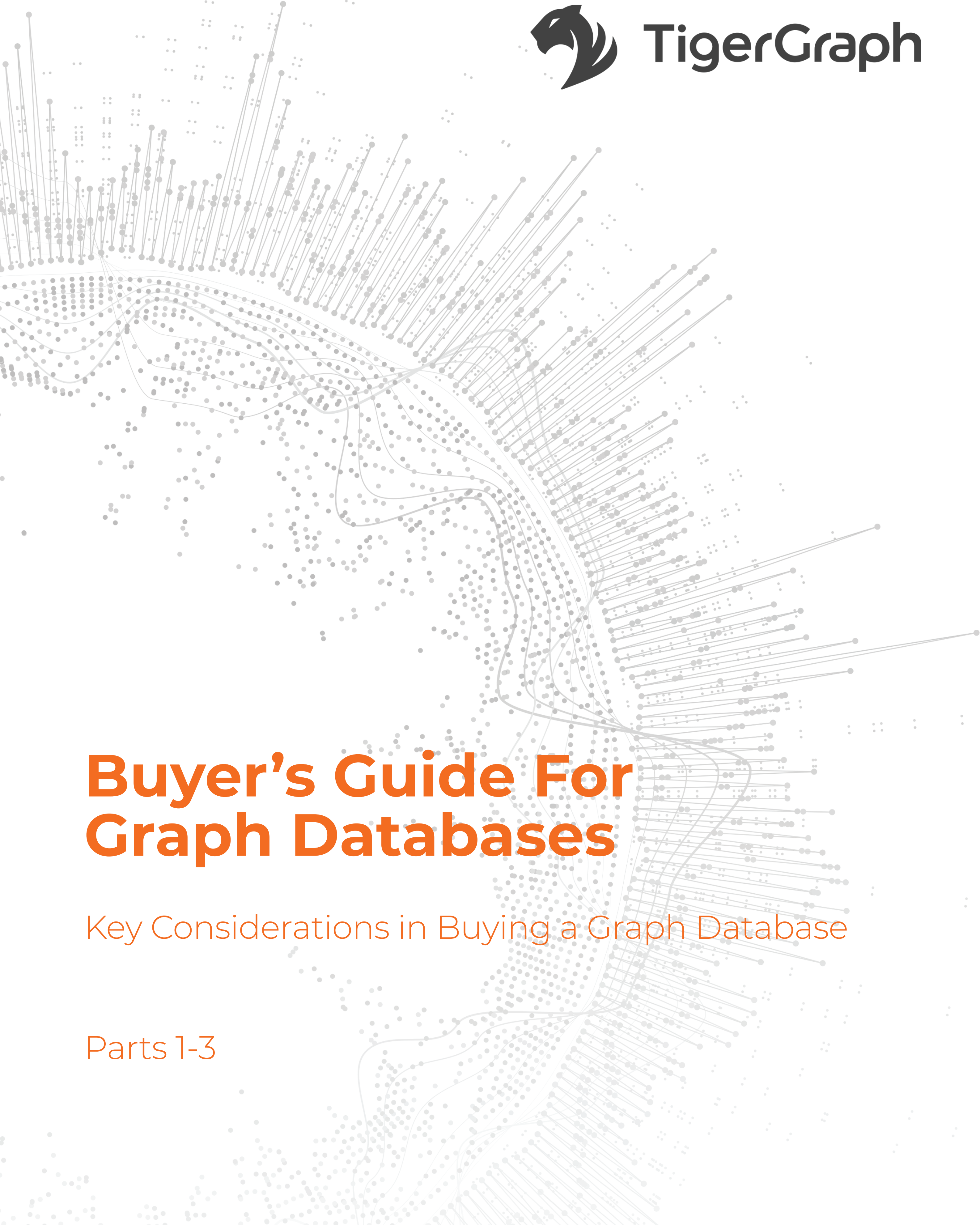
TigerGraph



# Buyer’s Guide For Graph Databases

Key Considerations in Buying a Graph Database

Parts 1-3

# SELECTING A GRAPH DATABASE FOR ON-PREMISE OR CLOUD DEPLOYMENT

Graph databases are the fastest growing category in all of data management. Since seeing early adoption by companies including Facebook, Google and LinkedIn, graph has evolved into a mainstream technology used today by enterprises in every industry across a wide variety of use cases. By organizing data in a graph format, graph databases overcome the big and complex data challenges that other databases such as Relational and NoSQL cannot.

Selecting graph software is an important decision which can shape the success of your organization. Unfortunately buyers often struggle to reconcile the conflicting claims made by different graph software vendors - these claims are often characterized by misinformation.

This guide is intended to assist you in your buying decision by providing a side-by-side comparison of three leading graph databases, Neo4j, Amazon Neptune, DataStax and TigerGraph. The guide is divided into three parts. Part 1 compares Neo4j and TigerGraph, while part 2 compares Amazon Neptune and TigerGraph Cloud, and part 3 compares DataStax and TigerGraph.

**Part 1 - TigerGraph and Neo4j**

**Part 2 - TigerGraph and Amazon Neptune**

**Part 3 - TigerGraph and DataStax**

## 1.1 TigerGraph and Neo4j Comparison Summary

This section provides a high-level comparison of TigerGraph and Neo4j and includes information on speed, scalability, cloud availability, total cost of ownership and more. As you consider the purchase of a graph database, here are the key questions that most buyers work through:

- Where is the graph market going?  How should I prepare?
- Will my selected graph database continue to serve my needs now and into the future as the complexity as well as the volume of my data grows? In other words, is my choice or investment future-proof?
- How easy is it to distribute the data across multiple machines to avoid adding CPUs or RAM to a single, expensive machine?
- What is the total cost of ownership for the selected graph database considering the initial cost of license (on-premises) or subscription (cloud-based service), cost of infrastructure, whether physical hardware or cloud resources, and cost of maintaining and upgrading my solution?

### Graph is Fundamental to Machine Learning, AI and Analytics

Graph is quite common as foundation and enabler in the analytics world. Business people are asking increasingly complex questions across structured and unstructured data - it often requires blending of data from multiple sources, multiple business units, and increasingly external data.

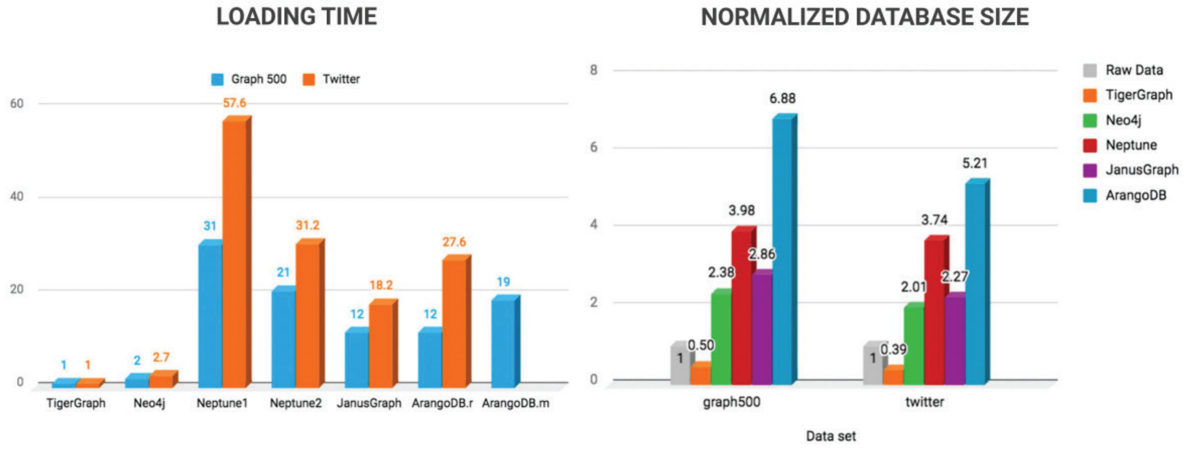
Analyzing this at scale is not practical, and in some cases, not possible with traditional database systems. Graph analysis shows and analyzes the relationships in the data. Processing and computation of the data requires a distributed, scalable system that can run on the cloud.

### Benchmarking

TigerGraph is 40x to 337x faster than other graph databases owing to its native massively parallel processing (MPP) graph architecture. In benchmark tests comparing TigerGraph to Neo4j, Amazon Neptune, Janus Graph and ArangoDB ,TigerGraph consistently outperformed all competitors by wide margins. The complete benchmark report is available at https://www.tigergraph.com/benchmark

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

## Comparison Summary

| Category | TigerGraph | Neo4j |
|---|---|---|
| Speed | Traverses 10M+ entities and relationships per second per machine and 100K+ updates per second per machine. | 10 times to 1000 times slower in independent tests. |
| Scale-out | A true distributed database, with automatic partitioning, seamless to users. | Workaround by manually sharding model, data, and queries across multiple individual non-connected graph databases - slow, with expense and risk of manual handling. |
| Deep-Link Analytics | Complex 5 to 10+ hop queries on all sizes of datasets - from small to ultra-large, distributed graphs. Runs in-database graph analytics. | Unable to support six or more hop queries on even moderate size dataset. Workaround is to export data to Spark for external processing, which is an extra infrastructure cost. |
| Graph Query Language | GSQL - Turing-complete, can express complex graph computations and analytics natively, for ad hoc queries and complex, parameterized procedures. TigerGraph is an active contributor to the upcoming GQL standard. | Cypher - for basic queries, including pattern matching. Neo4j is an active contributor to the upcoming GQL standard. |
| Transactions and Cluster Consistency | ACID across cluster. | ACID only at single-machine level. |
| Graph Algorithm Library | Open source, user extensible and customizable. Runs within the database. | Pre-compiled JAVA API calls, no ability to modify parameters or logic. |
| Visual Interface | GraphStudio for full workflow: visual modeling, ETL, exploration, and query development. AdminPortal for monitoring and management. Both included. | Bloom for graph exploration only. Available at an additional cost. |
| Standard APIs | Industry standards: REST APIs, JSON output, JDBC, Python, Spark. | Numerous industry standards + Proprietary Bolt API. |
| Cloud Offering - Graph Database as a Service | Free tier for lifetime for non-commercial usage. Contains 18+ starter kits for popular use cases. | No free tier on Neo4j cloud. No starter kits available. |
| Design | • C++ core engine<br>• Native distributed graph storage<br>• Massively parallel processing<br>• Compressed data<br>• Schema-first design optimizes query performance | • Java core engine<br>• Native single-node graph storage<br>• Limited parallelism<br>• Uncompressed data<br>• Schema-free design slows down query performance |
| Total Cost of Ownership | Best-in-class due to storage and computational efficiency, yielding the smallest hardware footprint. Hardware costs for TigerGraph are typically 50% or less when compared to Neo4j. | Storage alone is five times larger.<br>Trying to match scale and performance by using more, faster machines: 10 times to impossible.<br>Need to use Spark for OLAP queries which is an added cost (infrastructure and potentially license). |

## 1.2 Customer Feedback

If you have selected and deployed a graph database and analytics solution, congratulations to you regardless of the product you have selected for the initial deployment - you are an early adopter of technology that is becoming a core component for all IT stacks. Here are a few examples of customers who have upgraded to TigerGraph due to higher performance and scalability, more functionality and lower total cost of ownership (TCO). Team TigerGraph is happy to connect up the graph database buyers with these and other customers who can share additional details.

| Customer Profile | Customer Feedback |
|---|---|
| **CUSTOMER**<br>Sayari Labs - Fintech startup focused on Financial Crimes Prevention<br>**USE CASE**<br>Fraud Detection (Raw data < 100GB) | Sayari Labs chose TigerGraph to power Sayari Graph, the first purpose-built tool for navigating the complexity of global corporate ownership and commercial relationships. Sayari's users value the complete picture of customers, vendors, third-parties, and other deeper relationships, which can be surfaced with help of TigerGraph.<br>***"TigerGraph's graph analytics platform allowed us to go six to nine levels deep into the dataset, in real-time, to uncover questionable patterns and potentially suspicious activities. These extra levels of 'data depth' are the difference between detecting all kinds of financial crime or letting them go undiscovered."*** -- Andrew Hoagland, VP of Engineering, Sayari Labs quoted from this recent press release |
| **CUSTOMER**<br>Jaguar Land Rover<br>**USE CASE**<br>Supply Chain Planning | ***"We were very clear that we needed to find a graph database that partitioned across a distributed network. Neo4j was easy to get hold of and play around with. For small models, it's great; for models that are relatively homogenous, with not too many links, it does fine. What we found was as soon as we move beyond [a] point model, we just hit a wall."***<br>-- Harry Powell, Director of Data & Analytics, quoted from Jaguar Land Rover reaches for graph database in search of supply chain knowledge during chip shortage, 10 May 2021, The Register |
| **CUSTOMER**<br>Merkle Science<br>Fintech startup focused on financial crimes detection in cryptocurrencies<br>**USE CASE**<br>Financial Crimes, Anti-Money Laundering | ***"TigerGraph's ability to handle large quantities of data coupled with their elegant and powerful query language GSQL have enabled us to build a graph data warehouse which we use to help our users understand flows of funds and determine their risk exposure. TigerGraph has proven to be invaluable in helping our users to differentiate between good actors and bad ones."***<br>-- Nirmal Aryath Koroth, co-founder and chief technology officer at Merkle Science - quoted from this article Merkle Science Selects TigerGraph |
| **CUSTOMER**<br>Gojek<br>**USE CASE**<br>Fraud Detection | TigerGraph's performance was compared to Neo4j's in these areas:<br>Data loading: ***"TigerGraph Outperforms, especially for larger graphs. TigerGraph reached up to 1.5 million created nodes per second, while for Neo4j that number stopped at 22,000 nodes per second and degraded over time. We tried to load data with Neo4j in parallel by dividing our dataset into multiple files and tried to run multiple processes to load the data. We ran into problems with the locking mechanism in Neo4j where for those processes to run in parallel, they ran into lock contention problem and eventually, the loading could not finish."***<br>Query execution: ***"TigerGraph finished four-hop queries in 30 minutes. However, with Neo4j we couldn't get those four-hop queries to finish after 10 hours."***<br>Visualization: ***"TigerGraph's GraphStudio offers a lot more functionality when it comes to explorations and query management than Neo4j's desktop."***<br>-- Hiep Doan, Software Engineer, Gojek at Graph + AI Summit 2021 session titled "Leverage Graph Data to Detect Fraud in Real-time" |

| | |
|---|---|
| **CUSTOMER**<br>Exact Sciences<br><br>**USE CASE**<br>Doctor and Product 360 for Marketing | Exact Sciences, a molecular diagnostics company that focuses on the early detection and prevention of colorectal cancer, is an example of a company that selected TigerGraph over the competitors. After evaluating software from Amazon Neptune and Neo4j, Exact Sciences selected TigerGraph to provide a graph database and analytics because it offered functionality that Neptune and Neo4j didn't. Exact Sciences exports data from its Snowflake warehouse to TigerGraph using an Apache Spark connector, along with data from HubSpot, Microsoft, Salesforce, and other applications. Once the data is in TigerGraph, a combination of deep link analytics and pattern matching identifies causality and recommends next-best-actions. Exact Science uses a visualization toolkit from Expero to provide its marketing team with dashboards that make data interpretation easy. They also found using TigerGraph allowed them to find relationships between all their data points to reduce customer churn. |
| **CUSTOMER**<br>Technical University of Denmark<br><br>**USE CASE**<br>Cancer Research | DTU opted for an on-premise graph database platform that would deliver the required performance and evaluated a number of options, in particular, Neo4j but concluded that only TigerGraph could scale and provide the analytical depth the project required.<br><br>***"In our testing, Tigergraph was the only solution offering the highest performance with the ability to scale to the levels we will eventually need."***<br>-- Jesper Vang, PhD Student, Department of Health Technology, Cancer Systems Biology at DTU. Quoted from this press release. |
| **CUSTOMER**<br>Innovative Media Company based in Germany<br><br>**USE CASE**<br>Recommendation Engine, Customer 360 | Prior to selecting TigerGraph, the customer conducted its own in-house benchmarks based on its requirements and thoroughly compared all available systems.<br><br>***"TigerGraph provides a scalable and high-performance graph database platform. The integration has proven straightforward and the flexibility of the GSQL environment makes it much easier for developers who are not yet Graph specialists to quickly get involved in our production processes."*** -- CEO |
| **CUSTOMER**<br>OpenCorporates<br><br>**USE CASE**<br>Knowledge Graph | ***"OpenCorporates is dedicated to making information on companies more usable and widely available for the public benefit, particularly to bring to light instances of criminal or anti-social activity - such as corruption, money laundering and organized crime," said Chris Taggart, CEO, OpenCorporates. "As our work continues and our data grows, we had challenges scaling our data to meet our business needs. TigerGraph's excellent scalability and performance enables us to achieve things we previously could not do, and to better support ongoing investigative work in the process."*** Source - Press release "World's largest open database OpenCorporates Migrates to TigerGraph", Feb 2019<br><br>OpenCorporates compared TigerGraph to other graph databases using a sample set of 17 million nodes and 10 million edges on a single machine. TigerGraph offered superior support for the following must-have query requirements:<br><br>**- Degrees of separation:** Support for queries of up to five degrees of separation between entities with real-time response times - a capability that was becoming increasingly difficult for OpenCorporates.<br><br>**- Siblings:** Support for sibling queries with real-time response times, to help answer questions like, "What else does the parent of a given company own?"<br>Up the chain only: Enables users to see what entities exist up the chain only for any given company, with real-time response times.<br><br>**- Temporal graph search:** Users can ascertain if a relationship existed for a particular time frame. They can search what entities have been created from a particular date, and remove all old relationships from their query - not possible with Neo4j.<br><br>**- Active vs. dead relationships:** Supports queries on a given network to see what relationships are active vs. dead, so that each one can be filtered out of the query accordingly, a feature that wasn't possible with Neo4j. |

## 1.3 Scalability

This section compares the ability of TigerGraph and Neo4j to scale out.

| Category | TigerGraph | Neo4j 4.0 Fabric |
|---|---|---|
| Schema Sharding | One schema. | User must manually shard the schema into different sub-schemas for each machine/database. |
| Data Loading and Sharding | One loading job and automatic partitioning. | User must manually partition data and load separately to each machine/database. |
| Querying | Query as a single database. | User must design multi-stage queries to manually query each machine/database and then stitch results together. |
| Transactions | Full ACID. | Not ACID across the fabric. |
| Summary | A truly distributed database with automatic partitioning. No hassle, high performance. | Federation of separate databases. |

## 1.4 Functionality

This section compares the key functionality offered by TigerGraph and Neo4j.

| Category | TigerGraph | Neo4j |
|---|---|---|
| OLAP: Deep-Link Analytics | Handles deep-link (3 to 10+ hops) on ultra-large, distributed graphs.<br>Runs large graphs in-database. | Tops out at 2 to 5 hops on medium to large graphs. Workaround is to export data to Spark for external processing, which is an extra infrastructure cost. |
| Graph Query Language | GSQL - Turing-complete, can express complex graph computations and analytics natively for ad hoc queries and complex, parameterized procedures. Excels at analytics due to built-in parallelism and innovative accumulators. TigerGraph is an active contributor to the upcoming GQL standard. | Cypher - for basic queries, including pattern matching. Neo4j is an active contributor to the upcoming GQL standard. |
| Transactions and Cluster Consistency | ACID across cluster.<br>Strong consistency. | ACID only at single-machine level. Eventual or causal consistency. |
| Graph Algorithm Library | Open source, user extensible and customizable. Runs within the database. | Pre-compiled JAVA API calls and no ability to modify parameters or logic. |
| Visual Interface | GraphStudio for full workflow:<br>visual modeling, ETL, exploration, and query development.<br>AdminPortal for monitoring and management. Both included. | Bloom for graph exploration only. Available at an additional cost. |
| Standard APIs | Industry standards: REST APIs, JSON output, JDBC, Python, Spark. | Numerous industry standards + Proprietary Bolt API. |
| Cloud Service | The only distributed graph database as a service. HA replication too. Free tier for lifetime for non-commercial usage.<br>Over 18 starter kits including popular use cases including Customer 360, Entity Resolution, Fraud Detection, Knowledge Graph, etc. | Single instances only.<br>No free tier.<br>No starter kits. |

## 1.5 Total Cost of Ownership (Excluding Software Costs)

The cost of ownership associated with TigerGraph is dramatically lower than that of Neo4j. Here are some key considerations:

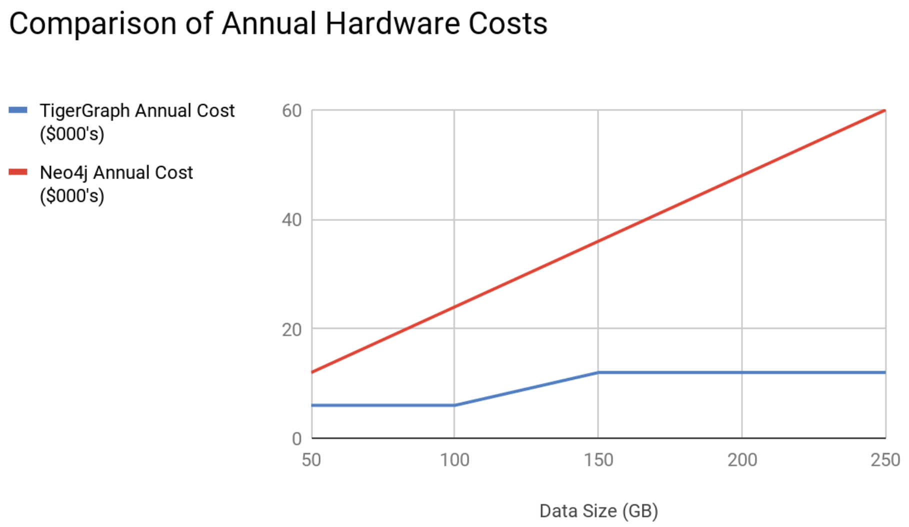
- **Storage efficiency:** TigerGraph stores your data more efficiently than any other graph database on the market, typically 4 to 5 times more compactly than Neo4j. That means TigerGraph can use fewer machines than other distributed databases.
- **Compute efficiency:** Independent testing using the LDBC benchmark test showed TigerGraph to be 10x to more than 1000x faster than Neo4j. Our faster execution helps in maintaining the higher QPS (Query per Second) rate over the longer period of time. This capability reduces the need for data replication for higher throughput purposes.  Using more expensive machines and running machines in parallel for more throughput can partially compensate for lower core performance.
- **Operational efficiency:** On TCO matrix it is a no brainer that fewer the servers, lesser is the direct cost of operations, administrations, tech-support and training; The performant clusters leads to lesser in-direct cost of security, configurability, upgrades, data storage,  backups, and so on.

Unlike TigerGraph, which compresses raw data when loaded into a graph, Neo4j typically expands it. The following table shows the loaded data storage size for TigerGraph and Neo4j:

| Dataset | Raw Data | TigerGraph | Neo4j |
|---|---|---|---|
| graph500 | 967 MB | 482 MB (50% of raw data) | 2300 MB (237% of raw data) |

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

The following figure shows the annual computing or **hardware** costs for TigerGraph vs Neo4j based on the memory or RAM requirements for each graph database. Software costs are not included:



The following table explains how the figure was created. It starts with 50 GB of raw data and scaling up to 250 GB of raw data. The hardware or computing cost savings start at 50% and increase to 80% as the amount of raw data increases to 250 GB. Software costs not included.

| Raw Data (GB) | Graph size in TigerGraph (GB) = raw data x 50% | Annual cost for computing for TigerGraph (6,000 dollars per 64GB RAM) | Graph size in Neo4j (GB) = raw data x 237% | Annual cost for computing for Neo4j (6,000 dollars per 64GB RAM) | Percentage cost savings with TigerGraph |
|---|---|---|---|---|---|
| 50 | 25 | $6,000 | 119 | $12,000 | 50% |
| 100 | 50 | $6,000 | 238 | $24,000 | 75% |
| 150 | 75 | $12,000 | 357 | $36,000 | 66% |
| 200 | 100 | $12,000 | 476 | $48,000 | 75% |
| 250 | 125 | $12,000 | 595 | $60,000 | 80% |

Moreover, the speed that TigerGraph analyzes the data on these servers will be many times faster than the speed that Neo4j analyzes it - TigerGraph traverses 10M+ entities and relationships per second per machine, while Neo4j has been shown to be 10 times to 1000 times slower in independent tests. This translates into more queries per second or QPS with TIgerGraph when compared to Neo4j, allowing customers to scale up the deployment with more users or systems accessing the insights from the graph database.

## 1.6 Additional Resources

**DOWNLOADS**

- In-Depth Benchmarking of Graph Database Systems with the Linked Data Benchmark Council (LDBC)Social Network Benchmark (SNB)
- Benchmarking Graph Analytic Systems: TigerGraph, Neo4j, Neptune, JanusGraph and ArangoDB
- TigerGraph's Native Parallel Graphs eBook
- Try TigerGraph for Free

**CUSTOMER FEEDBACK**

- Jaguar Land Rover article in the Register
- Gojek: TigerGraph vs Neo4j benchmark results see video at 17 min 50 seconds
- Customer News: TigerGraph 3.0 announcement: Innovative fintech startup focused on financial crimes prevention upgrades to TigerGraph
- Customer News: Database of corporate information upgrades to TigerGraph
- Customer News: Innovative media company based in Germany upgrades to TigerGraph
- Customer Video: Pharmaceutical manufacturer upgrades to TigerGraph

## 1.7 Evaluating TigerGraph

The best way to evaluate the product as a buyer is to experience it yourself. Please sign up for TigerGraph Cloud at https://www.tigergraph.com/cloud/ to get started with the FREE tier - no credit card needed. You can contact sales@tigergraph.com to learn more about upgrading to TigerGraph.

# PART 2 - COMPARING AMAZON NEPTUNE AND TIGERGRAPH CLOUD

## Selecting a graph database for cloud deployment

Graph databases are the fastest growing category in all of data management. Since seeing early adoption by companies including Facebook, Google and LinkedIn, graph has evolved into a mainstream technology used today by enterprises in every industry across a wide variety of use cases. By organizing data in a graph format, graph databases overcome the big and complex data challenges that other databases such as Relational and NoSQL cannot.

Selecting graph software is an important decision which can shape the success of your organization. Unfortunately buyers often struggle to reconcile the conflicting claims made by different graph software vendors - these claims are often characterized by misinformation.

Part two of the buyer's guide is intended to assist you in your buying decision by providing a side-by-side comparison of two leading graph databases with cloud offerings, TigerGraph and Amazon Neptune. It includes the following information:

### PART TWO - COMPARING AMAZON NEPTUNE AND TIGERGRAPH CLOUD

**Table of Contents**

## 2.1 TigerGraph and Amazon Neptune Comparison Summary

This section provides a high-level comparison of TigerGraph and Amazon Neptune and includes information on speed, scalability, cloud availability, the total cost of ownership and more. As you consider the purchase of a graph database as a service, here are the key questions that most buyers work through:

1. Will my graph database as a service continue to serve my needs now and into the future as the volume and complexity of my data grow?
2. Can my data scale across multiple machines to enable me to analyze growing datasets?
3. What is the performance-to-price ratio of my graph database as a service?

### Graph is Fundamental to Machine Learning, AI and Analytics

Graph is quite common as a foundation and enabler in the analytics world. Business people are asking increasingly complex questions across structured and unstructured data - it often requires blending of data from multiple sources, multiple business units, and increasingly external data.

Analyzing this at scale is not practical, and in some cases, not possible with traditional database systems. Graph analysis shows and analyzes the relationships in the data. Processing and computation of the data requires a distributed, scalable system that can run on the cloud.

## Comparison Summary Table

| Category | TigerGraph | Amazon Neptune |
|---|---|---|
| Speed | Traverses 100M entities and relationships per second per machine and 100K+ updates per second per machine. | 6 times to 60 times slower in benchmark tests. |
| Scale-out | A true distributed database, with automatic partitioning, seamless to users. | Computation does not scale horizontally. Not a distributed database. |
| Deep-Link Analytics | Complex 5 to 10+ hop queries on all sizes of datasets - from small to ultra-large, distributed graphs. Runs in-database graph analytics including complex OLAP. | Tops out at 3 to 6 hops on medium to large graphs. Not designed and no capability for OLAP. |
| Graph Query Language | GSQL. Turing-complete, can express complex graph computations and analytics natively, for ad hoc queries and complex, parameterized procedures. TigerGraph is an active contributor to the upcoming GQL standard. | SPARQL or Gremlin, but not both at the same time. SPARQL is for RDF data. Gremlin is for property graphs - Turing-complete but advanced programming skills are needed for asking complex questions and solving real-life business problems. Less intuitive than GSQL or other language alternatives. |
| Transactions and Cluster Consistency | ACID across an entire cluster. | ACID-compliant. |
| Graph Algorithm Library | Open source, user extensible and customizable. Runs within the database. | None (lacks even Gremlin's modest algorithm library). |
| Visual Interface | GraphStudio for full workflow: visual modeling, ETL, exploration, and query development. AdminPortal for monitoring and management. | Neptune offers visualization via partners which come with add-on costs. |
| Standard APIs | Industry standards: REST APIs, JSON output, JDBC, Python, Spark. | Wide range of options in API support. |
| Cloud Offering - Graph Database as a Service | Fully-managed, cloud-based graph database. No cloud vendor lock-in. Free tier for lifetime for non-commercial usage. Contains 18+ starter kits across popular use cases. | Fully-managed, cloud-based graph database, available on AWS. Vendor lock-in: users cannot move on-premises or switch cloud providers. |
| Design | • C++ core engine<br>• Native distributed graph storage<br>• Massively parallel processing<br>• Compressed data<br>• Schema-first design optimizes query performance | Does not distribute data, so "horizontal scaling" is just making a replica. Vertically scalable system that relies on expensive machines (with lots of RAM), and data replication for higher throughput |
| Developer Community | Rapidly growing developer community. | Small developer community, limited resources and tutorials. |
| Performance-to-Price Ratio | TigerGraph is a cost-effective graph solution (as demonstrated by a benchmarking test). | The cost per query time for Amazon Neptune is 2.6 times higher than that of TigerGraph at least, and can be even as high as 9.7 times more costly (for a three-hop path query, the best case scenario for Amazon Neptune in a benchmarking test). |

## 2.2 Customer Feedback

If you have selected and deployed a graph database and analytics solution, congratulations to you regardless of the product you have selected for the initial deployment - you are an early adopter of technology that is becoming a core component for all IT stacks.

Here are a few examples of customers who have upgraded to TigerGraph due to higher performance and scalability, more functionality and lower total cost of ownership (TCO). TigerGraph is happy to connect graph database buyers with these and other customers who can share additional details.

| Customer Profile | Customer Feedback |
|---|---|
| CUSTOMER<br>Large US Financial Services Payment Processor<br>USE CASE<br>Fraud Detection | - Despite the fact that they are a loyal AWS customer, they evaluated Neptune and found it to be lacking on key performance requirements: data ingestion, speed, graph analytics query response time<br>- Additionally, TigerGraph was determined to be architecturally superior - the only native parallel graph database<br>**"We are impressed by TigerGraph's built-in massive parallel processing architecture, unique vertices optimization for storage and indexing, ability to support ACID for both OLTP and OLAP queries, and superb performance/ scaling for complete deep traversal queries, and its developer-focus and hunger for growth."**<br>**- Senior Architect** |
| CUSTOMER<br>Cybersecurity Company<br>USE CASE<br>Knowledge Graph with Machine Learning | - Unable to scale their cybersecurity services with their existing SQL Server<br>- Tested Neptune as an alternative, but it was unable to meet their performance requirements<br>- Harnessing graph technology to continuously update and expand its knowledge of URL classifications and risk scores in the face of rapid URL expansion, and identify new cyber threats at scale with real-time analytics |
| CUSTOMER<br>Cloud based supply management software company<br>USE CASE<br>Pattern Matching, Supply Chain Management | - Needed a way to identify specific patterns across purchase orders to accelerate order fulfillment and improve efficiencies<br>- Attempted to solve their business challenge with Neptune, but they ran into significant performance challenges and queries that simply wouldn't return, so they turned to TigerGraph<br>**"We've been misled by a number of graph database companies but TigerGraph is as advertised"**<br>**- Data Sciences Engineer** |
| CUSTOMER<br>Innovative Media Company based in Germany<br>USE CASE<br>Recommendation Engine, Customer 360 | - Prior to selecting TigerGraph, the customer conducted its own in-house benchmarks based on its requirements and thoroughly compared all available systems<br>- With the shortlist decided, the customer then built prototypes and performed more detailed performance tests. Despite the ubiquity of AWS in their stack, the company chose TigerGraph for its powerful performance.<br>**"TigerGraph provides a scalable and high-performance graph database platform," says the customer. The integration has proven straightforward and the flexibility of the GSQL environment makes it much easier for developers who are not yet Graph specialists to quickly get involved in our production processes."**<br>**- CEO** |

## 2.3 Scalability

This section compares the ability of TigerGraph and Amazon Neptune to analyze increasingly larger amounts of data and so on within reasonable timeframes.

| Category | TigerGraph | Amazon Neptune |
|---|---|---|
| Distributed Database | Yes | No |
| Scalable Storage | Unlimited. Data can be partitioned across any number of instances. Instances can be added. | Increase in increments of 10GB up to 64TB. |
| Storage efficiency | Typically compresses raw data down to 50% of original size. | Typically expands raw data to ~400% of original size. |
| Scalable Compute | Scale-up or scale-out. Users can both use more powerful machines AND increase the number of machines. | Scale-up only: cannot distribute a query across multiple machines. |
| Read Replicas | Yes | Yes |
| Summary | Distributed, replicated complete database supports both<br>• high transaction throughput (OLTP) AND<br>• analyzing massive, growing datasets (OLAP). | Scalable, replicated storage with read replicas supports<br>• high transaction throughput (OLTP) only<br>• Not suited for analytics of large datasets. |

## 2.4 Functionality

This section compares the key functionality offered by TigerGraph and Amazon Neptune.

| Category | TigerGraph | Amazon Neptune |
|---|---|---|
| OLAP: Deep-Link Analytics | Handles deep-link (3 to 10+ hops) on ultra-large, distributed graphs.<br>Runs in-database large graphs. | Tops out at 3 to 6 hops on medium to large graphs. Not designed and no capability for OLAP. |
| Graph Query Language | GSQL. Turing-complete, can express complex graph computations and analytics natively, for ad hoc queries and complex, parameterized procedures. Excels at analytics due to built-in parallelism and innovative accumulators. TIgerGraph is an active contributor to the upcoming GQL standard. | Gremlin. Turing-complete but does not offer the same ease of use as GSQL - advanced programming skills are needed for asking complex questions and solving real-life business problems. Less intuitive to learn than GSQL or other language alternatives. |
| Transactions and Cluster Consistency | ACID across an entire cluster.<br>Strong consistency. | ACID-compliant. |
| Graph Algorithm Library | Open source, user extensible and customizable. Runs within the database. | None (lacks even Gremlin's modest algorithm library). |
| Visual Interface | GraphStudio for full workflow:<br>visual modeling, ETL, exploration, and query development.<br><br>AdminPortal for monitor and management. Both included. | Neptune offers visualization via partners which come with add-on costs. |
| Standard APIs | Industry standards: REST APIs, JSON output, JDBC, Python, Spark. | Wide range of options in API support. |
| Cloud Service | The only distributed graph database as a service. HA replication too.<br><br>Free tier for lifetime for non-commercial usage.<br><br>Over 18 starter kits including popular use cases such as customer 360, entity resolution, fraud detection, knowledge graph, recommendation engine and industries such as healthcare, financial services, internet, pharmaceutical and telecom. | Fully-managed, cloud-based high-performance graph database, available on AWS. Vendor lock-in: users cannot move on-premises or switch cloud providers. |

## 2.5 Performance-to-Price Ratio

The performance-to-price ratio of TigerGraph Cloud is significantly superior to that of Amazon Neptune. This conclusion is derived from the following two considerations:

**1) Computing costs:**
TigerGraph stores data more efficiently than any other graph database on the market: Neptune typically needs 8 times more disk storage for the same input graph data. Unlike TigerGraph, which compresses raw data when loaded into a graph, Neptune typically expands it. The following table compares how TigerGraph and Amazon Neptune store 1 GB of input data:

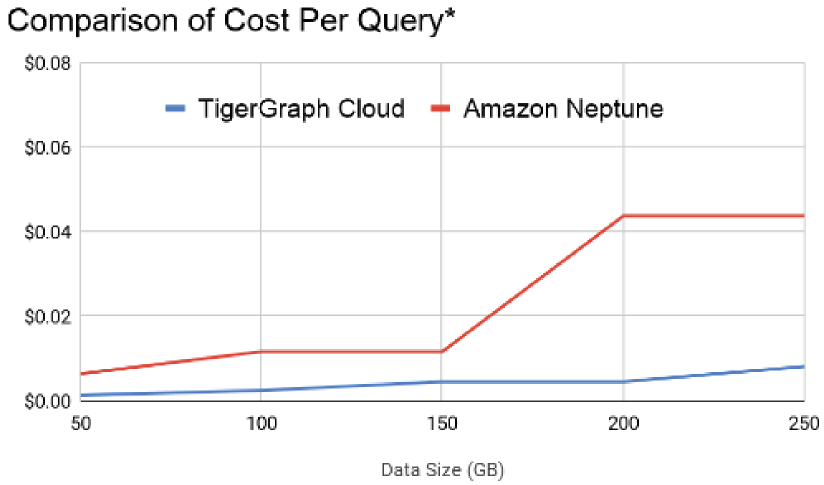
| Dataset | Raw Data | TigerGraph | Neptune |
|---|---|---|---|
| graph500 | 967 MB | 482 MB (50% of raw data) | 3,850 MB (400% of raw data) |

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

Both TigerGraph Cloud and Amazon Neptune are tuned to run well when the graph can be loaded into memory. TigerGraph Cloud, however, is a more economical option - a TigerGraph instance with X CPUs and Y RAM costs approximately the same as an Amazon Neptune instance with X CPUs and 8Y RAM. Additionally, TigerGraph Cloud does not charge for I/O whereas Neptune does.

**2) Computing efficiency:**
Amazon Neptune's documentation emphasizes that their Gremlin implementation is for graph traversal, without mention of computation or analytics. Nevertheless, on benchmark tests which ran a stream of graph traversals (a strong point for Neptune), Neptune was 5.5 times slower than TigerGraph at a minimum and, in some instances, didn't complete queries at all. For example, Neptune required 2.27 seconds to complete a three-hop path query, while TigerGraph required only 0.41 seconds. TigerGraph's faster execution helps in maintaining higher QPS (Query per Second) for sustained higher performance.

The combination of the differences in computing costs and compute efficiency are encapsulated in the following figure:



*Based on three-hop path query time

The figure shows that cost per query for Amazon Neptune is three times higher than that of TigerGraph at least, and can even be 10 times more costly. This demonstrates that the performance-to-price ratio of TigerGraph Cloud is dramatically better than that of Amazon Neptune, even assuming the smallest performance difference (three-hop path query).

## 2.6 Additional Resources

### DOWNLOADS

- Benchmarking Graph Analytic Systems: TigerGraph, Neo4j, Neptune, JanusGraph and ArangoDB
- TigerGraph's Native Parallel Graphs eBook
- Try TigerGraph for Free

### CUSTOMER FEEDBACK

- Customer News: Innovative media company based in Germany upgrades to TigerGraph

## 2.7 Evaluating TigerGraph

The best way to evaluate the product as a buyer is to experience it yourself. Please sign up for TigerGraph Cloud at https://www.tigergraph.com/cloud/ to get started with the FREE tier - no credit card needed. You can contact sales@tigergraph.com to learn more about upgrading to TigerGraph.

# PART3 - COMPARING DATASTAX AND TIGERGRAPH

## Is it time to update your graph database?

Graph databases are the fastest growing category in all of data management. Since seeing early adoption by companies including Facebook and Google and LinkedIn, graph has evolved into a mainstream technology used today by enterprises in every industry across a wide variety of use cases. By organizing data in a graph format, graph databases overcome the big and complex data challenges that other databases such as Relational and NoSQL cannot.

Selecting graph software is an important decision which can shape the success of your organization. Unfortunately, buyers often struggle to reconcile the conflicting claims made by different graph software vendors - these claims are often characterized by misinformation. This guide is intended to assist you in your buying decision by providing a side-by-side comparison of two leading graph databases, DSE Graph and TigerGraph. It includes the following information:

### Table of Contents

## 3.1 Summary and Comparison

This section provides a high-level comparison of TigerGraph and DataStax and includes information on speed, scalability, cloud availability, total cost of ownership and more. As you consider the purchase of a graph database, here are the key questions that most buyers work through:

1. Where is the graph market going? How should I prepare?

2. Will my selected graph database continue to serve my needs now and into the future as the complexity as well as the volume of my data grows? In other words, is my choice or investment future-proof?

3. How easy is it to distribute the data across multiple machines to avoid adding CPUs or RAM to a single, expensive machine?

4. What is the total cost of ownership for the selected graph database considering the initial cost of license (on-premises), the cost of infrastructure, whether physical hardware or cloud resources, and cost of maintaining and upgrading my solution?

### Graph is Fundamental to Machine Learning, AI and Analytics

Graph is quite common as a foundation and enabler in the analytics world. Business people are asking increasingly complex questions across structured and unstructured data - it often requires blending of data from multiple sources, multiple business units, and increasingly external data.

Analyzing this at scale is not practical, and in some cases, not possible with traditional database systems. Graph analysis shows and analyzes the relationships in the data. Processing and computation of the data requires a distributed, scalable system that can run on the cloud.

## Comparison Summary Table - Page 1

| Category | TigerGraph | DSE Graph |
|---|---|---|
| Architecture Design | Native, distributed, graph storage.<br>MPP (Massively parallel processing).<br>Compressed data. | Not a native graph.<br>Storage is in Cassandra (key-value store), resulting in an "impedance mismatch" and slow performance.<br>Data is compressed in Cassandra. |
| Query Performance | Traverses 10M+ entities and relationships per second per machine and 100K+ updates per second per machine. | Doesn't store data natively - uses Cassandra for storage. As a result, cannot do deep link/multi-hop queries in performant fashion. |
| Loading Performance | Ultra-fast parallel loading with 100 GB/hour/server throughput. Flexibility to choose different bulk loaders. | Cassandra has a bulk loader, but does not support loading data in any other format such as CSV directly. |
| Scale-out | A true distributed database, with automatic partitioning, seamless to users. | Although DSE Graph can scale, query performance degrades with larger dataset sizes. |
| Deep-Link Analytics | Complex 5 to 10+ hop queries on all sizes of datasets - from small to ultra-large, distributed graphs.<br>Runs in-database graph analytics. | Unable to support deep link analytics in a performant manner. A workaround is to export data to a platform like Spark GraphX (eg. Amazon EMR) for external processing, which is an extra infrastructure cost. |
| Graph Query Language | GSQL can express complex graph computations and analytics natively, for ad hoc queries and complex, parameterized procedures.<br>TigerGraph is an active contributor to the upcoming GQL standard. | Gremlin is less intuitive than GSQL or other language alternatives.<br>The Gremlin community has not participated in the GQL standards committee thus far. |
| Transactions and Cluster Consistency | ACID across cluster. Strong consistency. Designed for real-time updates. | Does not offer ACID transactions, sacrificing consistency for availability. Not designed for real-time updates. |
| Graph Algorithm Library | Open source, user extensible and customizable. Runs within the database. | Offers some deep-link algorithms, but its inherent weakness in analytics (breadth-first) traversal makes a large class of algorithms difficult to run at any scale, limiting their utility. For example, in-graph machine learning, and community detection, are not useful. |
| Visual Interface | GraphStudio for full workflow: visual modeling, ETL, exploration, and query development.<br>AdminPortal for monitoring and management. Both included. | Datastax Studio is supplied, as well as DSE OpsCenter and DSE Lifecycle Manager. |

Comparison Summary Table - Page 2

| Category | TigerGraph | DSE Graph |
|---|---|---|
| Standard APIs | Industry standards: REST APIs, JSON output, JDBC, Python, Spark. | Range of options in API support. |
| Cloud Offering - Graph Database as a Service | Free tier for lifetime for non-commercial usage. Contains 25+ starter kits across popular use cases. | No managed cloud platform today and nothing planned. |
| Developer Community | Rapidly growing developer community. | Although DataStax enjoys a wide developer community, DSE Graph has only a limited community. |
| Total Cost of Ownership | Best-in-class due to advanced storage compression and computational efficiency, yielding the smallest hardware footprint.<br>Hardware costs for TigerGraph are lower than for DSE Graph due to advanced compression technology.<br>Additionally, being able to do both OLTP and OLAP on a single platform, means cost savings from not having a separate graph platform for OLAP (eg. Amazon EMR) | High hardware costs arise because it is relatively inefficient at scale, therefore requiring a lot of hardware to achieve consistent OLTP throughput and latency. Even more hardware is needed for analytics as DSE Graph relies on a completely different technology (Spark) to perform distributed processing.<br>DSE Graph is not a native graph platform. Persistence layer is Cassandra, a NoSQL database. Because of “impedance mismatch”, it cannot do deep link multi-hop analytics.<br>Companies often require another platform for deep OLAP type queries (eg. Amazon EMR). This is an extra cost. |

## 3.2 Customer Feedback

If you have selected and deployed a graph database and analytics solution, congratulations to you regardless of the product you have selected for the initial deployment - you are an early adopter of technology that is becoming a core component for all IT stacks.

Here are a few examples of customers who have upgraded to TigerGraph due to higher performance and scalability, more functionality and lower total cost of ownership (TCO). TigerGraph is happy to connect graph database buyers with these and other customers who can share additional details.

| Customer Profile | Customer Feedback |
|---|---|
| CUSTOMER<br>Innovative marketing technology startup based in UK<br>USE CASE<br>Customer 360<br>Raw data - few hundred GB | The customer provides an integrated platform to get the holistic customer view across all the digital touchpoints. They have the data for their clients stored in a data lake built on Cassandra, Apache Spark is used for data integration as well as analytics.<br>The customer needed a solution for deeper analysis of the relationship of audience segments with the campaign data across all touchpoints.<br>They evaluated TigerGraph, DataStax Enterprise Graph, Neo4j as well as Amazon Neptune in the first round. DataStax Enterprise Graph was eliminated early on as it slowed down considerably for queries going deep into the dataset. TigerGraph was the only vendor considered and selected in the second round, as TigerGraph met the performance requirements set forth for evaluation. |
| CUSTOMER<br>Large Fortune 500 Healthcare<br>USE CASE<br>Single View of Patient, Patient Similarity and Cohort Building, Real-time Clinical Support and Recommendations<br>Raw data - several Terabytes | This healthcare juggernaut conducted an extremely thorough RFP process that involved eight graph vendors including TigerGraph, DataStax, Neo4j and others. DataStax Enterprise Graph was not considered beyond the first round, as the performance requirements for loading and analyzing patient data from 200+ sources with real-time patient similarity matching was not possible.<br>TigerGraph is used in daily operations at the customer with over 23,000 users. TigerGraph delivers a single view of the member journey that integrates data across over 200 sources and finds similar patients based on over 200 features across 10 million+ patients in real-time. |

## 3.3 Scalability

This section compares the ability of TigerGraph and DataStax Enterprise Graph to scale out.

| Category | TigerGraph | DSE Graph |
|---|---|---|
| Distributed Database | Yes, TigerGraph is a truly distributed database, with automatic partitioning, seamless to users. | Although DSE Graph can scale, query performance is impacted. |
| Schema Sharding | One schema. There is no need to deal with sharding in the schema as data distribution is transparently and automatically managed by the cluster storage engine. | One schema. |
| Querying | Query as a single database. | Limited ability to distribute queries without use of additional technologies (Spark). |
| Transactions | TigerGraph offers full ACID transactions via MVCC and snapshot isolation. | DSE inherits the same transaction model as Cassandra - in other words, it sacrifices consistency for availability. |
| Summary | A truly distributed database with automatic partitioning. High transaction throughput on relatively modest clusters due to MPP. Excellent native analytics support. No hassle, high performance. | Scalable, replicated storage with read replicas supports<br>- High transaction throughput (OLTP) with sufficient hardware resources.<br>- Poorly suited for analytics of large datasets. |

## 3.4 Functionality

This section compares the key functionality offered by TigerGraph and DataStax Enterprise Graph

| Category | TigerGraph | DSE Graph |
|---|---|---|
| OLAP: Deep-Link Analytics | From day one TigerGraph was designed to handle analytics workloads at scale. Handles deep-link (3 to 10+ hops) even on ultra-large, distributed graphs.<br>Runs large graphs in-database. | Typically tops out at 3 to 6 hops on medium to large graphs. Not designed for deep-link multi-hop analytics. |
| Multi-tenancy within a Graph | Multi-tenancy is supported using Multi-Graph | No support for multiple different users with DSE Graph |
| Graph Query Language | GSQL, TigerGraph's query language, can express complex graph computations and analytics natively, for ad hoc queries and complex, parameterized procedures. Excels at analytics due to built-in parallelism and innovative accumulators. TIgerGraph is an active contributor to the upcoming GQL standard. | Gremlin, required for DSE Graph, is less intuitive than GSQL and other languages.<br>The Gremlin community has not participated in the GQL standards committee thus far. |
| Transactions and Cluster Consistency | ACID across an entire cluster. Strong consistency. | DSE does not offer ACID transactions, sacrificing consistency for availability. |
| Graph Algorithm Library | Open source, user extensible and customizable. Runs within the database. | DSE offers some deep-link algorithms, but it's inherent weakness in analytic (breadth-first) traversal makes a large class of algorithms difficult to run at any scale, limiting their utility. So, for example, in graph machine learning, or community detection are not useful. |
| Visual Interface | GraphStudio for full workflow: visual modeling, ETL, exploration, and query development.<br>AdminPortal for monitoring and management. Both included. | Datastax Studio is supplied, along with DSE OpsCenter and DSE Lifecycle Manager. |
| Standard APIs | Industry standards: REST APIs, JSON output, JDBC, Python, Spark. | Range of options in API support. |
| Cloud Service | The only distributed graph database as a service. HA replication too.<br>Free tier for lifetime for non-commercial usage.<br>Over 25 starter kits including use cases such as Customer 360, Entity Resolution, Fraud Detection, Knowledge Graph, Recommendation Engine and industries such as healthcare, financial services, internet, pharmaceutical and telecom. | No managed cloud platform today and nothing planned. |

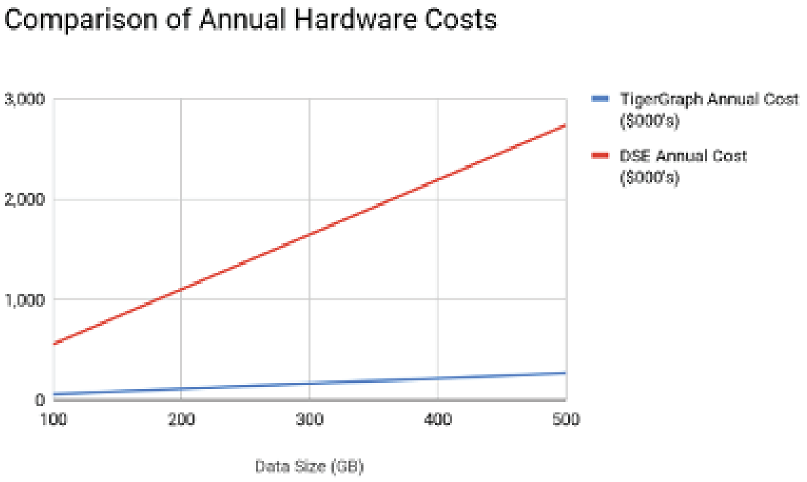
| Category | TigerGraph | DSE Graph |
|---|---|---|
| OLAP: Deep-Link Analytics | From day one, TigerGraph was designed to handle analytics workloads at scale. Handles deep-link (3 to 10+ hops) even on ultra-large, distributed graphs. Runs large graphs in-database. | Typically tops out at 3 to 6 hops on medium to large graphs. Not designed for deep-link multi-hop analytics. |
| Multi-tenancy within a Graph | Multi-tenancy is supported using Multi-Graph | No support for multiple different users with DSE Graph |
| Graph Query Language | GSQL, TigerGraph's query language, can express complex graph computations and analytics natively, for ad hoc queries and complex, parameterized procedures. Excels at analytics due to built-in parallelism and innovative accumulators. TIgerGraph is an active contributor to the upcoming GQL standard. | Gremlin, required for DSE Graph,  is less intuitive than GSQL and other languages.<br>The Gremlin community has not participated in the GQL standards committee thus far. |
| Transactions and Cluster Consistency | ACID across an entire cluster. Strong consistency | DSE does not offer ACID transactions, sacrificing consistency for availability. |
| Graph Algorithm Library | Open source, user extensible and customizable. Runs within the database. | DSE offers some deep-link algorithms, but its inherent weakness in analytic (breadth-first) traversal makes a large class of algorithms difficult to run at any scale, limiting their utility. So, for example, In-graph machine learning, or community detection are not useful. |
| Visual Interface | GraphStudio for full workflow: visual modeling, ETL, exploration, and query development.<br>AdminPortal for monitoring and management. Both included. | Datastax Studio is supplied, along with DSE OpsCenter and DSE Lifecycle Manager. |
| Standard APIs | Industry standards: REST APIs, JSON output, JDBC, Python, Spark. | Range of options in API support. |
| Cloud Service | The only distributed graph database as a service. HA replication too. Free tier for lifetime for non-commercial usage.<br>Over 25 starter kits including use cases such as Customer 360, Entity Resolution, Fraud Detection, Knowledge Graph, Recommendation Engine and industries such as healthcare, financial services, internet, pharmaceutical and telecom. | No managed cloud platform today and nothing planned. |

## 3.5 Total Cost of Ownership

The cost of ownership of TigerGraph is dramatically lower than that of DataStax. Here are some key considerations:

**Storage efficiency:** TigerGraph stores your data more efficiently than any other graph database on the market. That means TigerGraph can use fewer machines than other distributed databases.

Data from one recent study, for example, can be used to compare the storage costs of TigerGraph and DSE Graph as data size increases:



**Compute efficiency**: Independent testing, such as this study using the Linked Data Benchmark Council (LDBC) Social Network Benchmark (SNB), repeatedly demonstrates that TigerGraph is faster than other graph databases. Our faster execution helps in maintaining the higher QPS (Query per Second) rate over the longer period of time. This capability reduces the need for data replication for higher throughput purposes. Using more expensive machines and running machines in parallel for more throughput can partially compensate for lower core performance.

**Operational efficiency**: the combination of better storage efficiency and better compute efficiency result in TigerGraph having the lowest total cost of ownership - a lower amount of servers equates to a lower cost of operations, administrations, technical support and training.

## 3.6 Additional Resources

### DOWNLOADS

- Benchmarking Graph Analytic Systems: TigerGraph, Neo4j, Neptune, JanusGraph and ArangoDB
- TigerGraph's Native Parallel Graphs eBook
- Try TigerGraph for Free

CUSTOMER FEEDBACK

- Press: Innovative media company based in Germany upgrades to TigerGraph -
- Video - Pharmaceutical manufacturer
- Press: Database of corporate information

## 3.7 Evaluating TigerGraph

The best way to evaluate the product as a buyer is to experience it yourself. Please sign up for TigerGraph Cloud at https://www.tigergraph.com/cloud/ to get started with the FREE tier - no credit card needed. You can contact sales@tigergraph.com to learn more about upgrading to TigerGraph.

**TigerGraph, Inc. | www.tigergraph.com | (650) 206-8888**

**3 Twin Dolphin Drive, Suite 225, Redwood City, CA 94065**