McKNIGHT
Consulting Group

Trillion Edge Knowledge Graph

The first demonstration of a massive Knowledge Graph that consists of materialized and virtual graphs that span multiple cloud platforms

By McKnight Consulting Group
June 2021

Sponsored by

STARDOG

1 Biased to Stardog

2 Does replicate "real world" by spreading data at

3 1 Trillion Nodes is cool

4 They use the BSBM generator to make for 1 trillion nodes, but don't explain any other characteristics of it

5 They only measure latency, but using a server with almost 1 TB of memory to query suggests memory intensive.

6 Queries derived from the Berlin Benchmark. Only 10 queries and most very simplistic "explore" benchmark.

7 Further Reading:
1 Berlin Sparql Benchmark. 2009
2 Leigh University Benchmark. 2005

Both V OLD.

Table of Contents

Executive Overview

The requirement for enterprise data to deliver value to the business has never been stronger. However, enterprises often trip over their own data when the business landscape shifts or new information needs arise. Conventional relational database management systems worked acceptably well when the enterprise data landscape was itself predominantly structured. But the world has changed. The enterprise data landscape is increasingly voluminous, varied, and changing. The emergence of IoT, the rise in unstructured data volume, increasing relevance of external data, and the trend towards hybrid multi-cloud environments are challenges that must be overcome with each new request for data. Data strategies centered around relational data systems are rarely sufficient anymore, especially as the requirements to connect data across the enterprise increase.

How can enterprises create a proactive, responsive data strategy? Enterprise data fabrics offer a new path forward. A data fabric weaves together data from internal silos and external sources and creates a seamless network of information. They must support the full gambit of the connected enterprise.

The forerunners to the modern data fabric have been data federation and virtualization technologies. Many of these platforms have failed to deliver true inter-connectedness at scale with performance, because they are hampered by bottlenecks inherent in all the databases and data stores in the query chain. Rather than tackle the data fabric with another abstraction layer, it makes more sense to leverage a database technology that was engineered for data relationships—a graph database.

Graph databases provide tremendous utility to an organization whenever that organization has connected data. Functions are made available to understand the data relationships, prioritization of nodes in the relationship can be determined, and the visualization makes it easier for users to search, investigate, and analyze data, and expose patterns and trends in the data. A Knowledge Graph is a type of data integration platform that takes components from graph databases, data virtualization, query federation, and semantic inference capabilities and is designed to meet the organization's requirements to connect diverse forms of connected knowledge.

An enterprise today inevitably has many data stores with data interesting to a Knowledge Graph. There are Operational Databases, Operational Big Data Stores, Operational Data Hubs, Master Data Management, Data Warehouses, Data Marts, Data Lakes, Analytic Big Data Applications, etc., and they are spread across multiple cloud platforms. Not to mention all of the "application data silos" that exist in the public clouds and other cloud-based apps and platforms. Physically consolidating data for a Knowledge Graph can be prohibitive. Yet that is necessary when you cannot use data from its natural locations in the architecture. With Stardog, we set out to build a demonstration of an enterprise-class Knowledge Graph that consists of materialized and virtualized graphs that span multiple cloud platforms.

This is the first demonstration of a massive Knowledge Graph that consists of materialized and virtual graphs that span **multiple cloud platforms**. We show that it is possible to have a **<u>one trillion-edge Knowledge Graph</u>** with **<u>sub-second query times</u>** **without storing all the data in a central location**. This capability has the ability to usher in a new era where the Knowledge Graph is a powerful component of company profitability and competitive advantage.

Data Stays
will be
integrity.–
is it actually 1
trillion edges or
is it smaller
graphs?

Benchmark Setup

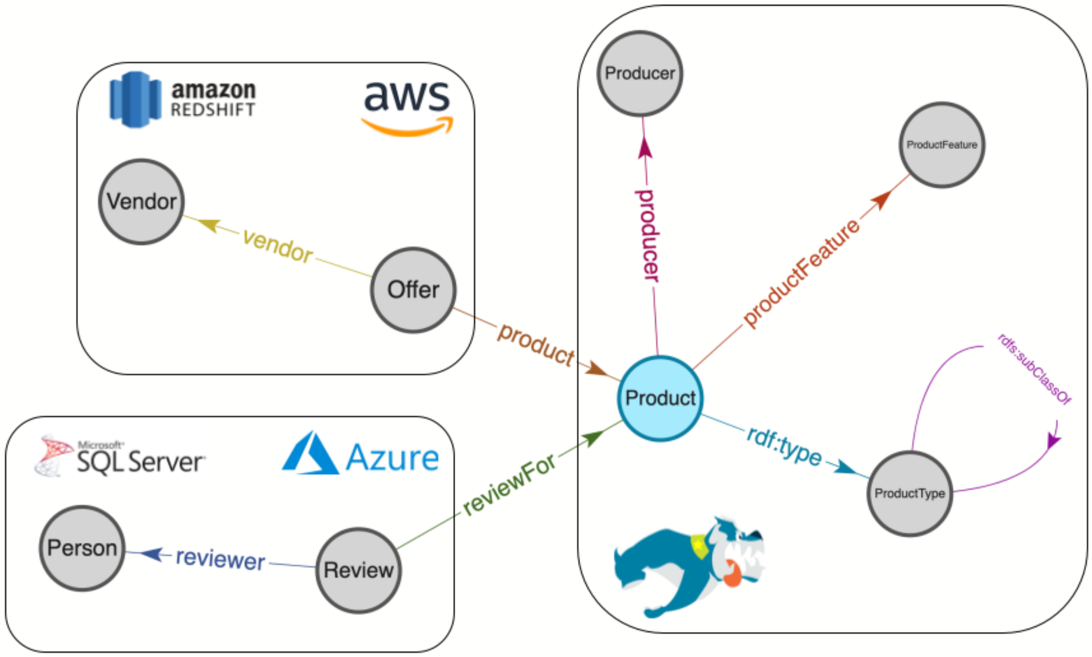
The purpose of this demonstration was to create a single unified Knowledge Graph with 1 trillion edges against which we could run structured graph queries. In reality, the data was spread across multiple heterogeneous data sources reflecting the common situation in large enterprises. In order to match the distributed nature of the enterprise, we have created an environment where our 1 trillion-edge graph was distributed over three systems: **Stardog, Amazon Redshift in AWS, and SQL Server in Azure**. Stardog handles the execution of SPARQL graph queries by reaching out to Redshift and SQL Server as needed and hiding the complexity of data distribution from end users.

In order to demonstrate the scalability of the Stardog system we chose the Berlin SPARQL Benchmark, which is commonly used for measuring the performance of systems that support SPARQL query answering. The benchmark suite is built around an e-commerce use case, where a set of products is offered by different vendors and different consumers have posted reviews about products.

look up.

The BSBM dataset contains eight main classes (tables) and the dataset can be generated at different scales. The data is product-oriented with information on product, product features, vendors providing offers, review, etc.

Typically, all the data is loaded into a single storage system, but we have partitioned the data into three parts and loaded it into Stardog, Amazon Redshift in AWS, and SQL Server in Azure. Stardog supports connectors for over 100 different data stores. This benchmark environment corresponds to how an enterprise may have their data stored for a Knowledge Graph: part in a graph database and the rest in 2 common databases across 2 cloud providers.



The BSBM dataset comes both in an RDF graph version and a relational version. We have used the RDF representation[1] for materializing the graph data in Stardog and used the relational for loading the data into Redshift and SQL Server. We have then defined virtual graph functionality in Stardog using the **Stardog Mapping Syntax to map the relational data sources to RDF format**. As a result, no data is moved from relational sources into Stardog, but Stardog can answer SPARQL queries submitted by users, converting all or parts of the query into SQL automatically for external data sources.

The size of the data distributed over the three data sources looks as follows:

Data Source	Graph Type	Number of Nodes	Number of Edges	Amount of data
Stardog	Materialized	**8.8 billion**	115 billion	6.1TB
SQL Server	Virtualized	**30 billion**	220 billion	4.5TB
Redshift	Virtualized	**57 billion**	660 billion	2.8TB

Of course, there are no actual graph nodes or edges in the relational data sources, but if we transformed the relational data into the graph representation, we would get the number of nodes and

[1] Stardog uses semantic graphs to create meaning by mapping entities, their metadata, and their relationships. Semantic graph, also called RDF graph, is the only way to represent data that is natively stored in other structures while maintaining all relevant metadata and context.

edges as shown in the table. The sizes of disks on the index are not directly correlated with the number of nodes and edges as seen in the table. Longer textual fields in the product and review data cause the corresponding index to take more space on disk compare to offers.

Synthetic Data :)

Dataset Generation

What is character of the data? Is it all integer? How many properties etc...

We used the open-source BSBM data generator to generate the data. We had to make some minor adjustments to that tool. First, the data generator used 32-bit integers in several places to represent the count of objects that will be created. This limits the total number of triples generated so we changed the code to use 64-bit integers so we can generate much more data.

The data generator can output data either in RDF format or as SQL commands for MySQL or MonetDb. We have created the product data as RDF files that were loaded into Stardog using the db create command and bulk load memory mode.

But it was not feasible to generate the entire dataset for Redshift and SQL Server, because just the generation of files would take weeks. We modified the SQL data generator to load the data into relational databases in a more efficient way. For SQL Server, we used its driver-specific API to load data directly from the generator and made the data generator to be multi-threaded. For Redshift, we used the bulk copy command to stream the data as CSV files from the data generator using 8 threads.

Benchmark Queries

The BSBM benchmark comes with an *Explore* use case that contains a mix of queries simulating the search and navigation pattern of a consumer looking for a product. The queries in BSBM benchmark are templates and not executable by themselves. The query driver program injects values into the query templates using the metadata files created by the data generator to construct executable queries.

The following table shows the list of the queries defined in the benchmark, which data sources each query uses and a short description of the query.

Queries	Data Sources	Description	
1	Stardog	Find products for a given set of generic features	Search
2	Stardog	Retrieve basic information about a specific product for display purposes	Search + List properties.
3	Stardog	Find products having some specific features and not having one feature	Filter.
4	Stardog	Find products matching two different sets of features	Search Filter

7	Stardog, Redshift, SQL Server	Retrieve in-depth information about a specific product including offers and reviews.
8	SQL Server	Give me recent reviews in English for a specific product.
9	SQL Server	Get information about a reviewer
10	Redshift	Get offers for a given product which fulfill specific requirements
11	Redshift	Get all information about an offer
12	Stardog, Redshift	Export information about an offer into another schemata

Search, Maybe expand -

expand?.

Cloud Environments

Our benchmark included two different environments—one for Amazon Redshift and one for Azure SQL Server along with the Stardog deployment itself.

Platform	**AWS Redshift**	**Azure SQL Server**	**Stardog**
Nodes	2 nodes (minimum required)	"Hyperscale" config (required for dbs. larger than 4TB); no replicas	Single node
Loading	"ra3.4xlarge" (per node: 12 vCPU 96 GiB) then scaled back down	8 vCores and 41.5 GB mem and 256 GB tempdb temporarily scaled up when building index after main loading phase. 80 vCores and 415 GB mem and 2.5 TB tempdb (750 GB was used)	m5.12xlarge EC2 instance (192GiB RAM, 48 vCPUs)
Querying	"ra3.xlplus" (per node: 4 vCPU 32 GiB)	2 vCores and 10.4 GB mem	x1.16xlarge EC2 instance (976GiB RAM, 64 vCPUs)

Benchmark Results

So not "worst-case"

The following table shows the **average query execution times** for each query. Before running the actual tests, we ran 500 iterations of random query mixes to warm up the system and caches. We have then executed 20 query mixes (with 25 random query instantiations in each mix) without caching to compute the average execution times for each query. All average query execution times for each query was under one second (except one).

Queries	Average Number of Results	Average Execution Time (sec)
1	10.0	0.445
2	18.3	0.010
3	9.8	0.769
4	10.0	1.107
7	10.9	0.049
8	3.4	0.034
9	6.0	0.017
10	1.3	0.015
11	10.0	0.015
12	8.0	0.022

Comparison with Other Knowledge Graph Demonstrations

Our results are the **first** demonstration of a distributed Knowledge Graph implementation at the scale of 1 trillion edges. Typical graph database benchmarks are much smaller in scale. The largest benchmarks that have been reported for Ontotext GraphDb (17 billion edges), Neo4J (20 billion edges), and TigerGraph (67 billion triples) are an order of magnitude smaller.

Several RDF-based systems have published benchmarking results for graphs with 1 trillion triples, for example Cambridge Semantics, Oracle and Cray. There are some key differences between the results we achieved here and what has been published by other vendors.

Yes it does nicely articulate what Stardog can do...

Distributed, i.e., Real-World Data

In all the previous benchmarking results **all the data was loaded into a single location**. Copying all the data into a single database approach is practically indistinguishable from data warehousing. In contrast, our setup queries data where it was designed to be stored without the need to create new copies making data lineage and traceability straight-forward, as well as speeding time to insight for customers.

Look at this.

Randomized Queries

All three benchmarks mentioned above use the Lehigh University Benchmark (LUBM). There are 14 fixed queries in the LUBM benchmark. The exact same queries are executed multiple times during the benchmarking making it much easier to cache results. **In contrast no query is executed more than once in BSBM as each query mix is completely randomized.** It's a better simulation of an enterprise workload.

Inference

The LUBM benchmark is mostly focused on inference capabilities. In each of the previous vendor benchmarks, the base graph data loaded into the graph databases was significantly lower than one trillion. For example, Oracle reports that 605.4 billion edges were loaded and another 475 billion edges were generated as a result of inference. This kind of materialization approach obviously limits the scalability of the system. Stardog uses a novel query-time reasoning approach that does not require generating inferred data at load time, which lets Stardog do query-time inference with distributed and virtualized data. Our BSBM benchmark does not focus on inference, although we have tested inference capabilities of Stardog with our setup and were able to run inference queries in similar times.

Cost of Operations

All of the previous benchmark results take advantage of very large number of servers to achieve the trillion-triple scale. For example, the Cambridge Semantics benchmark uses a cluster of **200 n1-highmem-32 type server instances** in the Google Compute Platform. Each server has 208GB memory and comes with 32 vCPU's, which correspond to 32 Intel hyper threads on 16 hardware cores. At the time of this writing, the n1 family of Compute Engines cost \$0.031611/vCPU-hour. Therefore, the Cambridge Semantics cluster would cost **\$202/hour**.

In contrast, we used a single Stardog server with 192GB memory to load the data and switched to a machine with 976GB for queries. **The server we used for Stardog costs \$6.6/hour whereas AnzoGraph cluster costs \$378/hour.** Even when the cost of Redshift and SQL Server are taken into account, (which is an additional \$2/hour), our distributed setup has an order of magnitude lower operational costs than the AnzoGraph setup, which of course under-estimates the ongoing operational cost differences between operating a single server and a 200-node cluster with respect to devops and other support personnel.

Conclusion

In this report we have shown that Stardog allows users to query trillion-edge graphs distributed over multiple data sources in different cloud providers. The combination of materialization and virtualization capabilities give companies the option to store data in Stardog when needed but leave other data in its desired data store and to be queried on-demand. Average query execution times below one second show that performance at this scale is in line with fully materialized enterprise queries, as is the cost.

Stardog has made data location irrelevant to Knowledge Graphs. It has finally given companies the ability to perform high-volume, realistic enterprise Knowledge Graphs, which can accelerate knowledge discovery across a wide range of assets or processes a company wants to optimize without limitation.

About McKnight Consulting Group

With a client list that is the "A list" of complex, sustainable and successful information management, McKnight Consulting Group (MCG) has broad information management market touchpoints. MCG advice is an infusion of the latest best practices culled from recent, personal experience. It is practical, not theoretical.

MCG anticipates its customer's needs well into the future with a full lifecycle approach. The focused, experienced teams generate efficient, economic, timely and politically sustainable results for their clients.

MCG services span strategy, implementation and training for turning information into the asset it needs to be for your organization. MCG strategizes, designs and deploys in the disciplines of Master Data Management, Big Data, Data Warehousing, Analytic Databases and Business Intelligence.

About Stardog

Stardog was founded in 2015 on the vision of powering the connected enterprise. Stardog's Enterprise Knowledge Graph technology turns data into knowledge to enable more effective digital transformations. With Stardog, customers reduce data preparation timelines by up to 90 percent by transforming enterprise data infrastructure into a comprehensive end-to-end data fabric. Industry leaders including BNY Mellon, Bosch, and NASA use Stardog to create a flexible data fabric that can support countless applications. Stardog is a privately held, venture-backed company headquartered in Arlington, VA. For more information, please visit www.stardog.com or follow them @StardogHQ.