Graph Analytics in Healthcare Operations

A case study of TigerGraph on Dell EMC Infrastructure

December 2020

White Paper

Abstract

This white paper discusses a collaboration between Dell Technologies, TigerGraph, and a major health care organization to determine the maximum performance that could be achieved by running TigerGraph on Dell EMC PowerEdge servers with AMD EPYC processors for several critical use cases using graph analytics in healthcare operations. After presenting background on graph technology, we explain the performance test methodology and results obtained during a proof of concept by Dell Technologies' Systems Engineering team in AI and Data Analytics on a database with over 100 million patient records, with results of up to ten times improvement in performance over other approaches.

Data-Centric Workloads & Solutions

Copyright

The information in this publication is provided as is. Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Dell Inc. believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

Contents

Introduction

Leading-edge healthcare organizations are, without question, intensely data driven. And the data sets are very large, highly complex, and come from a wide variety of sources, including administrative data, patient medical records, claims history, clinical data, treatment efficacies, pharmaceutical guidelines, and research and clinical trial information, to name just a few of the sources.

Fortunately, the healthcare industry is experiencing tremendous change with respect to technology, and the collective field of artificial intelligence (AI) and data analytics is one of the most significant and impactful areas of technology that is being applied to healthcare, in order to process and make use of these massive amounts of data – for benefits in patient care, customer service, and cost control.

At the forefront of advanced analytic technologies, graph databases and graph analytics are among the fastest growing. Graph technology is a set of analytic techniques that allows for the exploration of relationships between extremely large volumes of data with a speed and efficiency not easily achieved with traditional databases or analytics.

According to Gartner, graph analytics is one of the top ten trends in data and analytics for 2020 and will grow at 100 percent annually through 2022, making it one of the fastest growing markets in data and analytics. By 2023, graph technologies will facilitate rapid contextualization for decision making in 30% of organizations worldwide[1].

When combined with machine learning (ML) algorithms, graph technologies can be used analyze thousands of data sources with millions or billions of elements in order to help healthcare professionals make more rapid *and* more accurate decisions in both patient care and in administrative operations.

Because of the complexity of health data and the number and depth of the interconnections between the many data sets, traditional tabular data structures of relational databases are not always a good fit. But graph technology is particularly well suited to healthcare data analytics.

About this document

Recently Dell Technologies partnered with a large health care organization and TigerGraph, the only scalable graph database for the enterprise, to do a proof of concept (POC) in order to determine the maximum performance that could be achieved by running TigerGraph on Dell Technologies infrastructure for several critical healthcare use cases.

While graph analytics is being applied across many different vertical markets, including financial services, supply chain management, cybersecurity, e-commerce, media and entertainment, and energy management, the focus of this paper is on the application to healthcare.

This document presents background on graph analytics technology, shows the performance test setup and procedures used in the POC, and explains the results

[1] *Gartner Top 10 Trends in Data and Analytics for 2020*. Smarter With Gartner, 2020. https://www.gartner.com/smarterwithgartner/gartner-top-10-trends-in-data-and-analytics-for-2020/.

obtained from the performance testing by the Dell Technologies' Systems Engineering team in AI and Data Analytics.

Audience

This white paper is for IT directors, system administrators, system architects, and data scientists in the healthcare field who are interested in understanding graph analytics, how it applies to healthcare, and the types of results that can be achieved for several key use cases.

We value your feedback

Dell Technologies and TigerGraph welcome your feedback on this white paper and the information presented herein. Contact the Dell Technologies Solutions team by email or provide your comments by completing our documentation survey.

Note: For links to additional documentation for this solution, see the Dell Technologies Solutions Info Hub for Data Analytics.

Graph technology background

Graph analytics is the technology that allows for the deep exploration of connected data, tracing the complex interrelationships among various entities, such as organizations, people, transactions, records, but most importantly: in context. Graph databases excel at answering complex questions about relationships in large data sets, which is not always practical or even possible at scale using SQL queries.

Graph databases are purpose-built for storing and analyzing relationships among data, since the data entities, as well as the relationships among them, are pre-connected and there is no need for specialized programming experts to execute time-consuming nested queries that require expensive table joins or multiple scans across large tables.

The latest graph technology can traverse 10 or more hops – a four- or five-fold increase over earlier generations. This opens up a whole new world of information for fraud detection, recommendation engines, artificial intelligence, machine learning, and many other use cases – including healthcare.

Graph vs conventional databases

Relational databases fall short for analyzing very large data sets with complex and multiple relationships because their architectures simply aren't designed for this level of analytics. They store the data for each business entity such as customer, order, product, and payment data in separate database tables. To understand and analyze relationships across the business entities, relational databases require table joins, which can take hours, even days for the complex joins and are computationally expensive as the size of the data grows.

NoSQL databases, including key-value databases, typically store all of the data in a single data lake. This means that queries and algorithms to perform the relationship analysis requires iterative full-table scans across millions or billions of rows, making it very difficult to perform a deeper analysis of the relationships beyond two or three levels. Performance of NoSQL as a backend to a graph analytics engine encounters multiple sources and levels of impedance compared to a native graph database.

The figures below illustrate these points. In Figure 1, the relational database has a rigid schema and, while it may have high performance for transactions, it has poor performance for deep analytics as the relationships are calculated at query time and table joins are required. The key-value database has essentially no schema or a highly fluid schema, high performance for simple transactions, but poor performance for deep analytics as multiple scans of a massive table are required for large data sets.

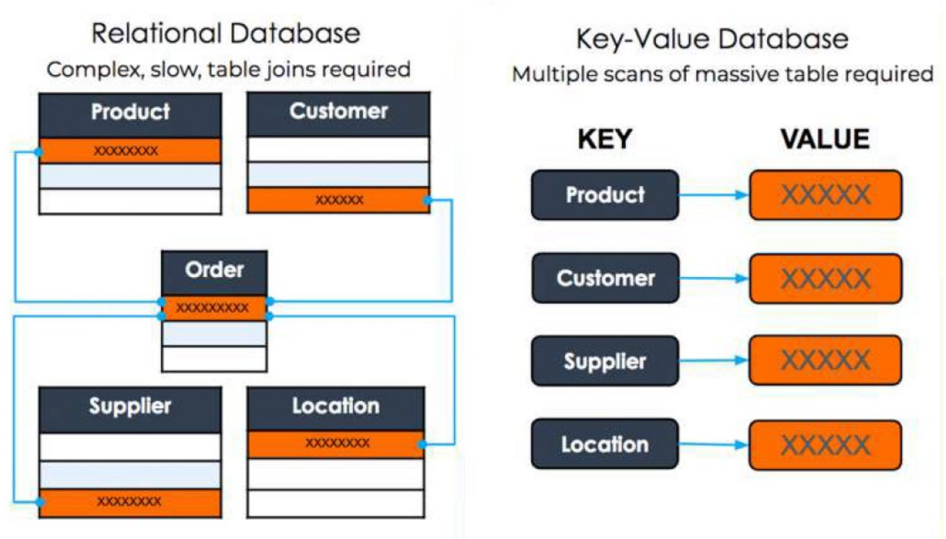

Figure 1. Relational and Key-Value Database Concepts

In Figure 2, because the graph database has pre-connected entities and a flexible schema, it has high performance both for complex transactions and for deep flexible analytics.

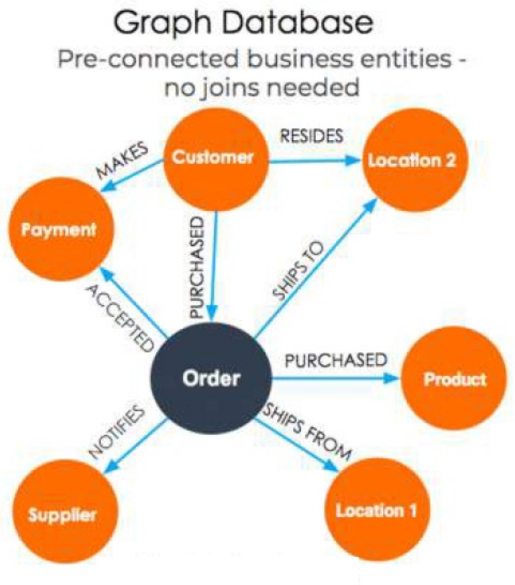

Figure 2. Graph Database Concept

TigerGraph

TigerGraph is a scalable graph database system by the company of the same name, built from the ground up to support massively parallel computation of queries and analytics. TigerGraph is a native parallel graph database, with its proprietary storage designed from the ground up to store graph nodes, edges, and their attributes, with an engine that computes queries and analytics in massively parallel processing (MPP) fashion for significant scale-up and scale-out performance.

TigerGraph allows developers to express queries and sophisticated graph analytics using a high-level query language called GSQL (Graph Structured Query Language). GSQL is designed for compatibility with SQL, while simultaneously allowing NoSQL programmers to continue thinking in Bulk-Synchronous Processing (BSP) terms and reap the benefits of high-level specification. As a Turing-complete language, GSQL is sufficiently high-level to allow declarative SQL-style programming, yet sufficiently expressive to concisely specify the sophisticated iterative algorithms required by modern graph analytics and traditionally coded in general-purpose programming languages like C++ and Java.

TigerGraph Architecture

A high-level view of the TigerGraph Architecture is shown in Figure 3, showing the core platform architecture, along with some of the numerous sources of operational and master data, and the many functional areas that are beneficiaries.

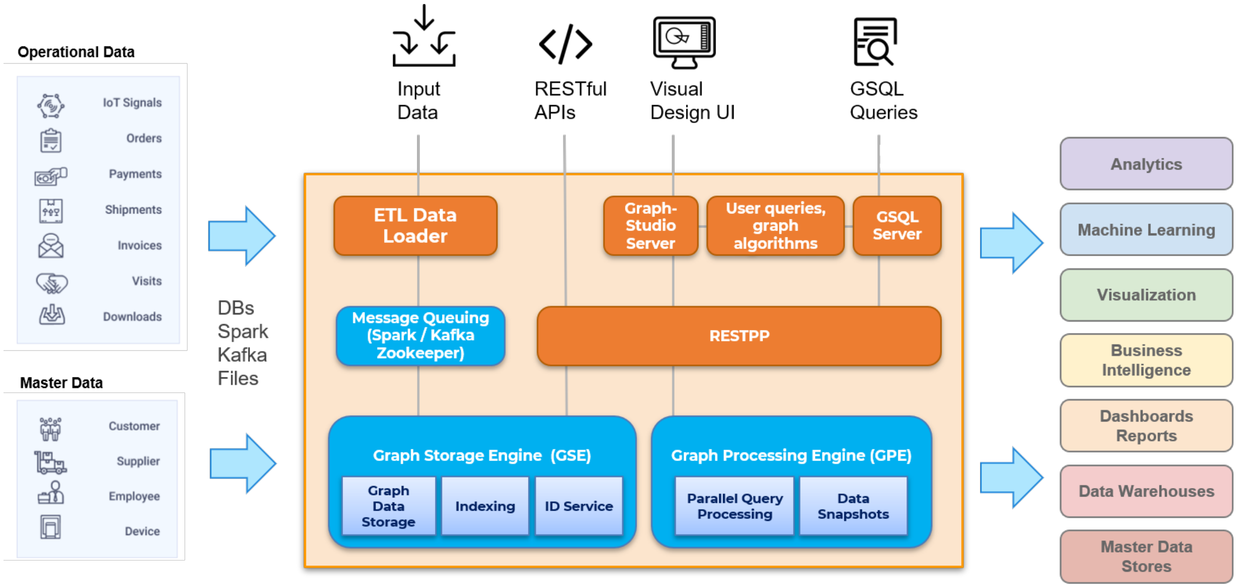

Figure 3. TigerGraph Architecture

Some of the unique features, design differences, and benefits of TigerGraph are described in Table 1.

Table 1. TigerGraph Platform Architectural Differentiation

Feature	Design Differences	Benefits
Real-time deep link queries (5 to 10+ hops deep)	• Native parallel graph for correctness and efficiency • C++ engine for category-leading performance • Massively parallel processing for cloud-scale	• In-graph concurrent transactions, deep-link analytics, and AI/ML at scale

Feature	Design Differences	Benefits
Massive scale	• Distributed architecture with continuous availability • Efficient graph storage reduces memory footprint	• All enterprise and external datasets supported • Automatic partitioning and active-active high availability (HA)
In-database analytics	• GSQL: Turing-complete SQL-like query language enables unique analytics features such as accumulators and user-extensible graph algorithm library	• Strong consistency *and* graph analytics in a single logical cluster even across multiple datacenters simplifies deployment and accelerates time to insight

Graph analytics use cases in health care

How does graph technology apply to healthcare? The bulk of the current tools for storing and analyzing healthcare data are built on relational databases. These databases store the data for each entity such as claim, provider, member, and facility (e.g., hospital or treatment center) in separate tables or even separate databases. In order to understand the relationships among members, providers, facilities and the claims connecting them to each other, for example, all of these tables or databases must be joined together. As the size and complexity of the data grows, database table joins become time-consuming and computationally expensive, thereby making the relational database an impractical solution for understanding and analyzing relationships.

A native parallel graph, such as TigerGraph, is built to explore and analyze the complex relationships in healthcare data, allowing data scientists and business users to go many levels deep, across billions of records and millions of members and providers, providing results in near real-time.

The book, *State of Healthcare Technology*[2], discusses the application of graph analytics and other advanced technologies to the healthcare field and defines at least five use cases currently in practice or under investigation. These healthcare use cases for graph technology include the following:

- **Ontology-based data access** – An ontology is a model that represents the properties and relationships of a subject area, such as medical terminology. Graph technology supports the storage and accessing of voluminous medical concepts and terminologies through ontology-driven rules systems, which is essential for uniform communications in the medical field.

- **Single view of the patient** – This use case allows a representative of a healthcare organization, such as a call center agent, to see the entire history and all interactions with a patient in a fraction of a second, in order to provide accurate and responsive customer service.

- **Patient similarity and cohort building** – Using a feature in graph technology called a similarity calculation, this allows a care provider such as a doctor or nurse, upon assessing a patient, to determine the best treatments and outcomes for similar patients, or to identify cohorts of similar patients for analysis.

[2] Holley, Kerrie, et al. 2020. *The State of Healthcare Technology*. O'Reilly.

- **Real-time clinical decision support and recommendations** – Related to the patient similarity use case, graph-based algorithms can also suggest care paths or alternatives for physicians in a clinical setting or for researchers.
- **Fraud detection** – Graph technologies are commonly used to build accurate fraud-detection models, which is highly applicable to billing operations and claims processing in healthcare.

Proof of concept

The POC was designed collectively by the healthcare organization (customer), Dell Technologies, and TigerGraph, and implemented by the Dell Technologies' Systems Engineering team in AI and Data Analytics.

Healthcare organization

The customer is a very large, innovative, and technologically advanced health services organization that provides health care services, health benefit plans, and insurance and financial services.

The characteristics of the organization and its data landscape include the following:

- Currently the largest connected healthcare graph data asset in the US, with 10B+ vertices and 50B+ edges
- Contains data from 100M members
- Houses 18 months of current data from members, claims, clinical interactions, providers, phone calls, house calls, and more
- Graph database contains 1.2TB of data
- Support over 33,000 online users on various applications

One of our customer's key goals is to provide validated carepath recommendations as quickly and efficiently as possible via their customer service agents. By delivering better and more efficient guidance in real time, their goal is for a typical 20-minute call to be 10% shorter, with better customer satisfaction metrics. In doing so, we could help them save upwards of $100M in call center savings by being able to deliver more efficient and effective guidance to members on every call.

Target use cases

Three specific use cases, some of which were based on the general cases discussed above, were selected for implementation and performance testing as part of the POC.

1. **Patient record retrieval** – This case is equivalent to the "single view of the patient" as discussed above and is also known by the healthcare organization as the Member Journey. In this scenario, the consumer (member) calls in to a customer service agent or nurse hotline for information related to their account or condition. This query takes the member ID as an input parameter and pulls up their complete health record. Today this takes 100ms for the healthcare organization
2. **Find a provider** – This is also a customer service use case, but different from the patient record retrieval. Here the member calls into the customer service agent or nurse hotline with a specific ailment and needs to find the closest doctor and

facility to them who are qualified to treat the issue. The input parameters are the member ID, the procedure and the member's location.

3. **Patient similarity** – In this case, also known as a medical twin, a doctor who is treating a patient wants to analyze other patients with characteristics like their own to determine a treatment regimen. To do that the doctor is looking for the most similar patients that match their patient's medical history and who had successful outcomes. The input parameters here are the member ID, the procedure code(s), and the number of similar patients to return.

Test environment

The test environment consisted of the following Dell EMC PowerEdge servers:

- (1) Client Node, consisting of a Dell EMC PowerEdge R6525 rack server with 2nd generation AMD EPYC™ 7702 processors, 128 cores, and 2 TB of memory
- (1-8) Compute Nodes, consisting of Dell EMC PowerEdge C6525 C-series servers with 2nd generation AMD EPYC 7702 processors, 128 cores, and 1 TB of memory each

Figure 4 shows a representation of the test environment. While this view shows two Compute nodes as an example, the actual environment consisted of one, four, or eight compute nodes, based on the phase of the testing, as explained below. The maximum configuration was eight compute nodes.

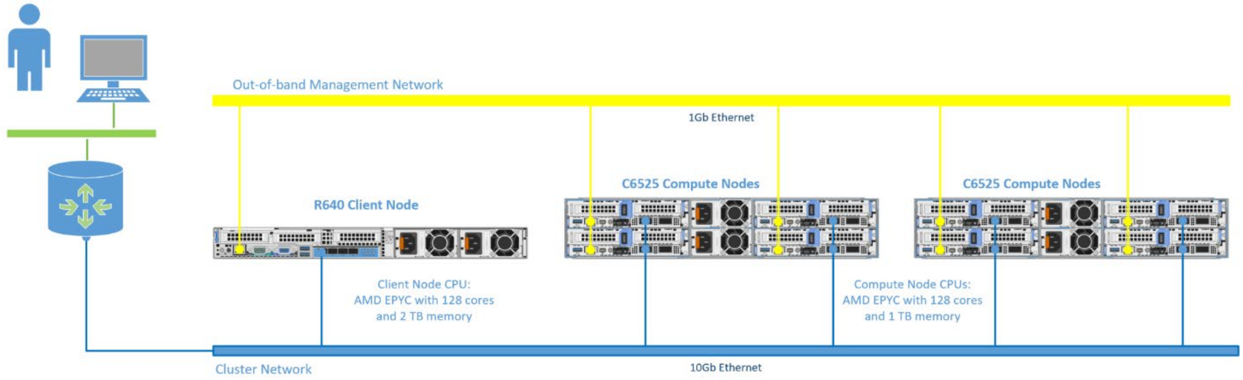

Figure 4. Performance test environment

Methodology

After setting up the physical test environment, since we could not use actual patient data, we generated a data set that replicated the size and characteristics of the customer using Synthea™, an open-source, synthetic patient and medical history generator.

We designed a set of single query executions using GraphStudio™, an interactive graph data analytics user interface from TigerGraph, and designed multiple parallel stress test queries using Apache JMeter™, an open source load and stress test application.

And we performed observations and measured results using JMeter's queries per second (QPS) reporting and Dell EMC LiveOptics, a free online software application used to collect, visualize, and share data about IT environments and workloads.

The POC was undertaken in three phases of test execution.

Phase 1

We began in phase 1 with a single Dell EMC PowerEdge C6525 compute node, a database of 11 million patients, a maximum number of parallel queries of 1250, and a QPS rate of 589 transactions per second. This phase helped us to establish a baseline. The query execution time results are shown in Table 2.

Phase 2

For phase 2, we expanded to four compute nodes, 11 million patients, a maximum number of parallel queries of 5000, and a QPS rate of 744 transactions per second. Upon obtaining stable results and recording the outcomes, we expanded the database to 42 million patients. The query execution time results are shown in Table 2.

Phase 3

This phase was the culmination of the performance testing, with an 8-node cluster; 104 million patients; 25,000 maximum parallel queries; and a QPS rate of 437 transactions per second.

Figure 5 shows the maximum CPU utilization under these conditions using LiveOptics, which yielded the results shown in Table 2.

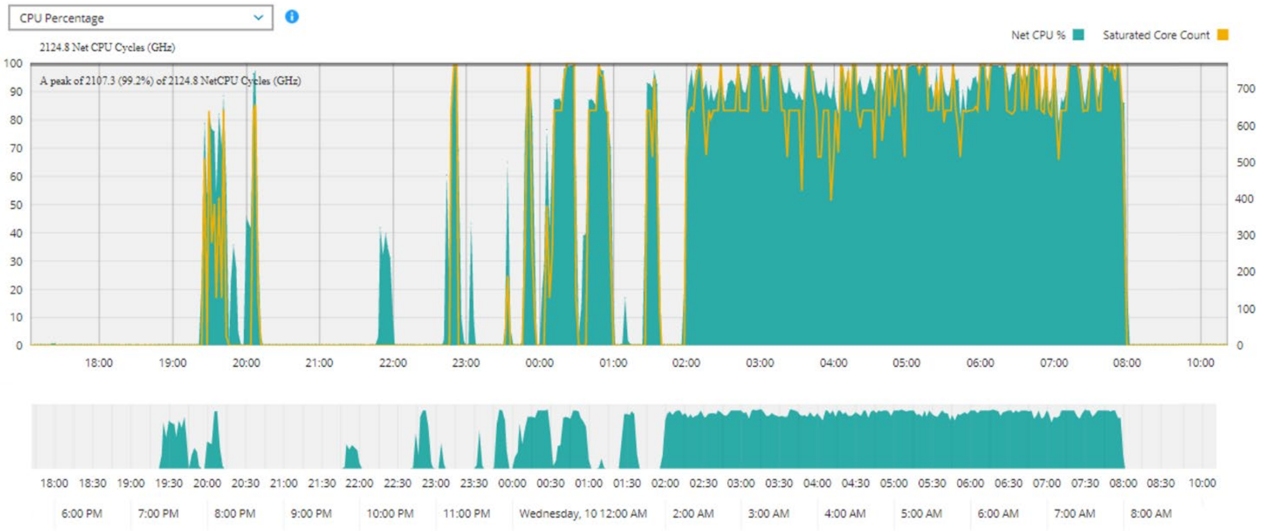

Figure 5. CPU utilization for highest-performing use case

Results

The performance results significantly exceeded the expectations of the customer and are show in Table 2 below.

Table 2. Performance test results

Use Case	Query Execution Time			
	11M Patients 1-Node Cluster	**11M Patients 4-Node Cluster**	**42M Patients 4-Node Cluster**	**104M Patients 8-Node Cluster**
1. Patient record retrieval	8.5 ms	4.6 ms	6.0 ms	10 ms
2. Find a provider	62.7 ms	8.9 ms	12.0 ms	1600 ms
3. Patient similarity	7 sec (1.1 B edges)	3.4 sec (1.1 B edges)	49 sec (3.3 B edges)	1 min (7.4 B edges)

For the patient record retrieval in the maximum scenario, the customer's goal was to achieve a query execution time for the patient record retrieval use case of better than 100 ms. With a result of 10 ms against a 104 million patient database, **we achieved an improvement of 10 times the expected performance** by the healthcare customer. Results for the find a provider and patient similarity use cases correlated closely and yielded results of the same magnitude.

Conclusion

The results of the proof of concept by Dell Technologies and TigerGraph for these three real-world healthcare use cases yielded results far beyond the expectations of the customer, not only due to the sheer speed of the queries, but because they enable higher quality information that leads to more confident recommendations. This will help put the organization well on their way to achieving the targeted reduction in customer service call times and their attainment of the $100 million savings, based largely on the patient record retrieval query time improvements.

Even more important than cost savings is the improvement in customer care made possible with graph analytics for all of the healthcare use cases. In these times of historic pandemic, rapid and high-quality healthcare *and* customer service are important for each and every patient interaction. Enabling a customer service agent to rapidly find the right provider for a member or enabling a doctor to accurately find a similar patient or medical twin for a doctor, can make all the difference. The combination of TigerGraph analytics and Dell Technologies infrastructure is the perfect example of using technology to advance human progress.

References

The following links provide additional information on topics covered in this paper:

- AMD Partner Website for Dell EMC Solutions
- Dell Technologies Info Hub for Data Analytics
- Dell Technologies Healthcare Solutions
- Gartner Top 10 Trends in Data and Analytics for 2020
- State of Healthcare Technology eBook
- TigerGraph website
- TigerGraph: A Native MPP Graph Database Technical Paper