Hewlett Packard Enterprise

Reference Architecture

HPE Reference Architecture for Graph Analytics on HPE ProLiant DL385 Gen11 Server with TigerGraph

Using Supply Chain use case

CONTENTS

EXECUTIVE SUMMARY

The supply chain industry is experiencing a growing need for advanced analytics solutions to optimize operations, enhance visibility, and improve decision-making. Graph analytics has emerged as a powerful tool for addressing complex supply chain challenges by modeling relationships, dependencies, and processes as interconnected graphs.

Graph analytics empowers organizations to unveil relationships within interconnected data on a large scale and in real-time, offering unprecedented capabilities. Serving various purposes such as enhancing revenue, cutting costs, boosting customer satisfaction, and elevating employee productivity. With Gartner predicting that by 2026, 65% of businesses will shift their decision-making from being intuition-based to data-driven, implementing graph technologies can be a strategic asset for organizations needing to accelerate their own data journey. Many data-driven organizations will seamlessly integrate technology into their data infrastructure to support graph analytics, whether it's within a specific department or spanning the entire enterprise. The ability to merge data perspectives, scale, and velocity is of utmost importance for companies to address relationship-based inquiries effectively while driving their business forward in today's competitive landscape.

Document purpose: This Reference Architecture provides comprehensive architectural guidelines and implementation of Supply chain optimization on AMD EPYC powered HPE ProLiant DL385 Gen11 servers using TigerGraph. Organizations can readily apply the blueprint tested in this document to implement a graph analytics solution quickly.

Target audience: The target audience for this performance report is the IT community studying solutions for their environments. Business users and IT professionals who are interested in implementing a Graph Analytics solution may find this report useful for a sample TigerGraph configuration.

This Reference Architecture describes solution testing performed in August and October 2023.

INTRODUCTION

A graph database is designed to facilitate analysis of relationships in data. A graph database stores data as entities and the relationships between those entities. It is composed of two things: vertices and edges. Vertices represent entities such as a person, product, location, payment, parts, order and so on; edges represent the relationship between these entities.

Graph analytics explores these connections in data and reveals insights about the connected data. These insights can help prevent fraud, enable better product recommendations, or help improve supply chains.

The TigerGraph Native Parallel Graph offers transformational technology, with significant advantages over the most well-known graph database solutions on the market.

Despite its comprehensive and well-documented graph database functionality, the current leading solution in the market is considerably slower in comparison. TigerGraph, however, runs much faster due to its massive parallel processing capability and native scale out architecture design. In benchmark tests, TigerGraph can load a batch of data much faster in comparison with other graph databases in the market.

Further, by offering parallelism for large scale graph analytics, TigerGraph supports graph parallel algorithms for Very Large Graphs (VLGs) – providing a considerable technological advantage as graphs inevitably grow larger. It works for limited, fast queries that touch anywhere from a small portion of the graph to millions of vertices and edges, as well as more complex analysis that must touch every single vertex in the graph itself. Additionally, real-time incremental graph updates make it suitable for real time graph analytics unlike other solutions.

The TigerGraph advantage lies in the fact that TigerGraph represents graphs as a computational model. Compute functions can be associated with each vertex and edge in the graph, transforming them into active parallel compute-storage elements, in a behavior identical to what neurons exhibit in human brains.

Vertices in the graph can exchange messages via edges, facilitating massively parallel and fast computation. The Native Parallel Graphs (NPG) offers a completely new computation paradigm which was absent from previous models, making it poised to become a truly transformational technology.

TigerGraph represents a new generation, distributed computational model, and offers the following advantages:

- Faster data loading to build graphs quickly.
- Faster execution of graph algorithms
- Real-time capability for streaming updates and insertions

- Ability to unify real-time analytics with large-scale offline data processing.
- Ability to scale up and scale out for distributed applications.

It's easy to develop a solution on top of the TigerGraph platform.

Key solution benefits

- Better, faster queries and analytics - superior performance for querying related data, big or small.
- Simpler and more natural data modeling – easy and free to define semantic meaning to represent relationships.
- More powerful problem-solving – solve problems that are both impractical and practical for relational queries.
- Offer real-time speed at scale – achieved by distributed architecture with concurrent querying and data updates in real-time.
- Enhanced machine learning and AI - allow for further infer indirect facts and knowledge by providing an excellent source of previously overlooked features.

SOLUTION OVERVIEW

This Reference Architecture provides guidance for installing TigerGraph on HPE ProLiant DL385 Gen11. The solution consists of three (3) HPE ProLiant DL servers; and the same hardware setup was used for the testing supply chain optimization use case testing. The servers in this solution are interconnected with 100Gb/s network communication using HPE Aruba 8325

The diagram of the hardware components that form the solution are shown here in Figure 1, inside a rack.

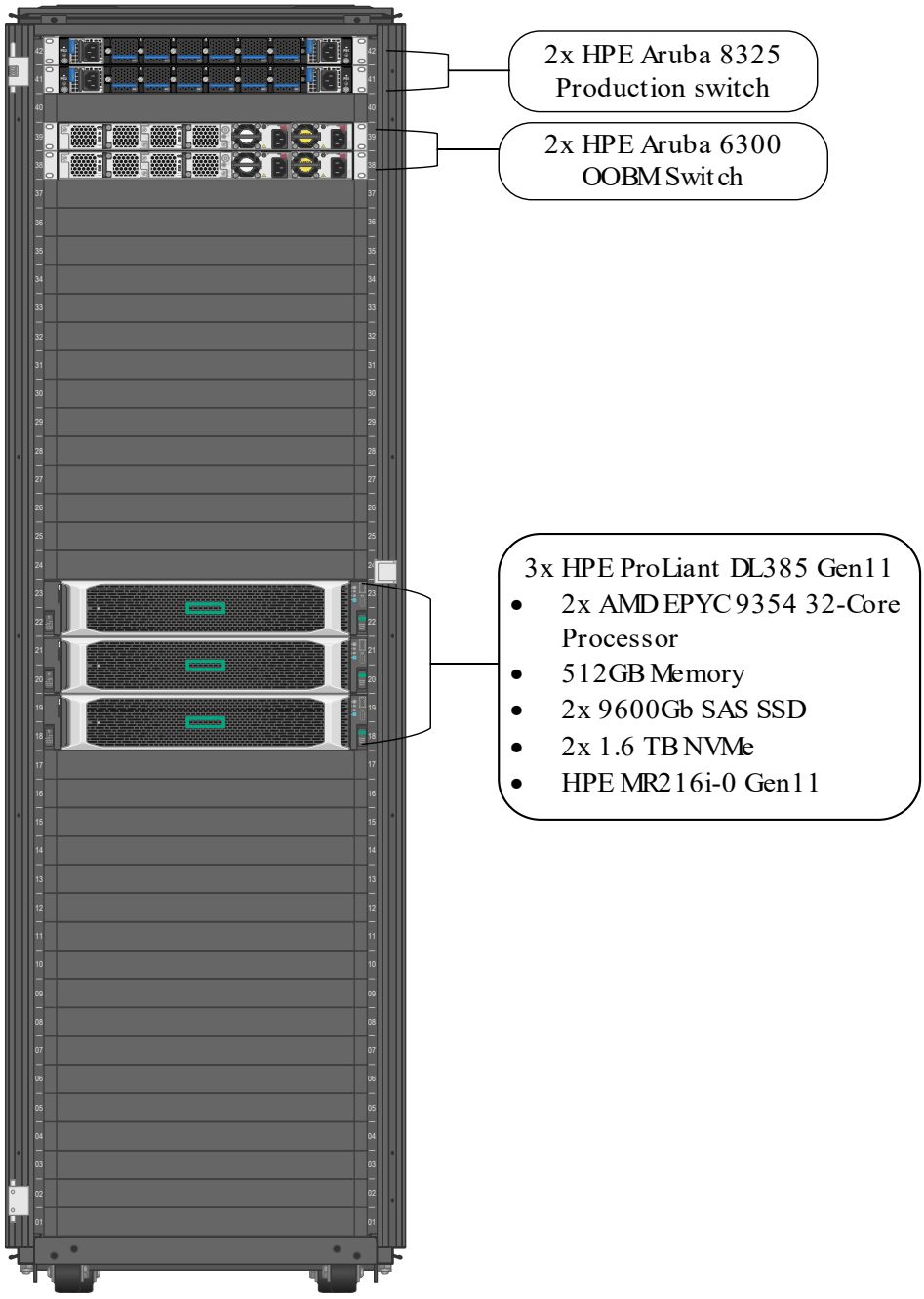

FIGURE 1. Solution for TigerGraph on HPE ProLiant DL385 Gen11

SOLUTION COMPONENTS

Hardware

HPE ProLiant DL385 Gen11 server

HPE ProLiant DL385 Gen11 server is an accelerator-optimized 2U 2P solution that delivers exceptional compute performance, upgraded high-speed data transfer rate and memory depth at 2P compute capability. Powered by 4th Generation AMD EPYC™ 9004 Series Processors with up to 96 cores, increased memory bandwidth (up to 6TB total DDR5 memory with 256GB memory DIMMs), up to 128 lanes of high-speed PCIe

Gen5 I/O, Gen5 EDSFF storage and the newly designed chassis supporting 8 single wide (SW) or 4 double wide (DW) GPUs*. The HPE ProLiant DL385 Gen11 server is a perfect accelerator-optimized 2U 2P solution.

What's new

- Adding two new 4th Generation AMD EPYC™ Series Processors with 128 and 112 cores (9754 & 9734)
- New CTO servers including EDSFF, GPU and 48SFF CTO servers.
- New GPUs are supported with the selection of GPU CTO Server - Nvidia H100, L40 and L4.
- Direct Liquid Cooling (DLC) is now supported.
- 12 DIMM channels per processor for up to 6TB total DDR5 memory with the 256GB memory DIMMs.
- Advanced data transfer rate and higher network speed from the PCIe Gen5 serial expansion bus.
- New HPE Integrated Lights-Out 6 (iLO 6) server management software.
- Supports hot-pluggable, high-availability RAID M.2 boot options - NS204i-u.
- OpenBMC Capable through iLO6 Transfer of Ownership Process.

FIGURE 2. HPE ProLiant DL385 Gen11 server

HPE iLO

HPE Integrated Lights Out (iLO) is embedded in HPE ProLiant platforms and provides server management that enables faster deployment, simplified lifecycle operations while maintaining end-to-end security thus increasing productivity.

HPE Aruba 8325 Switch Series

HPE Aruba Networking CX 8325 Switch Series offers a flexible and innovative approach to addressing the application, security, and scalability demands of the mobile, cloud and IoT era. These switches serve the needs of the next generation core and aggregation layer, as well as emerging data center requirements at the Top of Rack (ToR) and End of Row (EoR). They provide over 6.4Tbps of capacity, with line-rate Gigabit Ethernet interfaces including 1Gbps, 10Gbps, 25Gbps, 40Gbps, and 100Gbps.

The Aruba CX 8325 series includes industry-leading line rate ports 1/10/25GbE (SFP/SFP+/SFP28) and 40/100GbE (QSFP+/QSFP28) with connectivity in a compact 1U form factor. These switches offer a fantastic investment for customers wanting to migrate from older 1GbE/10GbE to faster 25GbE, or 10GbE/40GbE to 100GbE ports.

Features

- High performance 6.4Tbps with 2,000Mpps throughput
- High availability with industry leading VSX redundancy, and redundant power supplies and fans
- Designed for core/aggregation in the campus or Top of Rack (ToR) or End of Row (EoR) in the data center.

- HPE Aruba Storage Optimize, validated Ethernet Storage Fabric solution with HPE Storage and HCI
- AOS-CX automation and programmability using built-in REST APIs and Python scripts.
- Advanced Layer 2/3 feature set includes BGP, OSPF, VRF-lite, and IPv6
- Intelligent monitoring, visibility, and remediation with Aruba Network Analytics Engine
- Aruba NetEdit support for automated configuration and verification
- Compact 1U switches with 1/10/25GbE and 40/100GbE connectivity
- Supports Aruba Fabric Composer - a software-defined orchestration solution that simplifies and accelerates leaf-spine network provisioning and day-to-day operations across rack-scale compute and storage infrastructure.

FIGURE 3. Aruba 8325-32C 32-port switch

The compute modules used with the following configurations:

- 3x HPE ProLiant DL385 Gen11 server was used with the below configuration in the test environment:
  - Processors: 2 x AMD EPYC 9354 3.25GHz 32-core 280W
  - Memory: 512 GB RAM (8 x 64GB DIMMs)
  - Storage: 3.2TB (2 x HPE 1.6TB NVMe Gen4 High Performance Mixed Use SFF).

Software

The software components and their versions used in building this solution are as listed:

- Red Hat Enterprise Linux 8.7
- TigerGraph 3.9.2

TigerGraph

TigerGraph is the world's fastest graph analytics platform designed to unleash the power of interconnected data for deeper insights and better outcomes. TigerGraph fulfills the true promise and benefits of the graph platform by tackling the toughest data challenges in real time, no matter how large or complex the dataset.

TigerGraph is the only scalable graph database for the enterprise. TigerGraph architecture allows siloed data sets to be connected for deeper and wider analysis at scale. Additionally, TigerGraph supports real-time in-place updates for operational analytics use cases. These capabilities enable applications such as AI/ML, entity resolution, fraud detection, customer 360, hyper-personalized recommendations, and more: TigerGraph is focused on accelerating advanced analytics, AI, and machine learning projects with graph algorithms.

TigerGraph is a third-generation distributed graph database with six key facets:

- Deep Link Analysis & Real-time performance
- Advanced Analytics & Machine Learning
- Transactional (Mutable) Graph
- Ease of Development & Deployment
- Scalability for massive Datasets
- Enterprise Grade security

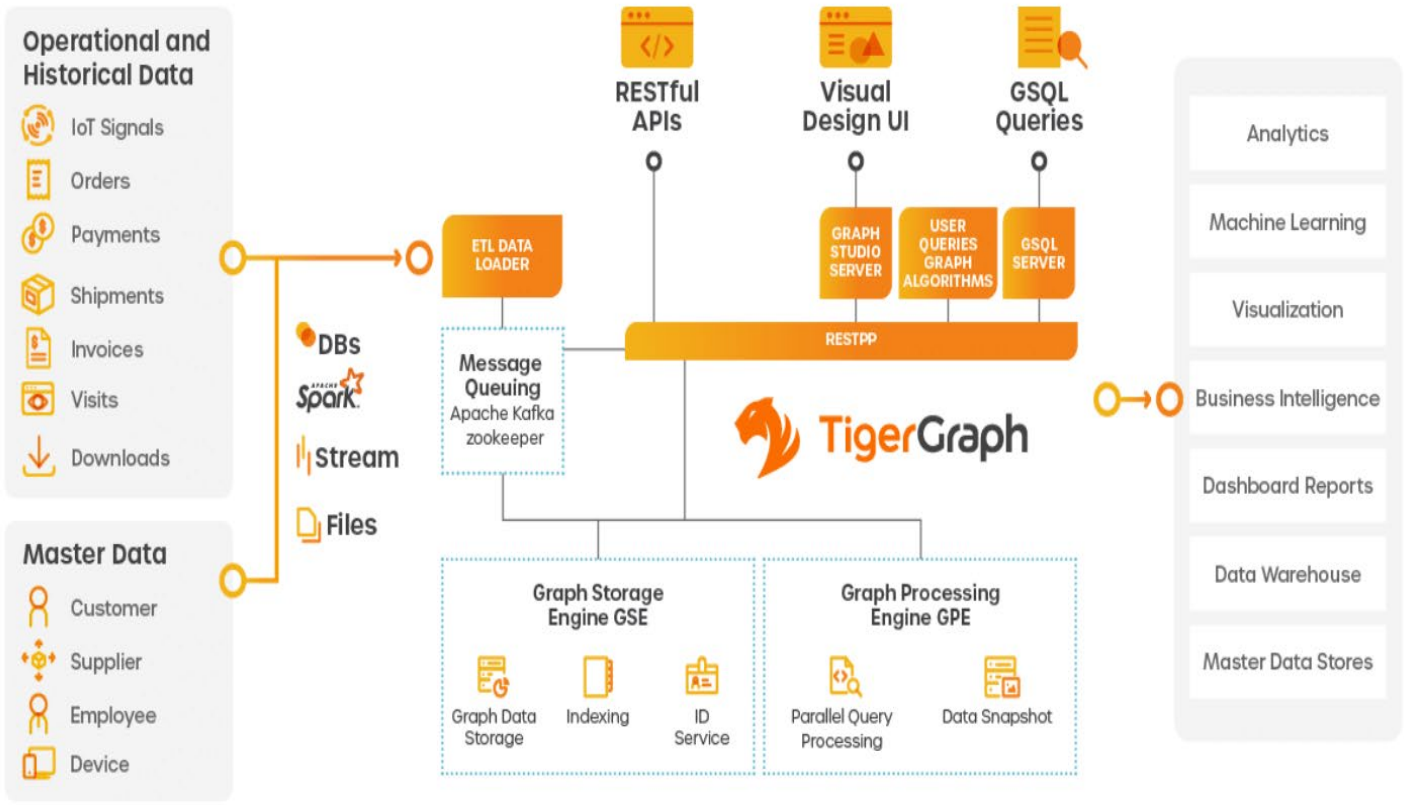

FIGURE 4. TigerGraph Architecture eco-system

TigerGraph benefits

- Maximum Price Performance - TigerGraph uses MPP architecture to uniformly utilize resources based on workload needs to deliver ROI.
- No limits on scalability - TigerGraph is designed to scale Petabytes. Customers can future proof their investment even if the use case needs to grow exponentially.
- Richer data context for applications - with TigerGraph, users can perform wider and deeper analysis of data and uncover hard-to-find patterns.

- Flexibility to run AI workloads - TigerGraph provides extensive in-database machine learning capabilities. If the users prefer, TigerGraph can help with feature extraction and expedite ML training in other downstream systems.[1]

BEST PRACTICES AND CONFIGURATION GUIDANCE FOR THE SOLUTION

Deploying TigerGraph on-prem cluster

For more information on TigerGraph Prerequisites for a cluster deployment are listed in Appendix B. For detailed, hardware and software requirement for TigerGraph on-premises, reference link https://docs.tigergraph.com/tigergraph-server/current/installation/hw-and-sw-requirements.

The loading performance has been tuned by changing the line batch size, number of concurrent requests and number of loading handler. The tuning parameter has been listed in Appendix C.

SUPPLY CHAIN USE CASE

Supply chain management delivers higher business value and competitive advantage when it is extended to provide an integrated view of the entire value chain. Graph analytics is uniquely suited to being the foundation for a new breed of solutions which do just that. Although many businesses rely on supply chains, managing them to their full potential can be a challenge. Supply chain management draws on a wide range of disciplines including operations management, logistics, procurement, and information technology, each of which is a specialized function.

Success in supply chain management comes from the integration of these disciplines – the tighter the integration, the smoother the operation. When it works, it creates net value for the business and builds a competitive infrastructure by leveraging worldwide logistics, synchronizing supply and demand, and enabling the organization to measure performance globally and locally.

The building blocks of supply chain management are strategic planning, demand planning, supply planning, procurement, manufacturing, warehousing, order fulfilment and transportation. Each of these are a professional discipline and taken together they represent a huge challenge to business. However, as we have seen, graph databases are designed to cope with the vast amounts of data involved in integrating these areas of the business.

Graph has some key data science capabilities which, as we will see, can be leveraged in supply chain management.

- Deep link analysis
- Multi-dimensional Entity and Pattern Matching
- Relational Commonality Discovery and Computation
- Hub and Community Detection
- Geospatial Graph Analysis
- Temporal (Time Series) Graph Analysis
- Machine Learning Feature Generation and Explainable AI

Supply Chain Optimization Workload description and implementation

In this supply chain use case scenario, we have data related to wheel set of cars and trucks, where the wheel sets have many inventories required for the assembly line like ABS modules, Aluminum rods, rubber for tires, values, sensors and many more. Each of these inventories are produced at different sites and produced and delivered by various vendors at their respective sites. These parts are assembled at various sites to produce the car wheel sets and truck wheel sets.

We have created a schema of four types of vertices and five types of edges. Some of the edges are directed edges and some of them are undirected edges. In the below example you can see how we can create the schema in TigerGraph using GSQL.

```
# Create schema
CREATE GRAPH demo_graph ()
```

```
CREATE SCHEMA_CHANGE JOB change_schema_of_demo_graph FOR GRAPH demo_graph {
        ADD VERTEX product(PRIMARY_ID pid STRING, name STRING, price FLOAT, formula STRING) WITH
STATS="OUTDEGREE_BY_EDGETYPE", PRIMARY_ID_AS_ATTRIBUTE="false";
        ADD VERTEX site(PRIMARY_ID sid STRING, name STRING) WITH STATS="OUTDEGREE_BY_EDGETYPE",
PRIMARY_ID_AS_ATTRIBUTE="false";
        ADD VERTEX p_order(PRIMARY_ID orderId STRING) WITH STATS="OUTDEGREE_BY_EDGETYPE",
PRIMARY_ID_AS_ATTRIBUTE="false";
        ADD VERTEX stocking(PRIMARY_ID stockingId STRING) WITH STATS="OUTDEGREE_BY_EDGETYPE",
PRIMARY_ID_AS_ATTRIBUTE="false";
        ADD DIRECTED EDGE usedBy(FROM product, TO product, formula_order STRING, useAmount FLOAT) WITH
REVERSE_EDGE="reverseUsedBy";
        ADD DIRECTED EDGE deliver(FROM site, TO site, itemId STRING) WITH REVERSE_EDGE="reverseDeliver";
        ADD DIRECTED EDGE produce(FROM site, TO product) WITH REVERSE_EDGE="reverseProduce";
        ADD UNDIRECTED EDGE prodOrder(FROM p_order, TO product, amount INT);
        ADD UNDIRECTED EDGE prodStocking(FROM stocking, TO product, amount INT);
}
RUN SCHEMA_CHANGE JOB change_schema_of_demo_graph
DROP JOB change_schema_of_demo_graph
```

Created schema that can be viewed in the Graphstudio.

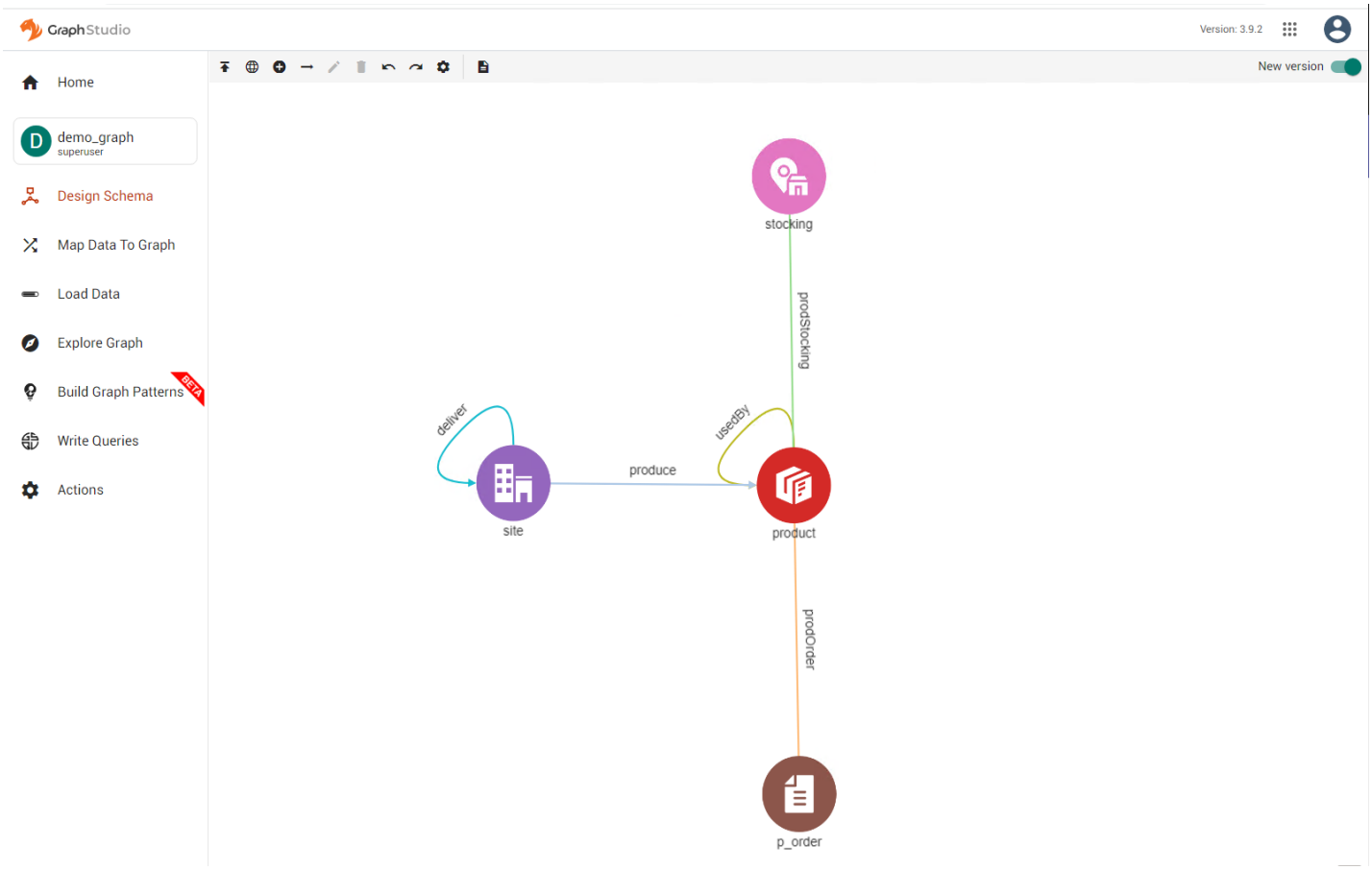

FIGURE 5. Created Schema in GraphStudio

After a graph schema is created, the system is ready to load data into the graph store. GSQL language offers data loading which performs many of the same data conversion, mapping, filtering, and merging operations that are found in enterprise ETL (Extract, Transform, and Load) systems.

Users can create loading jobs to import data into the graph. The sources on the data can be from local drive or Amazon S3 or Google Cloud Storage or Azure Blob storage. In the below case, we have imported the local csv files into the graph store.

```
# loading job to import the CSV data and load into the Graph

CREATE LOADING JOB load_job_usedBy_csv_1561418162526 FOR GRAPH demo_graph {
      DEFINE FILENAME MyDataSource;
      LOAD MyDataSource TO EDGE usedBy VALUES($1, $0, $2, $3) USING SEPARATOR=",", HEADER="true", EOL="\n";
    }
```

After executing the loading job and data is loaded, we have 36 vertices and 98 edges in this case study. We can write various queries as per the customer requirement for analysis using TigerGraph GSQL. The below query will show the impact of various sites and parts that could likely be impacted in the assembly line if a site A or any, is affected by any shortage of source material/parts or any other external factors.

```
# Query to check the Impact of any site is effected

create query impact_analysis(set<vertex<site>> affectedSites, uint maxDepth) for graph demo_graph syntax v2
{
   typedef tuple<string siteName, string  prodName> spPair;

   SetAccum<spPair> @pair_Set;

   // products a site that has been affected
   SetAccum<string> @affected_Prod;

   // number of iteration
   SumAccum<int> @@iteration = 0;

   // storing the result for unfinished goods
   SetAccum<edge> @@result_Set;

   // for pass along the edge info for visualization
   SetAccum<edge> @edge_Msg_Set;

   effectedSite = {affectedSites};

   while (effectedSite.size() > 0) limit maxDepth do

     // get the impacted products from the impacted sites
     effectedProduct = select p
                       from effectedSite:s -(produce>:e)- product:p
                       where @@iteration == 0 or s.@affected_Prod.contains(p.name) == true
                       accum p.@pair_Set += spPair(s.name, p.name)
                             // we assume all of the products got impacted from the input sites
                             ,case when @@iteration == 0 then
                                 s.@affected_Prod += p.name
                                 ,@@result_Set += e
                             end
     ;

     // from the impacted products get its downsteam products
     downStreamProd = select p
                      from effectedProduct:ep -(usedBy>:e) - product:p
                      accum p.@pair_Set += ep.@pair_Set
                            ,p.@edge_Msg_Set += e
     ;

     // find the impacted sites from the previous effectedSite set
     effectedSite = select st
```

```
                    from effectedSite:s -(deliver>:e) - site:st
                    accum
                      case when s.@affected_Prod.contains(e.itemId) then
                        st.@pair_Set += spPair(s.name,e.itemId)
                        ,st.@edge_Msg_Set += e
                      end
    ;

    // keep the impacted sites that have delivered impacted product, which are delivered from a upstream site,
and they also use this product in producing their product
    effectedSite = select st
                       from effectedSite:st -(produce>:e) - product:p
                       where COUNT(st.@pair_Set INTERSECT p.@pair_Set) > 0
                       accum
                           st.@affected_Prod += p.name
                           ,@@result_Set += e
                           ,@@result_Set += st.@edge_Msg_Set
                           ,@@result_Set += p.@edge_Msg_Set
                       post-accum
                           p.@pair_Set.clear()
    ;

    @@iteration += 1;
  end;

  print @@result_Set;
}

# installed v2
create query show_whole_graph () for graph demo_graph syntax v2
{
  // print all vertexes and edges
  SetAccum<edge> @@display_Set;

  Start = {ANY};

  print Start;

  Start = select s
          from Start:s-((prodOrder|prodStocking|usedBy>|produce>|deliver>):e)-:t
          accum @@display_Set += e;

  print @@display_Set;
}
```

Impact Analysis query output of a site5 for depth of 10 using the TigerGraph GraphStudio. TigerGraph can give deeper insights through queries which can traverse 10 or more hops and perform complex analytics.

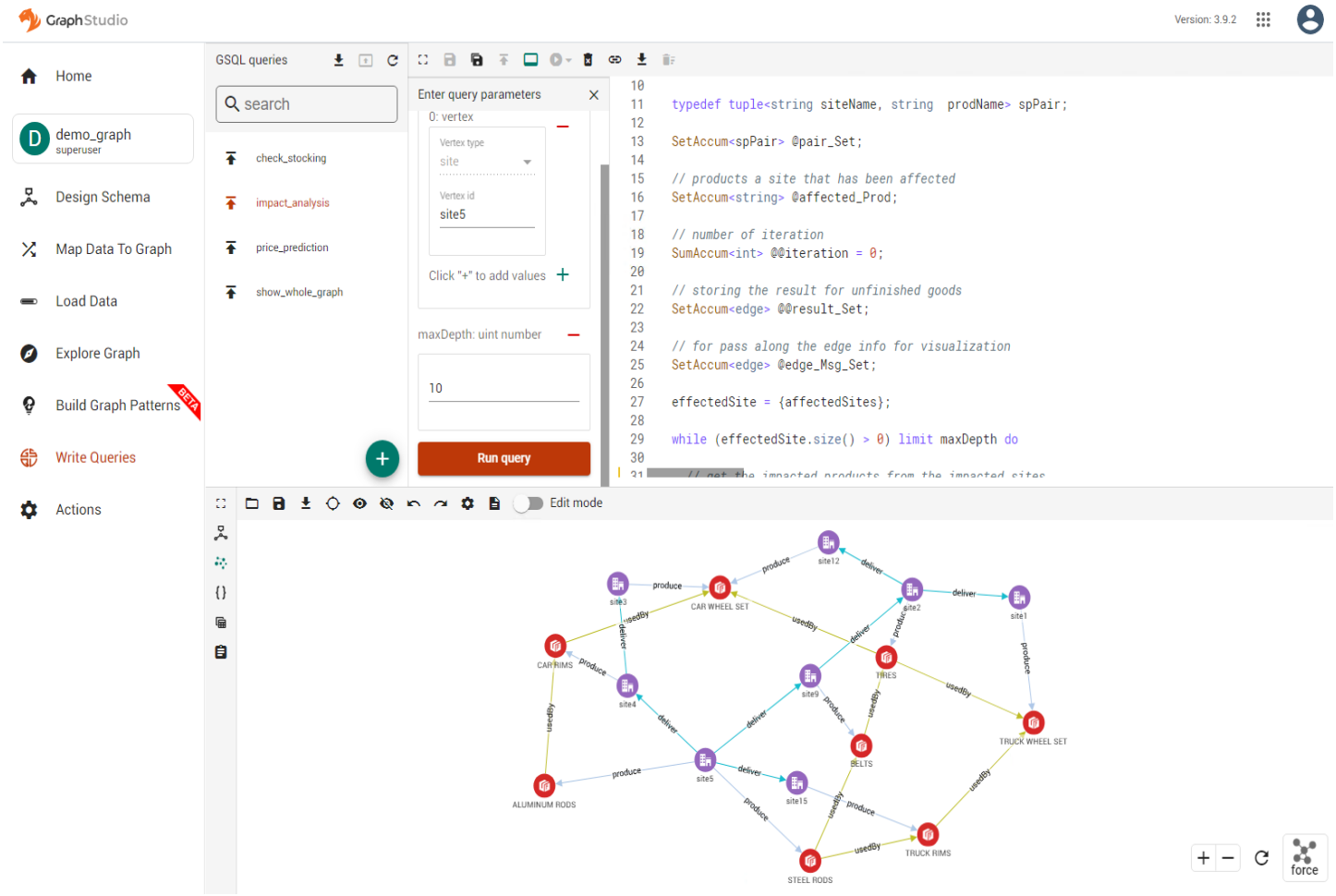

FIGURE 6. Impact Analysis from the created query

We can implement various use cases using TigerGraph, for example Customer 360, Cybersecurity, Enterprise Knowledge graph, Fraud detection, Social Network analysis and many more. TigerGraph offer various starter kits: https://www.tigergraph.com/starterkits/

SUMMARY

Hewlett Packard Enterprise with ProLiant DL385 Gen11 powered by AMD EPYC and TigerGraph can address the graph-related challenges as they work to unlock real value from connected data. HPE ProLiant DL385 and TigerGraph can accommodate large volumes of data in large clusters, to deploy completely distributed, parallel graph computing platforms. The cluster platform can query through tens of millions of edges and vertices with improved response time which will help customers in faster reporting in ever-higher transaction volumes that can flexibly grow with the business needs. Each HPE ProLiant Gen 11 server is powered by AMD EPYC CPUs featuring exceptional bandwidth with support for high-speed DDR5 memory, 12 memory channels per socket, 128 PCIe lanes for serving I/O devices like storage, network NICs and accelerators.

TigerGraph has addressed these pain points with the world's fastest and most scalable graph platform, providing massive scalability of data volumes, fast deep-link analysis for real-time performance. TigerGraph involves leveraging the power of graph-based data modeling and analytics to enhance efficiency, reduce costs and timely deliveries.

Some of the benefits of using TigerGraph for Supply Chain use cases:

- Data-driven Decision making
- Real-time Monitoring

- Reduced lead times
- Improved Customer satisfaction
- Cost reduction and Competitive advantage

IMPLEMENTING A PROOF-OF-CONCEPT

As a matter of best practice for all deployments, Hewlett Packard Enterprise recommends implementing a proof-of-concept using a test environment that matches as closely as possible the planned production environment. In this way, appropriate performance and scalability characterizations can be obtained. For help with a proof-of-concept, contact a Hewlett Packard Enterprise Services representative (hpe.com/us/en/services/consulting.html) or your Hewlett Packard Enterprise partner.

APPENDIX A: BILL OF MATERIALS

The following BOMs contain electronic license to use (E-LTU) parts. Electronic software license delivery is now available in most countries. Hewlett Packard Enterprise recommends purchasing electronic products over physical products (when available) for faster delivery and for the convenience of not tracking and managing confidential paper licenses. For more information, please contact your reseller or a Hewlett Packard Enterprise representative.

NOTE

Part numbers are at time of publication/testing and subject to change. The bill of materials does not include complete support options or other rack and power requirements. If you have questions regarding ordering, please consult with your Hewlett Packard Enterprise Reseller or Hewlett Packard Enterprise Sales Representative for more details. hpe.com/us/en/services/consulting.html

TABLE A1. Bill of materials

Part number	Quantity	Description
Compute		
P53921-B21	3	HPE ProLiant DL385 Gen11 8SFF Configure-to-order Server
P53701-B21	6	AMD EPYC 9354 3.25GHz 32-core 280W Processor for HPE
P50312-B21	24	HPE 64GB (1x64GB) Dual Rank x4 DDR5-4800 CAS-40-39-39 EC8 Registered Smart Memory Kit
P55083-B21	3	HPE ProLiant DL385 Gen11 8SFF Tri-Mode U.3 x4 BC Backplane Kit
P40510-B21	6	HPE 960GB SAS 12G Mixed Use SFF BC Value SAS Multi Vendor SSD
P50227-B21	6	HPE 1.6TB NVMe Gen4 High Performance Mixed Use SFF BC U.3 PM1735a SSD
P55097-B21	3	HPE ProLiant DL385 Gen11 x16 2U Secondary Riser Kit
P25960-B21	3	Mellanox MCX623106AS-CDAT Ethernet 100Gb 2-port QSFP56 Adapter for HPE
P01367-B21	3	HPE 96W Smart Storage Lithium-ion Battery with 260mm Cable Kit
P57884-B21	3	HPE ProLiant DL3X5 Gen11 Smart Storage Battery 2P 96W Cable Kit
P47789-B21	3	HPE MR216i-o Gen11 x16 Lanes without Cache OCP SPDM Storage Controller
P47789-B21	6	HPE 1600W Flex Slot Platinum Hot Plug Low Halogen Power Supply Kit
P57845-B21	3	HPE ProLiant DL385 Gen11 SFF Backplane Power Cable Kit
P57862-B21	3	HPE ProLiant DL385 Gen11 8SFF OROC x2 NVMe Box 3 Cable Kit
P57886-B21	3	HPE ProLiant DL385 Gen11 2U Standard/Performance FIO Air Baffle Kit
P58465-B21	18	HPE ProLiant DL3X5 Gen11 2U Performance Fan Kit
P50400-B21	3	HPE Gen11 2U Bezel Kit
P52351-B21	3	HPE DL3XX Gen11 Easy Install Rail 2 Kit
P58459-B21	6	HPE ProLiant DL3X5 Gen11 Performance 2U Heat Sink Kit
Network: Aruba 8325 and Aruba 6300		
R9F67A	2	Aruba 8325-32C Power to Port Airflow 6 Fans 2 Power Supply Units Bundle for HPE
R9G27AAE	2	Aruba Fabric Composer Device Management Service Tier 4 Switch 3 year Subscription E-STU for HPE
R9F59A	2	Aruba 4-post Rack Kit for HPE
R9F77A	2	Aruba 100G QSFP28 to QSFP28 1m Direct Attach Copper Cable for HPE
R9F63A	2	Aruba 6300M 48G Power to Port Airflow 2 Fans 1 Power Supply Unit Bundle for HPE
R9F61A	2	Aruba 6300M 12VDC 250W 100-240VAC Power to Port Airflow Power Supply Unit for HPE
R9F57A	2	Aruba 1U Universal 4-post Rack Mount Kit for HPE
R9G32AAE	2	Aruba Fabric Composer Device Management Service Tier 3 Switch 3 year Subscription E-STU for HPE
R9G06A	2	Aruba 50G SFP56 to SFP56 0.65m Direct Attach Copper Cable for HPE

APPENDIX B: TIGERGRAPH PRE-INSTALLATION

```
#On all node of cluster, create an OS user and password.
sudo useradd NEW-USER
sudo passwd NEW-USER

#Add the new user to the AllowUsers row in the /etc/ssh/sshd_config file.
AllowUsers OLD-USER NEW-USER

#restart sshd.
/sbin/service sshd restart
```

APPENDIX C: TIGERGRAPH CONFIGURATION PARAMETERS

```
gadmin config group timeout
# Change FileLoader.Factory.DefaultQueryTimeoutSec: 16 -> 6000
# Change KafkaLoader.Factory.DefaultQueryTimeoutSec: 16 -> 6000
# Change RESTPP.Factory.DefaultQueryTimeoutSec: 16 -> 6000

gadmin config entry GPE.BasicConfig.Env
# add MVExtraCopy=0; (default is 1)
# [optional] add ConcurrentRequest=[value]; (default value is 16)

gadmin config group RESTPP-LOADER
# [optional] change FileLoader.Factory.HandlerCount

# [optional]To imports any UDF's defined in the schema
gadmin config set GSQL.UDF.EnablePutTgExpr true
gadmin config set GSQL.UDF.EnablePutExpr true
gadmin config set GSQL.UDF.EnablePutTokenBank true
gadmin config set GSQL.UDF.Policy.Enable false

gadmin config apply -y
gadmin restart all -y
```

RESOURCES AND ADDITIONAL LINKS

HPE Reference Architectures, hpe.com/info/ra

HPE Servers, hpe.com/servers

HPE Storage, hpe.com/storage

HPE Networking, hpe.com/networking

HPE Ezmeral marketplace, https://www.hpe.com/us/en/software/marketplace.html

HPE GreenLake Advisory and Professional Services, https://www.hpe.com/us/en/services/consulting.html

TigerGraph, www.tigergraph.com/hpe

Tigergraph, https://www.tigergraph.com/amd/

AMD, https://www.amd.com/en/processors/epyc-server-cpu-family

To help us improve our documents, please provide feedback at hpe.com/contact/feedback.

a50009658enw

Hewlett Packard Enterprise