graph simulation

The First Superstep

Query Graph



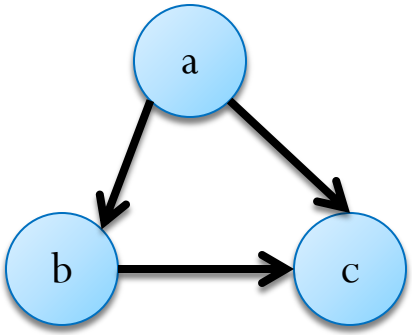
Data Graph



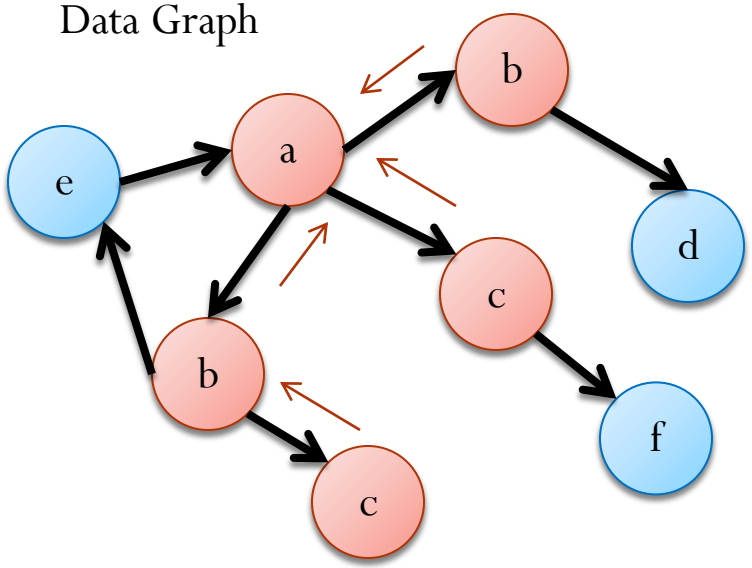
Distributed Graph simulation

The Second Superstep

Query Graph



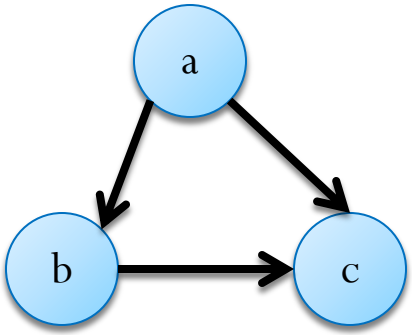
Data Graph



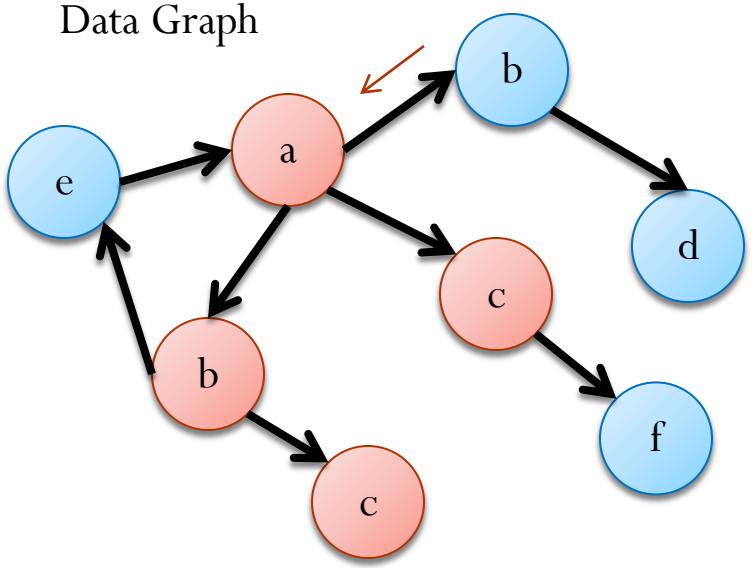
Distributed Graph simulation

The Third Superstep

Query Graph

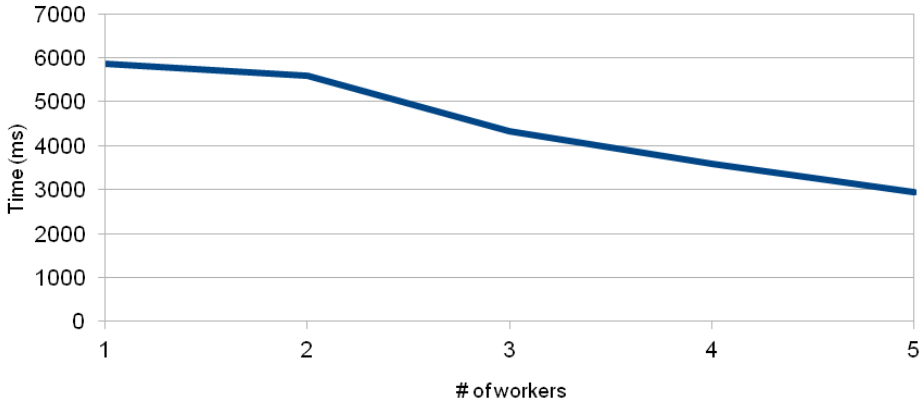


Data Graph



Preliminary results

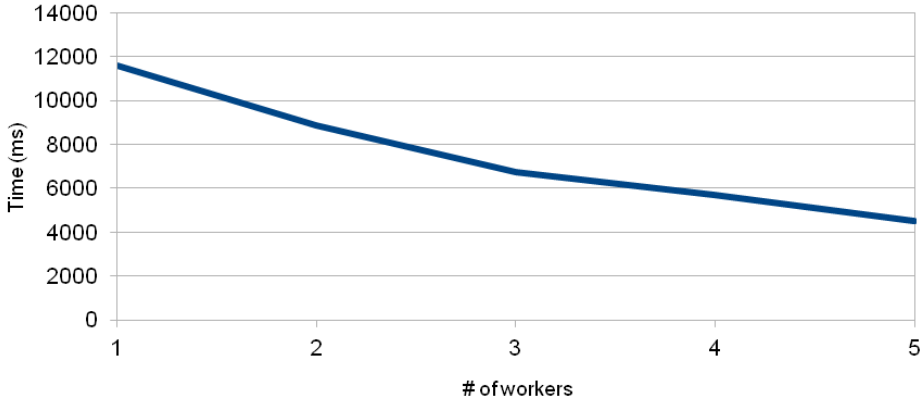
LiveJournal
4.8 million vertices, 69 million edges



Source of the graph:
http://snap.stanford.edu/data/

Number of vertices in the pattern: 20

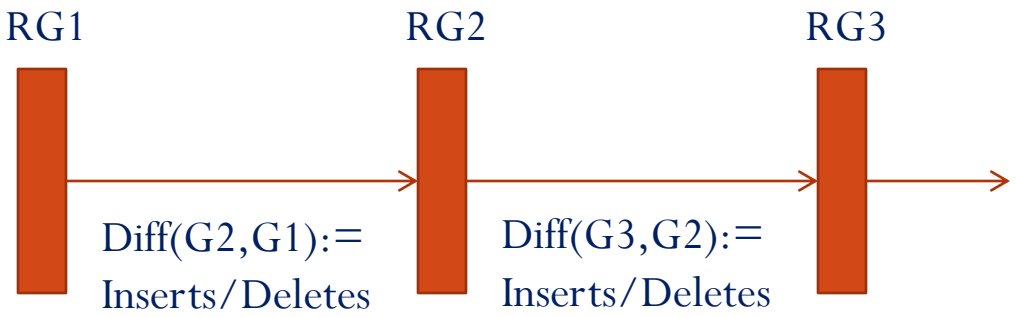
Synthesized data set
10 million vertices, 65 million edges



Graph Synthesizer:
http://projects.skewed.de/graph-tool/

Pattern Matching in TEGs

- We borrow the idea of result graphs from [1].
- Lists for requests of insert and delete, and time stamps for snapshots of the graph.
- Delete commands can only diminish the result graph
- Insert commands will expand previous result graph.
- Saving Result Graphs for some of the snapshots of the graph



[1] W. Fan, J. Li, J. Luo, Z. Tan, X. Wang, and Y. Wu, "Incremental graph pattern matching," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, ser. SIGMOD '11*. New York, NY, USA: ACM, 2011, pp. 925–936.

Reference

- Grzegorz Malewicz, Matthew H. Austern, Aart J.C Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. 2010. Pregel: a system for large-scale graph processing. In*Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*(SIGMOD '10).
- "Giraph website," http://giraph.apache.org/.
- S. Salihoglu and J. Widom, "Gps: A graph processing system," Stanford University, Technical Report, 2012.
- S. Ma, Y. Cao, J. Huai, and T. Wo, "Distributed graph pattern matching," in *Proceedings of the 21st international conference on World Wide Web, ser. WWW '12*.
- W. Fan, J. Li, J. Luo, Z. Tan, X. Wang, and Y. Wu, "Incremental graph pattern matching," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, ser. SIGMOD '11*.
- S. Ma, Y. Cao, W. Fan, J. Huai, and T. Wo, "Capturing topology in graph pattern matching," *Proc. VLDB Endow.*, vol. 5, no. 4, pp. 310–321, Dec. 2011.
- M. R. Henzinger, T. A. Henzinger, and P. W. Kopke, "Computing simulations on finite and infinite graphs," in *Proceedings of the 36th Annual Symposium on Foundations of Computer Science, ser. FOCS '95*.