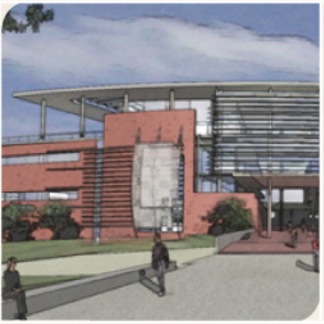
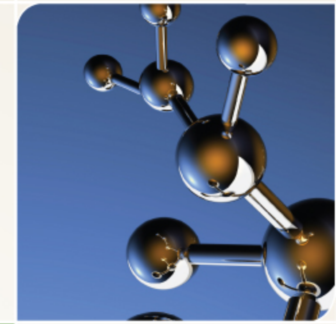
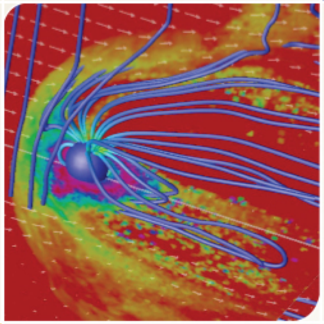
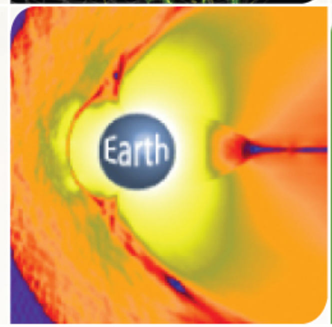




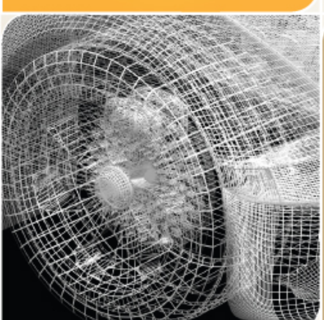




A Performance Evaluation of Open Source Graph Databases

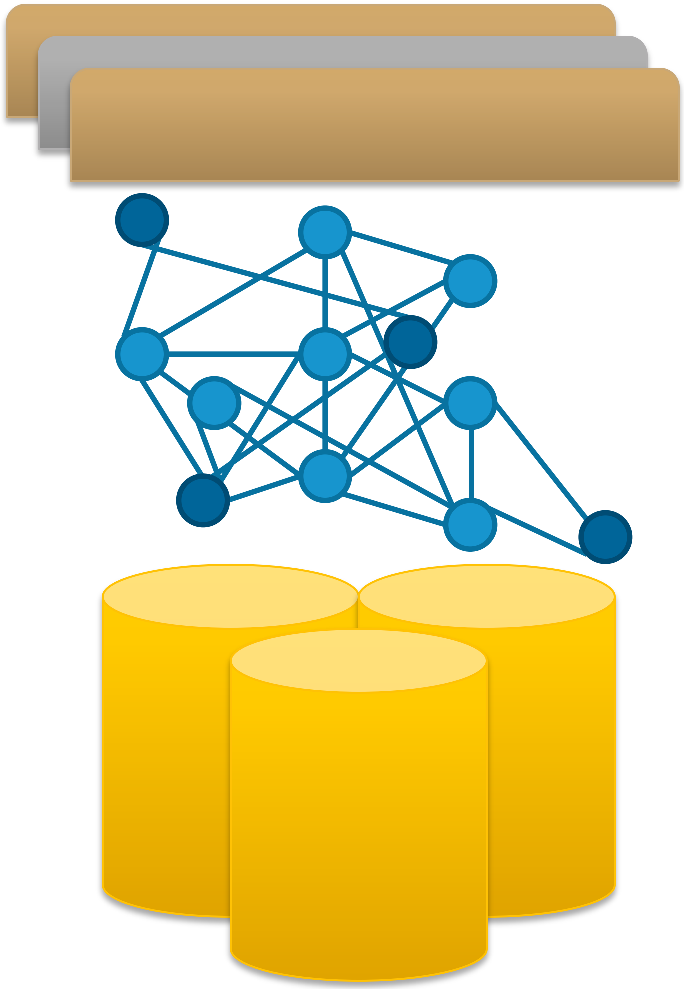
Robert McColl, David Ediger
Jason Poovey, Dan Campbell, David A. Bader

Georgia Tech
College of Computing
Computational Science and Engineering



Overview

- Motivation
- Options
- Evaluation
- Results
- Lessons Learned
- Moving Forward



Massive Streaming Semantic Graphs

Features

- Millions to billions of vertices and edges with rich semantic information (name, type, weight, time), possibly missing or inconsistent data
- Thousands to millions of updates per second
- Power-law degree distribution, sparse (d(v) ~ O(1)), low diameter

Financial

- NYSE processes >2TB daily, maintains 10's of PB

Social

- 50,000+ Facebook Likes per second, 1.2B users
- 6,000 Twitter Tweets per second, 500M users

Google

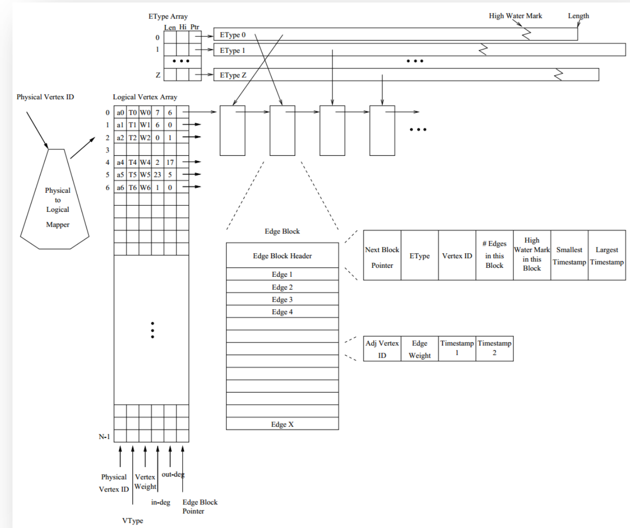
- "Several dozen" 1PB data sets
- Knowledge Graph: 570M entities, 18B relationships

Business

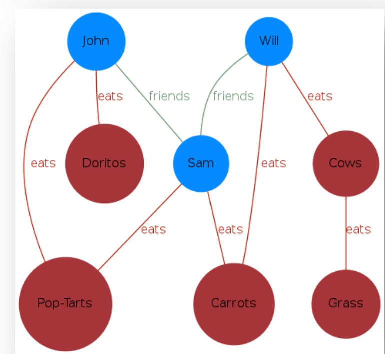
- eBay: >17 trillion records, 5B new records per day

Given these graphs…

- How do we store them?
  - Memory vs. Disk
  - ACID vs. loose consistency
  - Single node vs. distributed
  - Semantic and temporal information
  - Numeric IDs



- How do we query them?
  - Simple neighbor queries
  - Traversal methods, filtering
  - Programming models vs. query languages



- How do we compute meaningful answers?
  - Algorithms and implementation paradigms
  - BSP, MapReduce, MPI, OpenMP, single threaded
  - C, C++, Python, Java, Scala, Javascript, SQL, Gremlin

The Options

	Owner / Maintainer	License	Platform	Language	Distribution	Cost	Transactional	Memory-Based	Disk-Based	Single-Node	Distributed	Graph Algorithms	Text-Based Query Language	Embeddable	Software	Data Store	Type
MySQL	Oracle	GPL/Proprietary	x86	C/C++	Bin	Free	X	-	X	X	-	-	SQL	X	X	X	SQLDB
Oracle	Oracle	Proprietary	x86	C/C++	Bin	$180-$950	X	-	X	X	X	-	SQL	X	X	X	SQLDB
SQL Server	Microsoft	Proprietary	x86-Win	C++	Bin	$898-$8592	X	-	X	X	-	-	SQL	-	X	X	SQLDB
SQLite	Richard Hipp	Public Domain	x86	C	Src/Bin	Free	X	X	X	X	-	-	SQL	X	X	X	SQLDB
AllegroGraph	Franz, Inc.	Proprietary	x86	Likely Java	Bin	Free-ish/$$$$	X	-	X	X	-	X	SPARQL,RDFS++,Prolog	-	X	X	GDB
ArangoDB	ArangoDB	Apache	x86	C/C++/JS	Src/Bin	Free	-	-	X	X	-	-	AQL	-	X	X	GDB/KV/DOC
DEX	Sparsity-Technologies	Proprietary	x86	C++	Bin	Free Personal/Commercial $$	X	-	X	X	-	X	Traversal	X	-	X	GDB
FlockDB	Twitter	Apache	x86	Java,Scala,Ruby	Src	Free	-	-	X	X	X	-	-	-	X	X	GDB
GraphBase	FactNexus	Proprietary	x86	Java	Bin	Free,$15/mo,$20,000	?	-	X	X	-	-	Bounds	X	X	X	GDB
HyperGraphDB	Kobrix Software	LGPL	x86	Java	Src	Free	MVCC	X	X	X	X	-	HGQuery,Traversal	X	-	-	HyperGDB
InfiniteGraph	Objectivity	Proprietary	x86	Java/C++	Bin	Free Trial/$5,000	Both	-	X	X	X	-	Gremlin	X	X	X	GDB
InfoGrid	Johannes Ernst	AGPL/Proprietary	x86	Java	Src/Bin	Free + Support	-	X	X	X	X	-	-	X	X	-	GDB
Neo4j	Neo Technology	GPL/Proprietary	x86	Java	Src/Bin	Free,$6,000-$24,000	X	-	X	X	-	X	Cypher	X	X	X	GDB/NoSQL
OrientDB	NuvolaBase Ltd	Apache	x86	Java	Src/Bin	Free + Support	Both	X	X	X	X	-	Extended SQL, Gremlin	X	X	X	GDB/NoSQL
Titan	Aurelius	Apache	x86	Java	Src/Bin	Free + Support	Both	-	X	X	X	-	Gremlin	X	X	-	GDB
Bagel	UC Berkley	BSD	x86	Java/Scala/Spark	Src	Free	-	X	-	X	X	X	-	X	-	-	BSP
BGL	Boost / IU	Boost	x86 / C++	C++	Src/Bin	Free	-	X	-	X	-	X	-	X	-	-	Library
Faunus	Aurelius	Apache	x86	Java	Src	Free + Support	Both	-	X	X	X	-	Gremlin	X	X	-	Hadoop
Gephi	Gephi Consortium	GPL/CDDL	x86	Java,OpenGL	Src/Bin	Free	-	X	-	X	-	X	-	X	X	-	Toolkit
Giraph	Apache	Apache	x86	Java	Src	Free	-	X	-	X	X	X	-	X	-	-	BSP
GraphStream	University Le Havre	LGPL/CeCILL-C	x86	Java	Src/Bin	Free	-	X	-	X	-	X	-	X	-	-	Library
Hama	Apache	Apache	x86	Java	Src	Free	-	X	-	X	X	X	-	X	-	-	BSP
MTGL	Sandia NL	BSD	XMT	C++	Src	Free	-	X	-	X	-	X	-	X	-	-	Library
NetworkX	Los Alamos NL	BSD	x86	Python	Src/Bin	Free	-	X	-	X	-	X	-	X	-	-	Library
PEGASUS	CMU	Apache	x86	Java	Src/Bin	Free	-	-	X	X	X	X	-	-	X	-	Hadoop
STINGER	GT / GTRI	BSD	x86*/XMT	C	Src	Free	-	X	-	X	-	X	-	X	X	X	Library
uRika	Yarc Data / Cray	Proprietary	XMT	Likely C++	Bin	$$$$	?	X	-	X	-	-	SPARQL	-	X	X	Appliance

* see page 2 of the paper

* DEX is now Sparksee

Evaluation

- Four fundamental graph kernels
  - Prefer a particular implementation
  - Emphasize common programming / traversal styles
  - Measure performance in multiple ways
    - Time to completion
    - Memory in use (MemoryMXBean for Java, OS reporting otherwise)
    - Qualitative analysis of capabilities and development experience
  - If not provided by package, written by naïve programmer

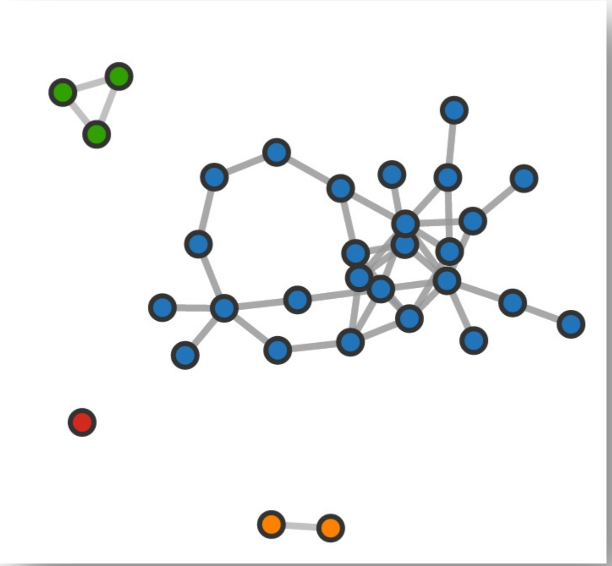
- Same R-MAT graphs used with all packages
  - Vertices: 1K (tiny), 32K (small), 1M (medium), and 16M (large)
  - Edges: 8K (tiny), 256K (small), 8M (medium), 128M (large)
  - Graphs of 1B edges considered, but not included due to number of systems that couldn't work with large

Evaluation

- Single Source Shortest Paths (SSSP)
  - Must be a breadth-first traversal of the graph
  - Level-synchronous parallel where possible
  - Output unweighted distances from source to all vertices

- Application
  - Used for routing and connectivity
  - Building block for other algorithms (e.g. betweenness centrality)
  - Graph 500 benchmark

Evaluation



- Connected Components
  - Based on Shiloach-Vishkin
  - Edge-parallel label-pushing style
  - Global graph metric that touches all edges in the graph
- Properties
  - Not theoretically work efficient, but edge-parallel pattern is representative of a broad class of graph algorithms
  - Read-heavy with sparse writes
  - Cache friendly graph traversal with cache-unfriendly random label read / assignment

Evaluation

- PageRank centrality algorithm
  - Vertex-parallel Bulk Synchronous Parallel (BSP) power-iteration style
  - Measure of influence over information flow and importance in the network
  - Representative of a broad class of iterative graph algorithms

- Update benchmark
  - Perform edge insertions and deletions in parallel
  - Random access and modification of the structure
  - Real-world networks are in constant motion as new edges and vertices enter the graph

DISCLAIMERS

- We are not expert Java programmers, Hadoop tuners/ magicians, Python gurus (maybe a few), Scala wizards
- We are some of the primary maintainers of the STINGER graph package
- The code for these implementations is available at github.com/robmccoll/graphdb-testing
  - Test code is BSD, package licenses vary
  - We apologize if you feel that your particular work has been slighted or misrepresented
  - If any of these packages belong to you, we invite and encourage you to improve on our results and submit them back to the repository

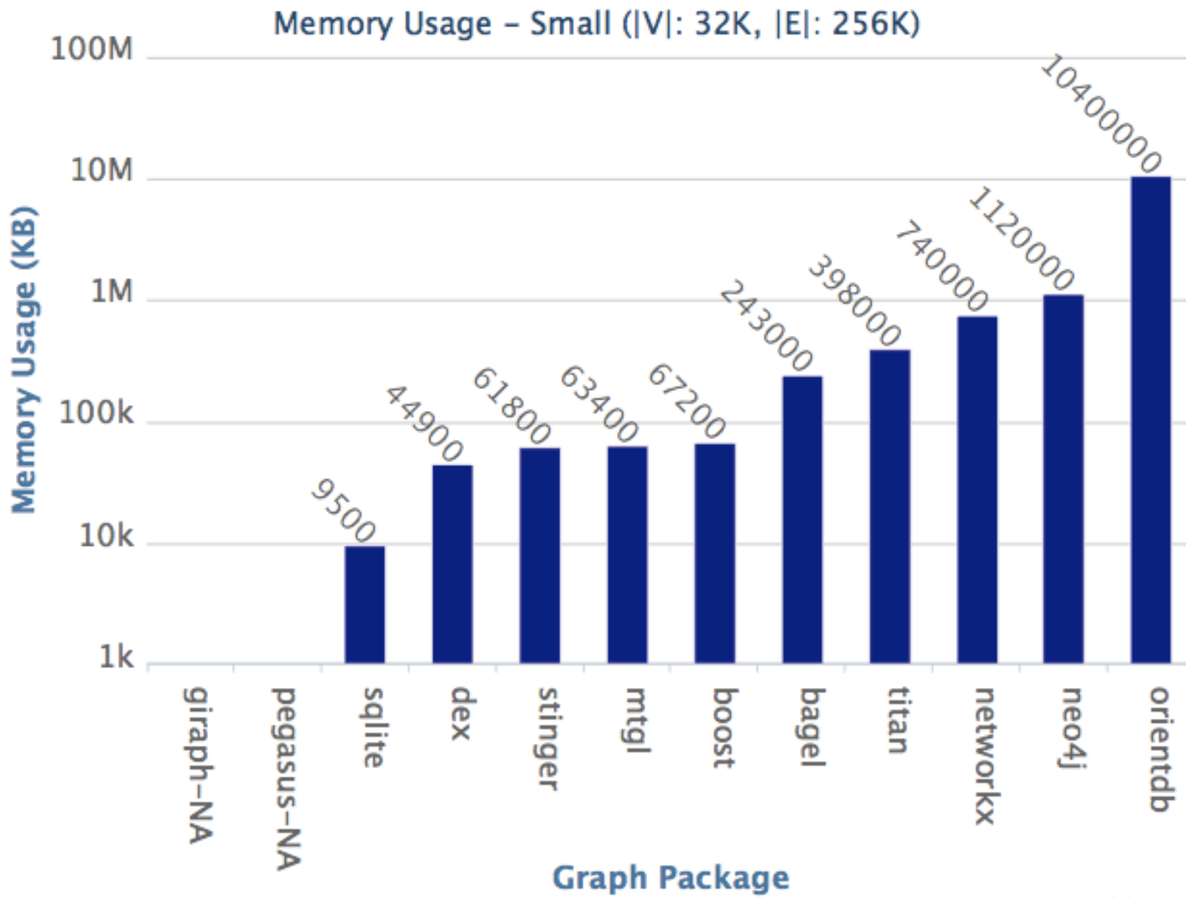
Maximum Size Completed

Package	Tiny	Small	Medium	Large
boost	X	X	X	X
dex	X	X		
mtgl	X	X	X	X
neo4j	X	X		
networkx	X	X	X	
orientdb	X	X	X	
sqlite	X	X		
stinger	X	X	X	X
titan	X	X	X	
bagel	X	X	X	
pegasus	X	X	X	X
giraph	X	X	X	X

Determined by crashing, running out of memory, or running an algorithm for longer than 24 hours.
Exception: Dex had a 1 million element (|V| + |E|) limit.

Memory Usage

3 orders of magnitude difference in memory usage

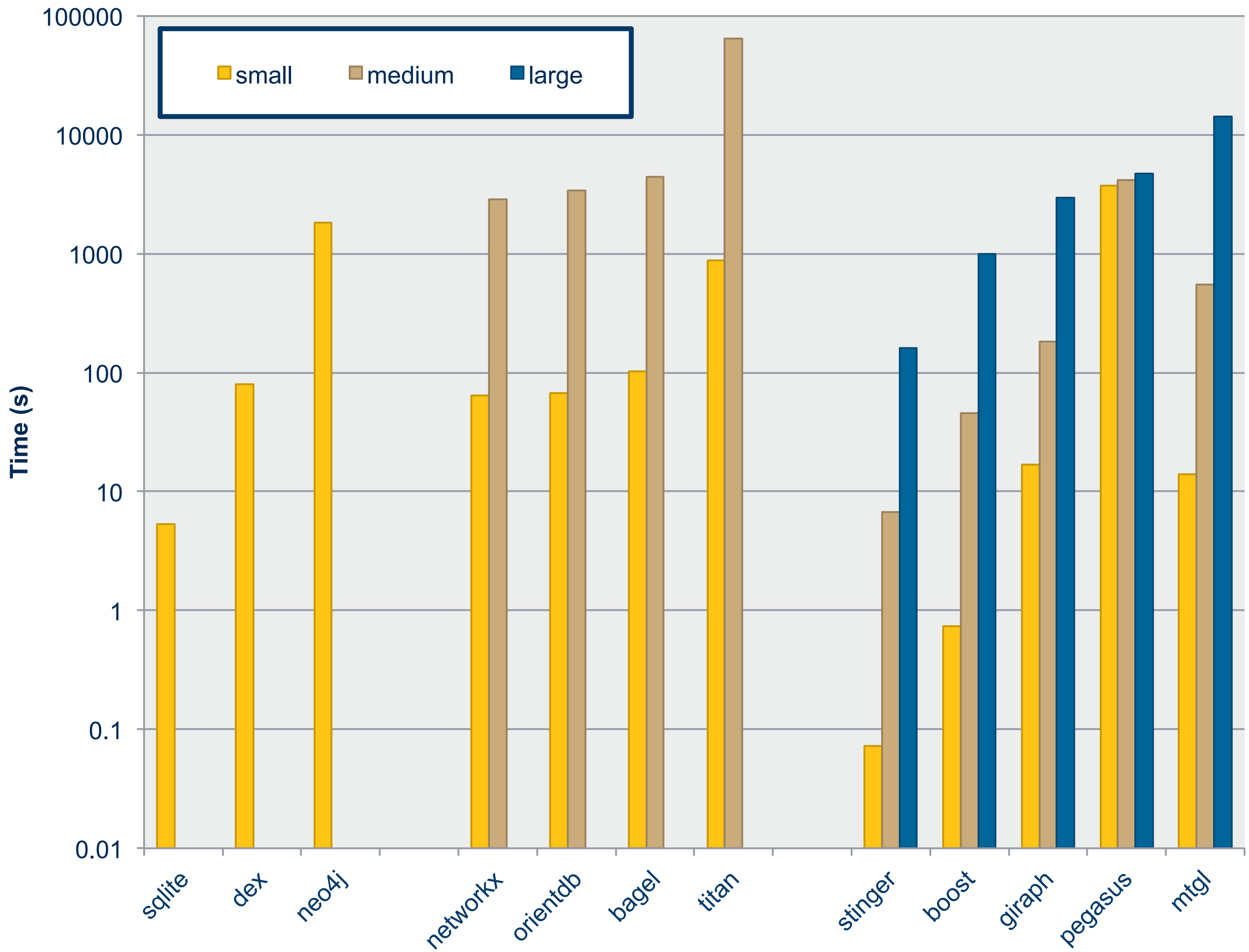


Page Rank Performance

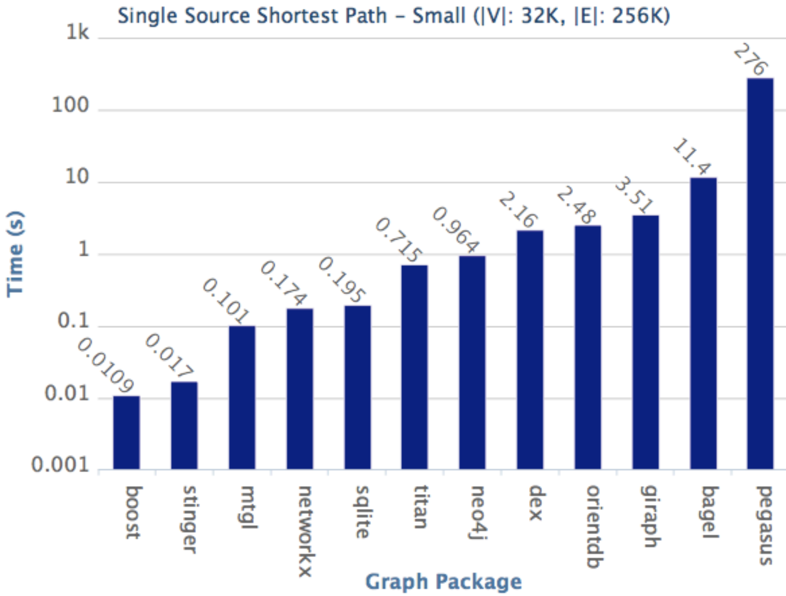
Small
Range 10,000x

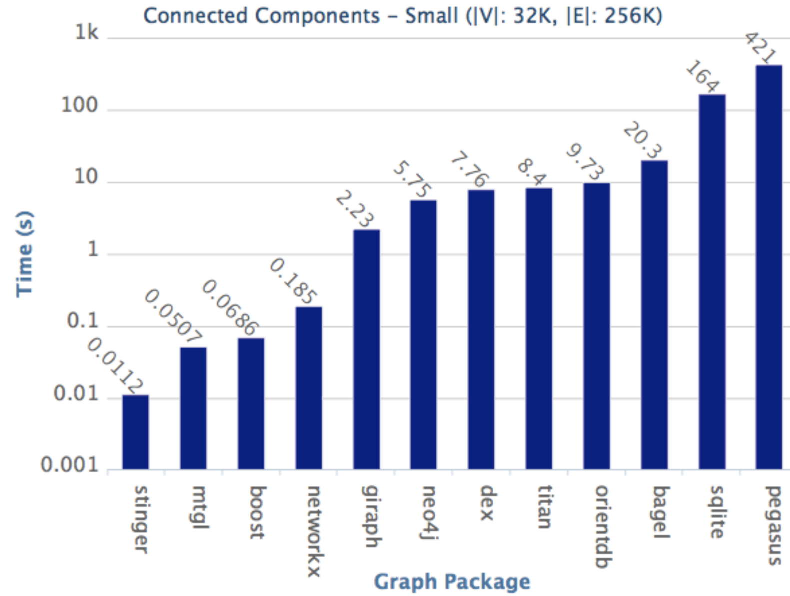
Medium (x32)
Range 10,000x
40~90x slower
Except Pegasus, Giraph

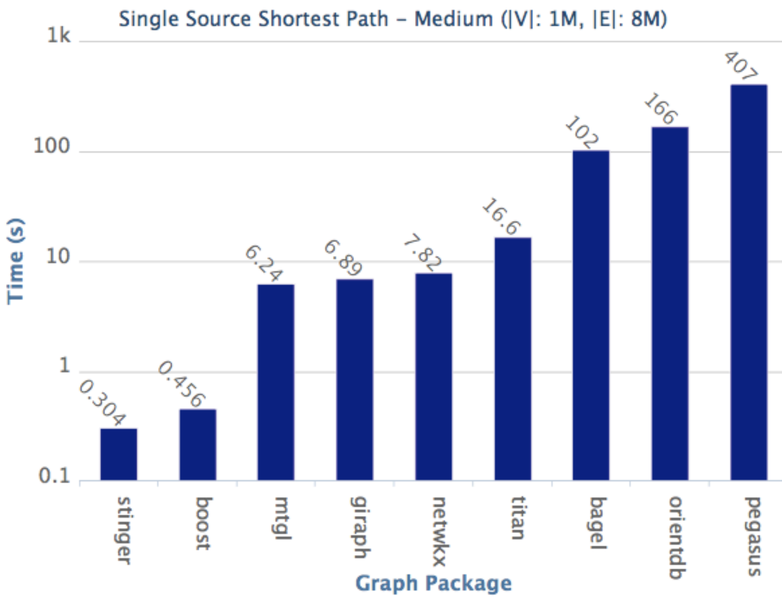
Large (x1024)
Range 100x
~20x slower
Except Pegasus

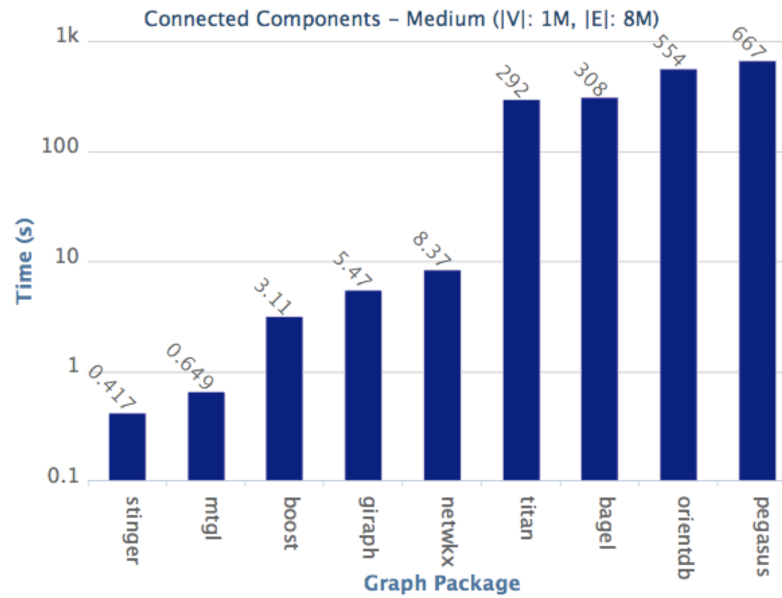


SSSP and Components Small vs. Medium









Observations on Quantitative

- Big performance gaps, possible groups
  - High Performance (ms response)
  - Medium Performance (sub minute response)
  - Storage Performance
    - Perhaps suited for storage, but not analysis
    - Not language or system, just not written with graph algorithms in mind
- Some parallel, but many not
  - Graph libraries generally parallel
  - Not much parallelism in (single node) disk-backed DBs
    - Parallelism only increases random read and write
    - Bottleneck is disk bandwidth and latency
- Data scaling is hard to predict
- Full results available at:
  - github.com/robmccoll/graphdb-testing
  - arxiv.org/abs/1309.2675

Qualitative Observation

- Multitude of input file formats supported
  - No universal support
  - XML, JSON, CSV, or other proprietary binary and plain-text formats
  - Trade-off between size, descriptiveness, and flexibility
  - Challenge for data exchange
  - Edge-based, delimited, and self-describing easily parsed in parallel, translated to other formats
- Same data, same algorithms, same HW, DB / library is the difference
- No consensus on ACID / Transactions
  - No clear result on its affect on speed
  - Not clear that the applications exist
- No consensus on query languages, models

Moving Forward

- To advance HPC outside of our community, software must be accessible

- Some applications don't compile out of the box
  - Self-contained packages are easiest
  - Package management systems don't work well on nodes inside a cluster
  - Build problems drive users away

- Lack of consistent approaches to documentation
  - Well-written tutorials better than sparse full API doc
  - Lack of adequate usage examples
    - Inserting vertices and edges, querying for neighbors is not enough
    - What is the best way to traverse the structure?
    - How should data ingest, analysis, and extracting results be performed?

  - Highlights
    - NetworkX is very well documented, both examples and API doc
    - Titan's API doc is very complete, but lacks examples of real-world usage
    - Bagel, STINGER lack formal documentation or have fragmented and occasionally inaccurate documentation

Acknowledgment of Support

DARPA

Department of Energy · United States of America

Department of Defense · United States of America

Oak Ridge National Laboratory

Pacific Northwest National Laboratory

NVIDIA.

intel®

CRAY

IBM®