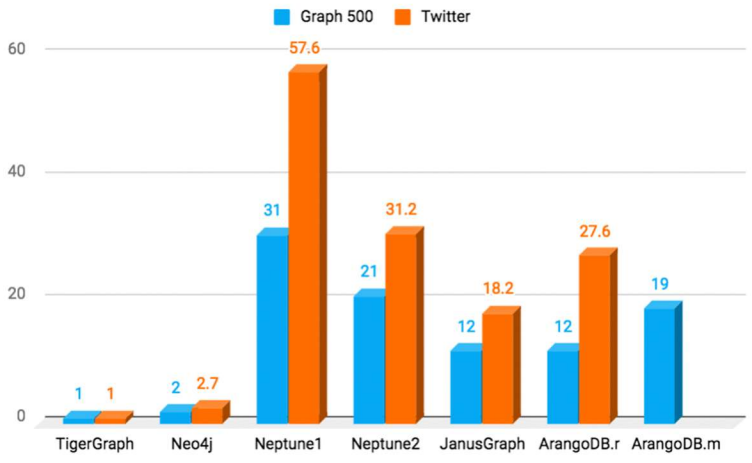
TigerGraph

Benchmarking Graph Analytic Systems:
TigerGraph, Neo4j, Neptune, JanusGraph, and ArangoDB

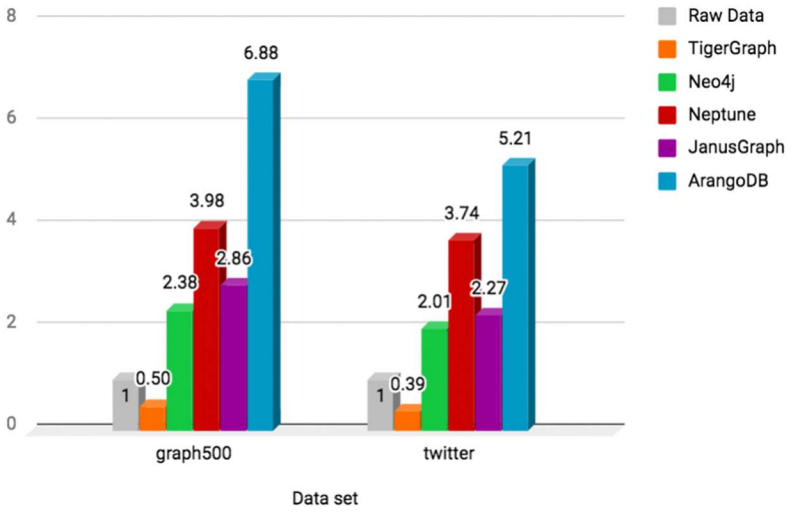
TIGERGRAPH LEAPS PAST COMPETITION

In benchmark tests comparing TigerGraph to Neo4j, Amazon Neptune, JanusGraph and ArangoDB, TigerGraph consistently outperformed all competitors by wide margins. The complete benchmark report is available at www.tigergraph.com/benchmark.

LOADING TIME



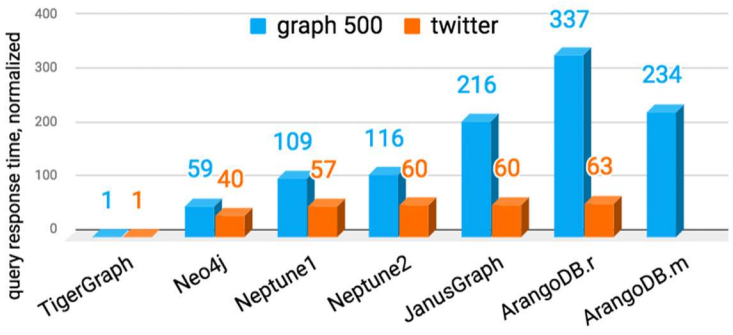
NORMALIZED DATABASE SIZE



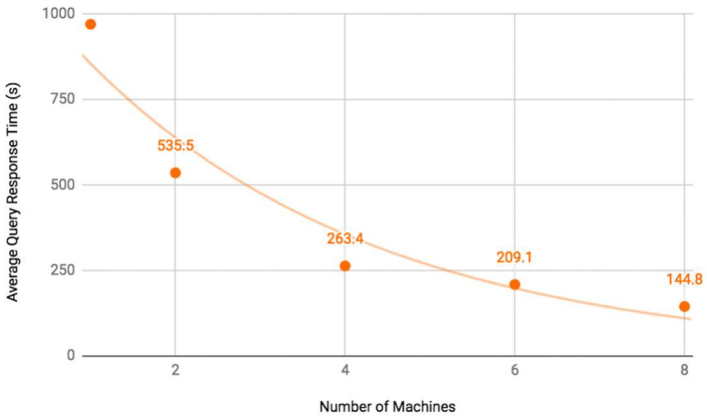
On our test data sets (a synthetic graph from graph500.org and a real-world Twitter graph with 1.47B edges):

- **Data Loading Speed:** TigerGraph loads data 1.8x to 58x faster than competitors with Neo4j having significantly improved their speed for offline loading (where concurrent database operations are not allowed) since last benchmark. Neo4j continues to have slow online loading speed.
- **Normalized Database Size After Loading:**
  - Raw data is compressed and stored in TigerGraph resulting in 50% storage savings for the Graph 500 dataset and 61% storage savings for the Twitter dataset.
  - Other graph databases create graphs that are larger than the raw data, demanding 5x to 13x more storage space than TigerGraph.

TWO-HOP PATH QUERY



TIGERGRAPH QUERY RESPONSE VS NUMBER OF MACHINES



Get the full report with all of the data at www.tigergraph.com/benchmark

Two-Hop Path Query Performance:

TigerGraph is 40x to 337x faster than other graph databases owing to its native massively parallel processing (MPP) graph architecture. You can read more about Native Parallel Graphs at www.tigergraph.com/ebook.

We could not run 3-hop and 6-hop queries for other graph databases, as they returned out-of-memory error or timed out in many cases.

Query response time vs number of machines:

- TigerGraph achieved 6.7x speedup with 8 machines, scaling almost linearly for PageRank, an iterative algorithm which traverses every edge during each iteration and results in a lot communication between the machines, with information being sent from one partition to another.
- We were unable to perform the scalability tests on Neo4j or Amazon Neptune. Neo4j must store the full graph on a single server and cannot partition a graph across multiple machines. Amazon Neptune also cannot partition a graph across multiple machines nor could we find a way to run PageRank.

Graph database/analytics systems:

- TigerGraph Developer Edition 2.1.4
- Neo4j 3.4.4 Community Edition
- Amazon Neptune 1.0.1.0.200233.0
- JanusGraph 0.2.1, with Cassandra as the storage backend
- ArangoDB 3.3.13, with each of two storage engines, MMFiles and RocksDB

Data Sets

Instructions for reproducing our tests:

https://github.com/tigergraph/graph-database-benchmark/tree/master/benchmark

Developer Edition Download: www.tigergraph.com/developer

Name	Description and Source	Vertices	Edges
graph500	Synthetic Kronecker graph http://graph500.org	2.4 M	67M
twitter	Twitter user-follower directed graph http://an.kaist.ac.kr/traces/WWW2010.html	41.6 M	1.47 B