Multicloud Data Analytics with Dell Technologies Powered by Starburst

Enabled by Dell PowerEdge R740xd Server, ECS EX500 Storage, PowerSwitch Networking, Red Hat OpenShift Container Platform 4.10, Starburst Enterprise Platform 402.2.0

March 2023

Dell Reference Architecture for Data Virtualization with Starburst

Abstract

This reference architecture describes how to set up a Starburst Enterprise Platform (SEP) with Dell PowerEdge R740xd servers and ECS EX500 object storage to enable simple and fast access to data, wherever it resides across the enterprise.

Revisions

Date	Description
Jan 2023	Initial Release
March 2023	Single and federated queries validated on data residing in the PowerScale. Added validation for: • Queries derived[1] from TPC-DS on data also derived from TPC-DS with Scale Factor of 10000 (10 TB dataset) • Stargate with SEP Clusters running remotely on Amazon Web Services (AWS) and on premises

About Starburst

Starburst is the analytics engine for all your data. Starburst provides fast and efficient analytics engine for your data warehouse, data lake, or data mesh. Starburst unlocks the value of distributed data by making it fast and easy to access, no matter where it lives. Starburst queries data across any database, making it quickly actionable for data-driven organizations. With Starburst, teams can lower the total cost of their infrastructure and analytics investments, prevent vendor lock-in, and use the existing tools that work for their business. Trusted by companies like Apache Corporation, Comcast, DoorDash, FINRA, Marks and Spencer, and VMware, Starburst helps companies make better decisions faster on all data. For more information, go to Starburst.

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.

Table of Contents

Revisions 2

About Starburst 2

Copyright 2

1.0 Executive Summary 5

2.0 Common Business Challenges and Data Integration 5

3.0 Multicloud Data Analytics with Dell Technologies Powered by Starburst 6

4.0 Key Benefits of the Solution 7

5.0 Key Use Cases 7

6.0 Introduction to Starburst 8

6.1 How does Starburst work? 9

7.0 Reference Architecture Components 10

7.1 Hardware 10

7.1.1 PowerEdge R740xd 10

7.1.2 ECS EX500 11

7.1.3 PowerScale File Storage 12

7.2 Networking 12

7.2.1 PowerSwitch S5248F-ON 12

7.3 Software 13

7.3.1 Operating System: Red Hat Enterprise Linux CoreOS (RHCOS) v410.84.202210061459-0 (Ootpa) 13

7.3.2 Kubernetes Platform: Red Hat OpenShift Container Platform v4.10.37 13

7.3.3 Analytics Engine: Starburst Enterprise Platform v402.2.0 13

7.3.4 Metastore: Hive Metastore v3.1.2-e.21 13

7.3.5 Object-Relational Database: PostgreSQL v14.15 13

8.0 Solution Overview 14

8.1 Hardware and Software Architecture 14

8.2 Installation 16

9.0 Test Setup 17

9.1 Test Data and Queries 17

9.2 Test Data Generation 17

9.3 Test Methodology 18

9.3.1 Tests Executed 18

9.3.2 Metrics Collected 18

9.3.3 Metrics Computed ... 18

9.3.4 Observations ... 18

10.0 Performance Summary ... 19

10.1 Validation Environment 1 ... 19

10.1.1 Results ... 19

10.1.2 Observations ... 20

10.2 Validation Environment 2 ... 21

10.2.1 Results ... 21

10.2.2 Observations ... 22

10.3 Validation Environment 3 ... 23

10.3.1 Results ... 23

10.3.2 Observations ... 24

10.4 Validation Environment 4 ... 25

10.4.1 Observations ... 25

10.5 Validation Environment 5 ... 26

10.5.1 Results ... 26

10.5.2 Observations ... 27

10.6 Validation Environment 6 ... 28

10.6.1 Observations ... 28

11.0 Conclusion ... 28

12.0 Terminology ... 29

13.0 Support ... 29

14.0 Related Resources ... 29

15.0 Dell Technologies Customer Solution Centers ... 30

16.0 Dell Technologies InfoHub ... 30

17.0 Contact Information for Additional Questions ... 30

18.0 Footnotes ... 30

A Connectors and Configuration ... 31

A.1 Iceberg ... 31

A.2 Hive ... 31

A.3 Stargate ... 32

A.4 Create a schema with views from multiple data sources and run queries derived from TPC-DS to achieve query federation ... 33

B TPC-DS Queries Generation and Required Modifications ... 36

C TPC-DS Data Generation Steps ... 37

1.0 Executive Summary

Multicloud data sprawl is a key challenging trend in many enterprises today. The variety and distributed nature of data sources poses new challenges to discover, access, and activate data. Data users often need to wait until data is consolidated into a central data warehouse/data lake before they can consume it for analytics, data science, or operational purposes.

Infrastructure teams are caught between the need to modernize and consolidate data platforms and business teams who need quick access to data to drive business value. Breaking down data siloes can be enabled in a systemic way to enable access to data no matter where it resides in the enterprise. Customers need an easy, efficient, and secure way to explore and publish their data assets spread across data sources (on-premises, colocation, cloud, and edge). Enterprises also need to move towards a multicloud by design strategy. Dealing with these situations requires acknowledging the distributed nature of data, workloads, and environments. It requires being proactive about choosing a data strategy that does not enforce constraints, rather embraces flexibility, and allows customers the choice of where and how they want to consume the data. Breaking down data siloes no longer requires constant migrations of data, but a more systemic way to enable access to data no matter where it resides in the enterprise.

Starburst, the analytics engine for data, provides a fast and efficient analytics engine for data warehouse, data lake, or data mesh. Starburst unlocks the value of distributed data by making it fast and easy to access, no matter where it lives.

Dell Technologies and Starburst together offer a Multicloud Data Analytics solution enabling customers to run their business intelligence, analytics, and data science workloads and access data spread across the enterprise using the Starburst federated query engine. The complexities of the underlying data sources and data formats are abstracted from the end user, thus greatly simplifying the consumption experience. Customers can also build data products to enable greater governance and reuse of data assets across their enterprise.

This reference architecture comprises a Dell server (PowerEdge R740xd), Starburst Enterprise Platform (SEP), and Dell object store (ECS EX500) deployed on RedHat OpenShift Kubernetes Platform.

2.0 Common Business Challenges and Data Integration

The data landscape for most customers is complex. It involves collecting high volumes of data spread across different locations and business units, often using infrastructure spanning on-prem deployments, public clouds, or edge systems. In such scenarios, data silos emerge naturally, precluding an effective utilization of the diversity of data that exists across the company.

Data integration is the term used to refer to techniques that enable processing and joining such data present across different database and storage systems. Data federation is a technique that allows multiple databases to function as one. It consists of implementing a query engine that processes data across a variety of sources. Such data sources can be a relational database, a data warehouse, or a storage system holding data using open formats. They can exist in a single location, or be remote to each other, on-premises, or in the public cloud.

Data federation provides a new alternative to popular techniques such as Extract-Transform-Load (ETL) or Extract-Load-Transform (ELT). Data federation can process data in place and does not require any data movement. It reduces the number of data paths and the complexity of the overall data landscape. Time to insight shrinks as data movement prior to executing a query is no longer necessary. Users can explore data sets and join them with little effort, avoiding the natural silos that arise with multiple organizations and locations.

Data federation usually comes accompanied by additional techniques to address the performance penalty induced by remote data sources. The ability to create materialized views and cache both raw and intermediate data enables an efficient execution of production queries and exploration queries leveraging popular data sets. The extension of data federation to include such techniques and others is very often referred to as data virtualization.

3.0 Multicloud Data Analytics with Dell Technologies Powered by Starburst

This reference architecture showcases how you can simplify and accelerate the data analytics experience for data users by enabling a single point of access for all your data. It aims to enable immediate access to data across multiple database and storage systems, independent of whether they are on premises or in the public cloud.

Our architecture combines Dell PowerEdge servers and an ECS object storage functioning as data lake storage, together with the analytics query engine from Starburst. The Starburst Enterprise Platform builds on the popular Trino open-source SQL query engine, developed originally under the name of Presto at Meta. Trino queries can process data across multiple data systems, including storage systems containing data in open file formats such as Apache Parquet and open table formats such as Apache Iceberg that adds table structure to formats such as Parquet.

With a query engine such as Trino, we implement a layer on top of data that abstracts away details on location, connectivity, language variations and even API. Such a layer of abstraction is critical to simplify data analytics over a diverse set of data sources, including storage systems such as the Dell Elastic Cloud Storage, capable of storing large amounts of data using open formats.

We built the environment described in this document using Dell Technologies infrastructure and evaluated the performance of the Starburst engine using queries derived from TPC-DS, a benchmark widely accepted by the industry. We have detailed our set up and experiments for reproducibility in this paper.

4.0 Key Benefits of the Solution

Today an enterprise data landscape comprises several data sources spread across their on-premises data centers, cloud, and edge. Most data users spend a considerable amount of time locating data, gain access to it, explore it, and consume it.

Data users can obtain different benefits from our solution:

- Data analysts, data scientists, and data engineers:
- High performance engine to access and explore data.
- In-place exploration of data without having to move data into a centralized store like data warehouse or data lake.
- Reduced dependency on data engineering or IT teams to provision data in a centralized repository.
- Simplified, single point of access for data users to connect to a variety of data sources.
- Data engineers:
- Create data products for frequently used data assets.
- Explore, sample and model data assets without creating data pipelines and moving data.
- Data Stewards, Data Governance admins, and Data Platform admins:
- Ability to define policies and ensure access control in a uniform manner.
- IT and Infrastructure teams:
- Ability to provision access quickly to data without making their users wait until data pipelines are set up.
- Ability to track data usage and use this insight to consolidate the highest value, frequently used data into higher performance/lower latency data stores like data lakes.
- Flexibility to scale and isolate workloads (BI, ad hoc, AIML, operational) for optimal performance.

5.0 Key Use Cases

This solution unlocks the following use cases for an enterprise:

- Data Lake Query engine – Fast query engine to read data from a data lake.
- Federated queries – Join data across multiple data sources and data types without having to physically move data.
- Multicloud queries – Federated queries joining data across multiple locations such as on-prem and public clouds, significantly reducing the need to move data between locations.
- Data governance – Unified access control and governance by enabling single point of access for all your data.
- Materialized Views – Create persistent copies of data integrated across systems for low latency, repeated use, for example, reporting dashboards.

- Data Products – Enable organization and reuse of data assets by publishing certified data products.
- Data Localization – Ensure compliance to regional regulations that require data to be processed locally.

6.0 Introduction to Starburst

Starburst Enterprise Platform offers a powerful SQL query engine which can be used to connect to different data sources. It promotes the fundamental paradigm shift from a single source of truth, that is, getting everything into one place, to a single point of access, that is, having access to everything from one location.

The Starburst Enterprise Platform comprises of the following key components:

- **Federated Query Engine** – Based on the Trino open-source software, which implements a distributed architecture with a coordinator and multiple workers. The coordinator accepts SQL queries from users, generates an optimized query plan for each query, and distributes the tasks of the query plan across available workers. Finally, it consolidates the results of the query execution from the workers and presents the output to the user.
- **Connectors** – Over 50 connectors to different types of data sources including relational databases, data lake, data warehouse, storage systems, cloud sources. Connectors are optimized to pushdown queries, add dynamic filtering, and deliver cost-optimal query responses.
- **Catalog** - Contains configuration and metadata to access data sources. To query a data source, configure a catalog for it and include it in the cluster. Catalog connects to a data source via a connector.
- **Virtual Views** – Create virtual, non-persistent views combining data across multiple data sources.
- **Materialized Views** – Create persistent views in your preferred storage systems for low latency production grade use cases. The views can be refreshed at regular intervals with the latest data from the sources.
- **Data Products** – Enable data producers and consumers to create, publish, discover, and manage curated datasets. This enables greater reuse of data assets across the enterprise.
- **Fine grained access control** – Column and row level permissions, role-based governance of data assets.
- **Connectivity to consumption tools** – Such as Tableau, PowerBI, Jupyter, R, Spark, and many more.
- **Hive Metastore** – Helps to map data in distributed file systems and object storage into tables and provides rich metadata.
- **Stargate** – Allows catalogs and data sources in one SEP cluster to be linked to those in another cluster, thus enabling a gateway to access data across geographies while complying with data residency requirements.

6.1 How does Starburst work?

With Starburst you can execute queries against a single system or queries that are federated across multiple systems. Consider a sample query as follows:

```
SELECT * FROM sales_order, customer_db
WHERE sales_order.c_id = customer_db.c_id
```

where `sales_order` and `customer_db` are tables in two different data sources. The Starburst query engine Coordinator uses various statistics and metadata to create an optimized query plan to execute the query. The query plan is then shared with the Worker nodes, that in turn access data sources using Connectors and execute the query. Worker nodes read data from the data sources for processing and might exercise pushdown operations when appropriate. The Coordinator finally collects the results from the Worker nodes and returns the final output.

In our architecture, we run coordinator and workers of Starburst Enterprise on Dell PowerEdge servers and use unstructured storage, such as Dell Elastic Cloud Storage (ECS), for materialized views, data products, caching, and more. We configure a Hive Metastore instance to store the metadata of the materialized views and enable them through the Iceberg connector.

7.0 Reference Architecture Components

We describe in this section those hardware components that we have used in our architecture. The configurations and models we present here, although recommended based on our experimentation, do not constitute the only possible deployment option. Dell offers a variety of configurations and models for servers, storage and switches, and different choices might be relevant depending on the use case and workload.

7.1 Hardware

7.1.1 PowerEdge R740xd

The PowerEdge R740xd delivers a perfect balance between storage scalability and performance. The 2U two-socket platform is ideal for software defined storage, service providers, or as virtual desktop infrastructure. The R740xd versatility is highlighted with the ability to mix any drive type to create the optimum configuration of NVMe, SSD, and HDD for either performance, capacity, or both. Every PowerEdge server is designed as part of a cyber resilient architecture, integrating security into the full server life cycle. For more information, refer to the Dell PowerEdge R740xd technical specification.

Note: The performance tests in this reference architecture were executed using the PowerEdge R740xd model. To benefit from the latest innovations in our PowerEdge 15G Servers, we recommend using the PowerEdge R650 or R750 server models. Our Customer Solution Centers with a team of IT experts can be leveraged to test this architecture. For more information on Customer Solution Centers, see Dell Technologies Customer Solution Centers.

In this solution, we install the Starburst Coordinator and Worker nodes on PowerEdge servers to run query workloads.

Figure 1 Dell PowerEdge R740xd Server

7.1.2 ECS EX500

The object storage platform from Dell, Elastic Cloud Storage (ECS), is the world's most cyber-secure object storage[2] that can deliver scalable public cloud services with the reliability and control of a private-cloud infrastructure. Deployable as a turnkey appliance that bundles software and Dell PowerEdge servers, or in a software-defined model, ECS empowers organizations to support enterprise workloads such as cloud-native, archive, IoT, AI, and big data analytics applications at scale. As a software-defined storage solution, ECS is designed to unlock data insights from both traditional and next-generation applications, allowing customers to deploy this platform on their own terms. It also provides comprehensive protocol support for unstructured (object and file storage).

The perfect blend of economy and density, the EX500 is a versatile option for most enterprises looking to support either modern applications or deep archive use cases. It's the ideal sandbox for in-house, cloud native, mobile, and web application storage. Rack capacity ranges from 120 TB to 7.68 PB. For more information, refer to the Dell ECS EX500 technical specification.

Note: While this reference architecture was validated using the ECS EX500 model, Dell also offers additional ECS models for different workloads, such as the high density EX5000 and NVMe based EX900 for extreme performance at scale.

In this solution, we use ECS EX500 for the following purposes:

- Data lake source where data is stored in Iceberg format and can be queried using Starburst query engine.
- Data lake destination to store Starburst materialized views, data products, caching and more.
- Object store for Starburst fault tolerant execution configuration.

Figure 2 Dell ECS EX500

7.1.3 PowerScale File Storage

PowerScale is a flexible and secure scale-out NAS solution. Regardless of the kind of data, where it lives, or how big it gets, your data lake always stays simple to manage, simple to grow, simple to protect, and simple enough to handle the most demanding workloads of today and tomorrow. New Quad-level cell (QLC) drives for PowerScale offer optimum economics for demanding NAS workloads with massive raw capacity of up to 186 PB in a 252-node cluster. For more information, refer to the Dell PowerScale Data Sheet.

Note: This reference architecture was validated with single and federated queries with the data residing at the PowerScale F200 file storage, and the tests were successful.

Figure 3 PowerScale File Storage Family

7.2 Networking

7.2.1 PowerSwitch S5248F-ON

The PowerSwitch S5248F-ON fixed switch comprise Dell's latest disaggregated hardware and software data center networking solutions, providing state-of-the-art, high-density 25/100GbE ports, and a broad range of functionality to meet the growing demands of today's data center environment.

Figure 4 Dell PowerSwitch S5248F-ON Switch

7.3 Software

7.3.1 Operating System: Red Hat Enterprise Linux CoreOS (RHCOS) v410.84.202210061459-0 (Ootpa)

Red Hat Enterprise Linux is the world's leading enterprise Linux platform, certified on hundreds of clouds and with thousands of hardware and software vendors. For more information, refer to the OpenShift Container Platform 4.10 Documentation.

7.3.2 Kubernetes Platform: Red Hat OpenShift Container Platform v4.10.37

Red Hat OpenShift Container Platform is an enterprise-ready Kubernetes container platform that automates the provisioning, management, and scaling of applications to any cloud platform or on-prem.

In this solution, we deploy Red Hat OpenShift on OS-ready PowerEdge servers. Starburst is then deployed on top of OpenShift. For more information, refer to the OpenShift Container Platform 4.10 Documentation.

7.3.3 Analytics Engine: Starburst Enterprise Platform v402.2.0

Starburst provides a single point of access to query data that lives in most of the data systems. Starburst Enterprise platform (SEP) is a fully supported, enterprise-grade distribution of Trino. For more information, refer to the Starburst Enterprise Platform documentation.

7.3.4 Metastore: Hive Metastore v3.1.2-e.21

A Metastore maps files in distributed objects stores such as Amazon Simple Storage Service (Amazon S3) and HDFS into tables and provides metadata such as columns names and type mapping. It also provides metadata to the cost-based optimizer. A metastore service is required for SEP and Starburst Stargate for numerous use cases, including the following:

- Querying object storage systems with connectors such as Hive, Iceberg, Delta Lake, and others in SEP, or with object storage catalogs in Starburst Stargate.
- Creation of views or materialized views.
- Use of data products and cache service in SEP.
- For more information, refer to the Hive Metastore Configuration documentation.

7.3.5 Object-Relational Database: PostgreSQL v14.15

PostgreSQL is a popular, open-source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

In this solution PostgreSQL is used to store Starburst components such as the Hive Metastore and Query logger.

8.0 Solution Overview

We configure a 12-node cluster of PowerEdge servers to run SEP. On each of the nodes, SEP is installed on RedHat OpenShift Container Platform. In this section, we provide details of the overall setup along with the installation process and configuration we followed.

8.1 Hardware and Software Architecture

The reference architecture cluster comprises of the following components:

- Main cluster: 12 PowerEdge R740xd server nodes:
- Three (3) Red Hat OpenShift Controller nodes:
- These nodes are also used to install one (1) Postgres primary and two (2) replicas.
- Nine (9) Red Hat OpenShift Compute nodes:
- One (1) SEP Coordinator node.
- Seven (7) SEP Worker nodes.
- One (1) node for SEP Hive Metastore and Postgres backup.
- ECS EX500 object storage: Eight (8) nodes.
- Auxiliary cluster: We set up an additional cluster uniquely to test the Stargate functionality and run federated queries across multiple SEP clusters.
- Cloud cluster: We set up an additional cluster in the US East region on Amazon Elastic Kubernetes Service (Amazon EKS) to test the Stargate functionality between on-prem and cloud, and to execute federated queries across both SEP clusters.

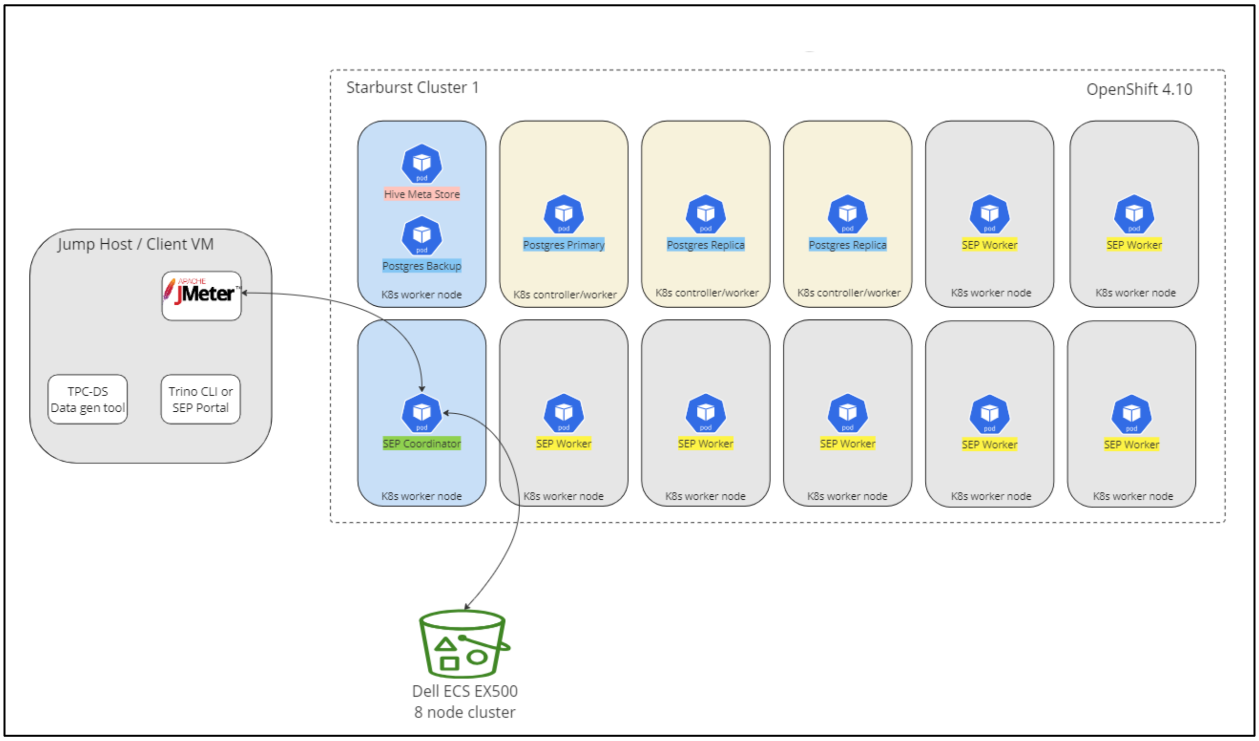


Figure 5 Main Cluster Architecture

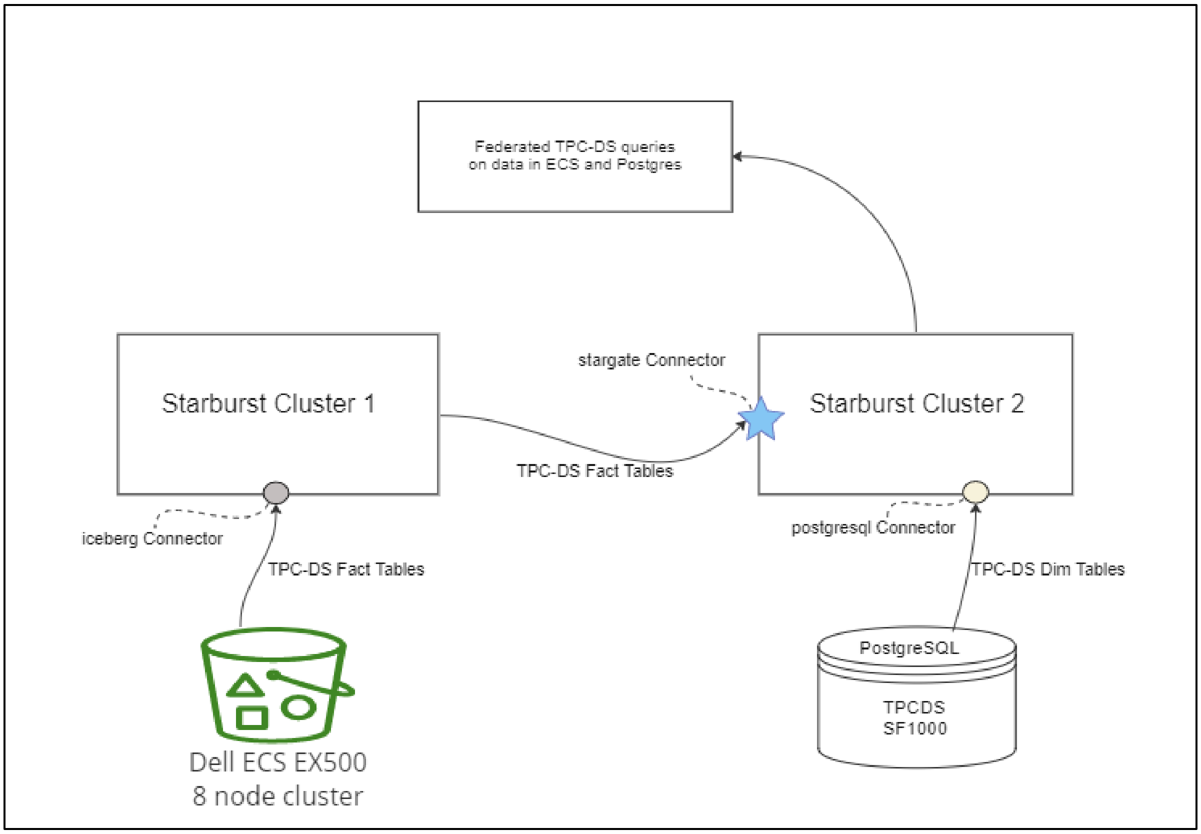


Figure 6 Multi Custer SEP Architecture Connected Using Stargate

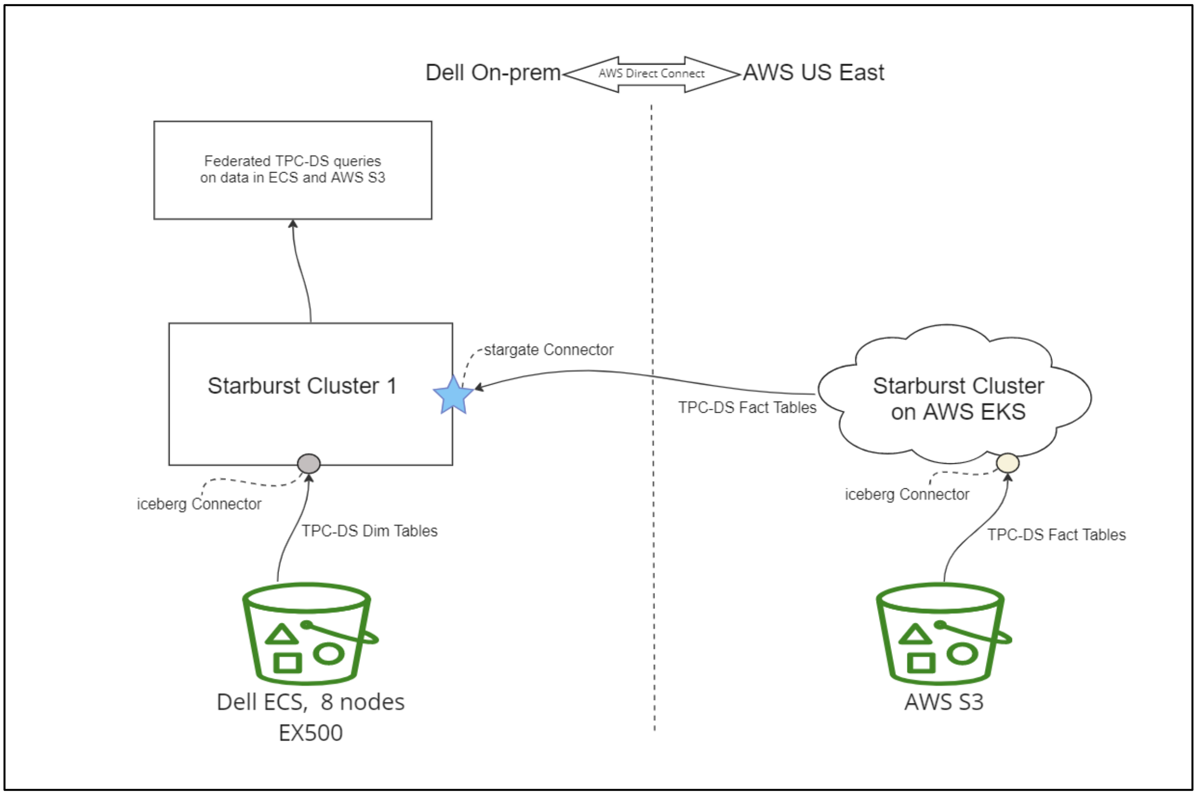


Figure 7 Multicloud SEP Architecture Connected Using Stargate

8.2 Installation

To set up SEP with Dell:

1. Set up the Dell PowerEdge R740xd server. For more information, refer to the Dell PowerEdge R740xd Installation and Service Manual.
2. Set up networking. The networking configuration and components used in this validation are shown in the illustration below.

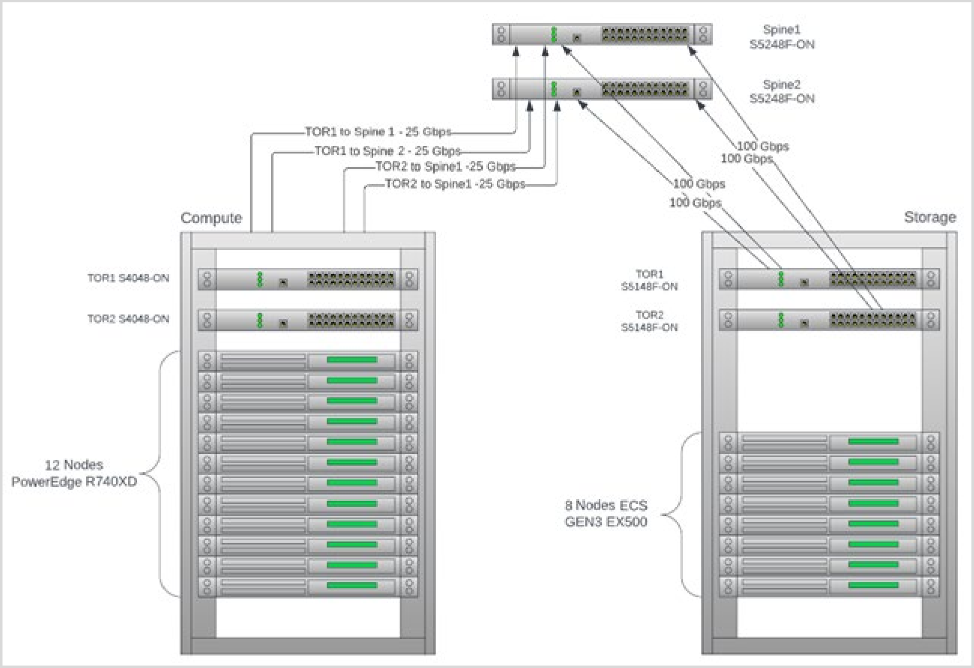


Figure 8 Network Options

Note: All the nodes use two (2) NIC cards with 2-port 25GbE on each card.

3. Set up the Dell ECS EX500. For more information, refer to Dell ECS EXSeries Hardware Guide.
4. Deploy the Red Hat Enterprise Linux CoreOS (RHCOS) on the PowerEdge 740xd Servers. For more information, refer to the Red Hat Enterprise Linux CoreOS documentation.
5. Deploy the Red Hat OpenShift Container Platform. For more information, refer to the Implementation Guide—Red Hat OpenShift Container Platform 4.10 on Intel-powered Dell Infrastructure.

   In this solution, we deploy a 3+9 node OpenShift cluster with three Controller nodes and nine Compute nodes. Out of the nine nodes, one is used to deploy Starburst Coordinator, seven are used to deploy Worker nodes and one is used to deploy the Hive Metastore and Postgres backup.
6. Deploy SEP on eight Red Hat OpenShift Compute nodes. For more information, refer to the Starburst Enterprise and Kubernetes documentation.

7. Install Postgres Primary on one of the Red Hat OpenShift Controller nodes and Postgres Replicas on the remaining 2 Controller nodes. Install Postgres backup on the ninth Compute node along with the Hive Metastore. For more information, refer to the PostgreSQL documentation.
8. Install Hive Metastore on the ninth Red Hat OpenShift Compute node. For more information, refer to the Starburst Deploy Hive Metastore with Kubernetes documentation.

9.0 Test Setup

In this section, we describe the test set up, including the queries we have used, how we generated data and a summary of the methodology we followed.

9.1 Test Data and Queries

We use queries derived from the industry-standard TPC-DS benchmark to evaluate Starburst performance. The TPC-DS benchmark data is modeled on the decision support functions of a retail product supplier and consists of seven (7) fact tables and seventeen (17) dimensions. We use the Scale Factor of 1000 and 10000 which corresponds to 1 TB and 10TB of data. The derived queries consist of the 99 queries TPC-DS specifies, for more information on how to generate the 99 queries, see Appendix B TPC-DS derived Queries Generation and Required Modifications. The 99 queries are divided into four broad classes:

- Reporting queries
- Ad-hoc queries
- Iterative OLAP queries
- Data Mining queries

9.2 Test Data Generation

The TPC-DS derived data set is generated using TPC-DS *dsdgen* utility in CSV format The data set is stored in an ECS Filesystem enabled bucket, the bucket is mounted as an NFS mount to the host where *dsdgen* utility is run during the creation of the data set. The Starburst Create Table as Select (CTAS) utility is used to convert the generated CSV data and write it to ECS in Iceberg table format and Parquet file format using the Iceberg connector. The data is generated in both partitioned and non-partitioned layout. Partitioning is done by date on all the seven (7) fact tables. Further partitioned data is compacted for higher read performance. For more information on how to generate these data, see Appendix C TPC-DS derived Data Generation Steps.

9.3 Test Methodology

The Apache JMeter test suite is used to execute benchmarking tests. Starburst Iceberg connector is used to connect to TPC-DS data in ECS.

9.3.1 Tests Executed

- Execute 3 iterations of 99 queries for each of the below mentioned test conditions.
- Execute federated queries against data sources spread across two (2) SEP clusters connected using Stargate connector.

9.3.2 Metrics Collected

- **Total execution time** = total time taken to execute derived 99 TPS-DS queries from the 3 runs
- **Per query execution time** = time taken to execute each of the derived 99 TPC-DS queries

9.3.3 Metrics Computed

- **Average execution time for 99 queries** = average of execution time to run all derived 99 queries from the 3 runs
- **Average execution time** = average execution time for each derived query from the 3 runs

9.3.4 Observations

- Variation in *Average execution time for 99 queries* and *Average execution time* per query with increasing number of concurrent users 1, 5, 10. Each concurrent thread corresponds to a User and will execute the derived 99 TPC-DS queries in its entirety and completely independently of other test threads. For example, number of concurrent users 10 will run 10 parallel sessions of 99 queries each. Multiple threads are used to simulate concurrent users to Starburst. Apache JMeter Thread groups are used to achieve these concurrent users.
- Variation in *Average execution time for 99 queries* and *Average execution time* per query when number of worker nodes is changed from 3 to 8, while keeping the per node hardware configuration the same.
- Variation in *Average execution time for 99 queries* and *Average execution time* per query when per node hardware configuration is changed from 128 GB/ 32 vCPUs from 256 GB/ 64 vCPUs, while keeping the number of worker nodes the same.

10.0 Performance Summary

The following tests were executed using Starburst against ECS as a data source. All tests were executed directly against the data source without being materialized.

10.1 Validation Environment 1

The first validation environment consisted of a small size cluster with low memory/ vCPUs node configuration:

- SEP Cluster: One (1) Coordinator plus three (3) worker PowerEdge R740xd nodes; each node having 128 GB memory and 32 vCPUs.
- Data stored in ECS EX500.
- SEP configuration parameter `query.max-memory-per-node` set to 60 GB. For more information, refer to [Starburst Enterprise 402-e.2 LTS reference documentation - Resource management properties]().

10.1.1 Results

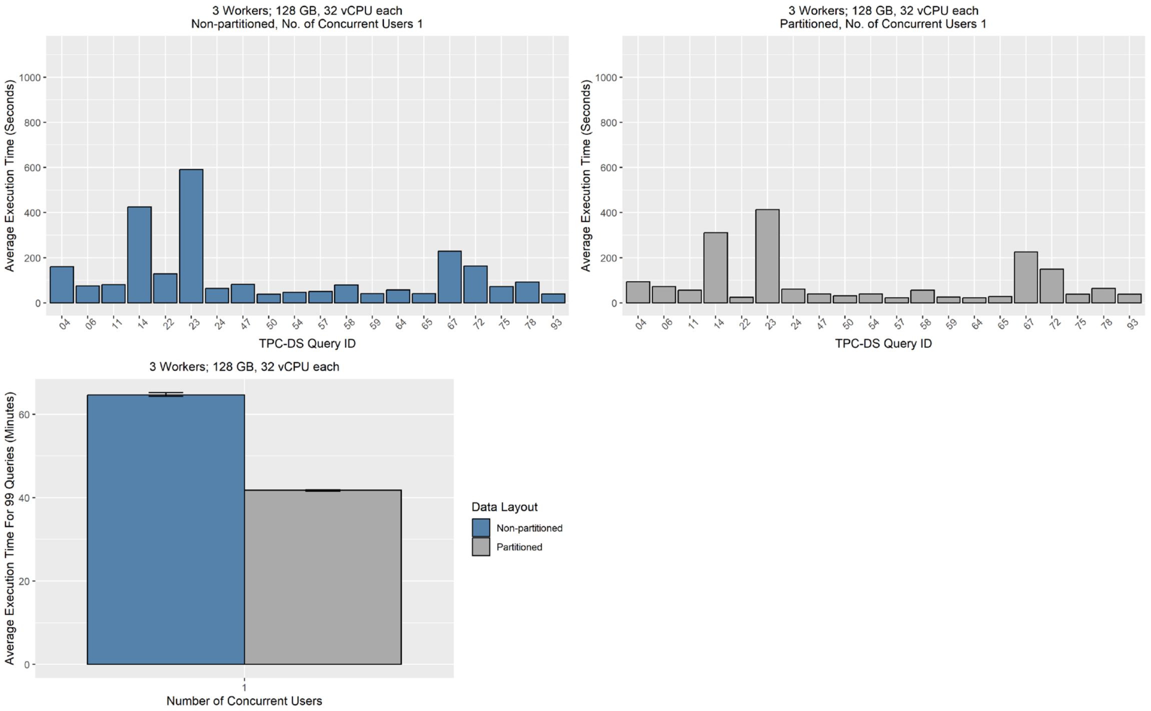


Figure 9 Validation Environment 1 Results

10.1.2 Observations

The bar charts above display the 20 queries that consumed the most time, arranged in order by their respective query IDs.

- Three (3) iterations of the derived 99 TPC-DS queries executed successfully for number of concurrent users 1.
- *Average execution time for 99 queries* for non-partitioned data is roughly 1.6 times that of partitioned data.
- The variations in the *Average execution time* per query across the derived 99 TPC-DS queries is due to the difference in the complexity of the queries and the volume of data being analyzed.

10.2 Validation Environment 2

The second validation environment consisted of a medium size cluster with low memory/ vCPUs node configuration:

- SEP Cluster: One (1) Coordinator plus seven (7) worker PowerEdge R740xd nodes; each node having 128 GB memory and 32 vCPUs.
- Data stored in ECS EX500.
- SEP configuration parameter `query.max-memory-per-node` set to 60 GB. This parameter is set to ensure that none of the worker nodes run out of memory while executing the queries. For more information, refer to Starburst Enterprise 402-e.2 LTS reference documentation - Resource management properties.

10.2.1 Results

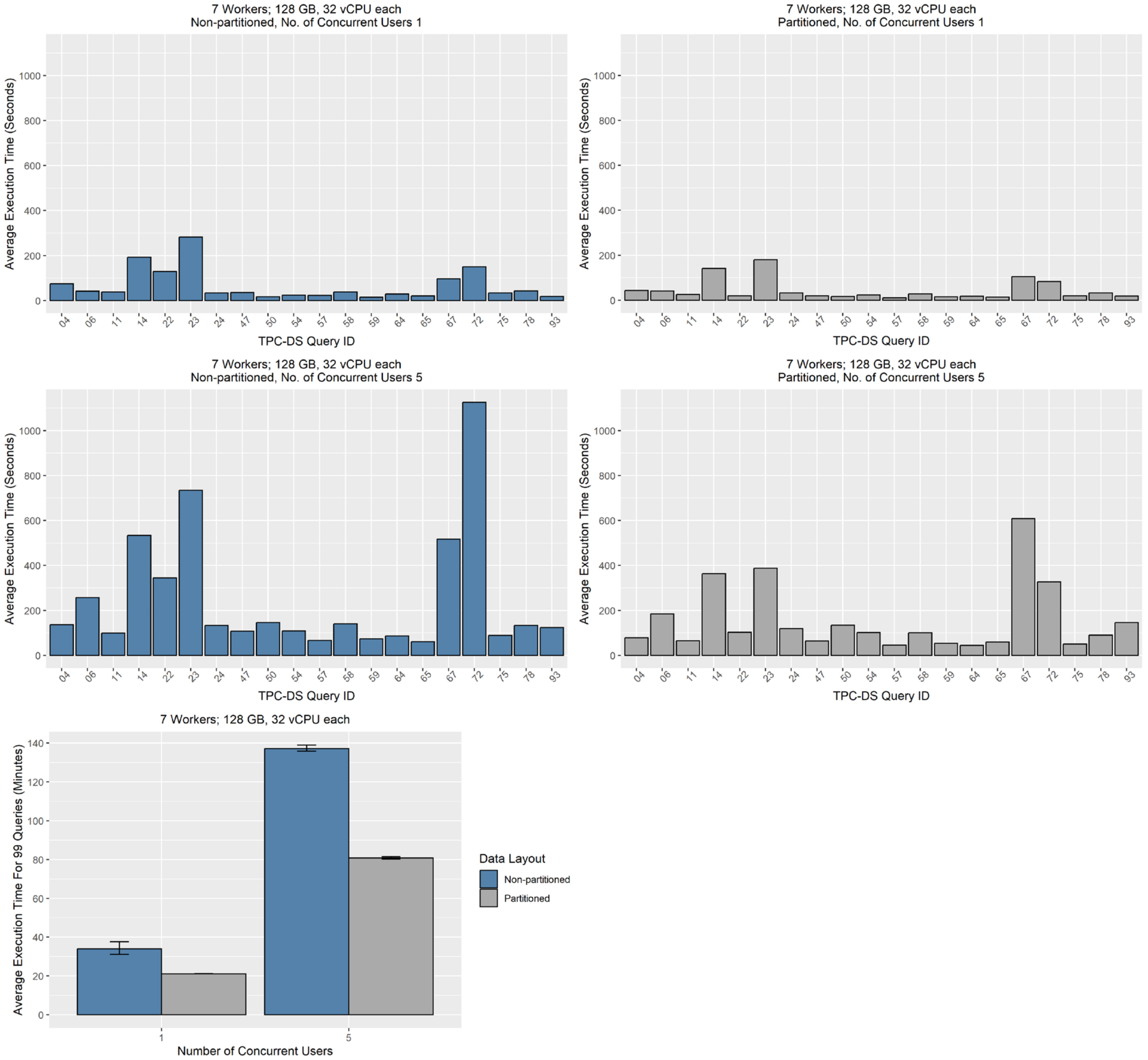


Figure 10 Validation Environment 2 Results

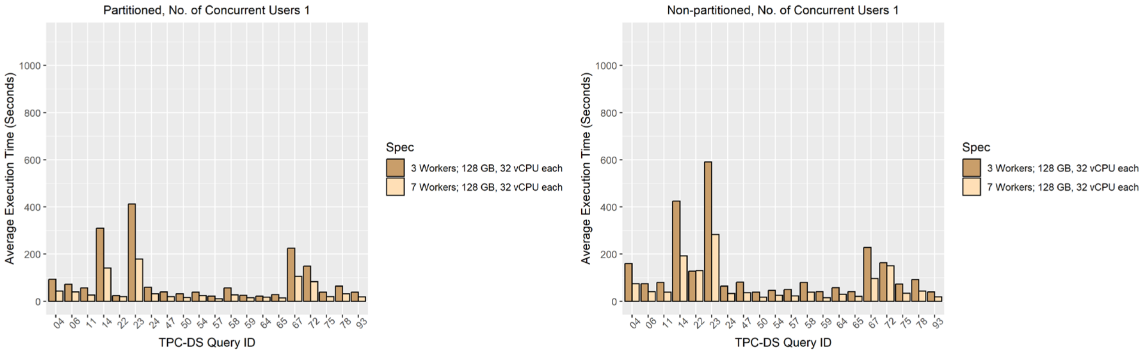


Figure 11 3 Workers vs 7 Workers 128 GB/ 32vCPUs

10.2.2 Observations

The bar charts above display the 20 queries that consumed the most time, arranged in order by their respective query IDs.

- Three (3) iterations of the derived 99 TPC-DS queries executed successfully for 1 and 5 concurrent users.
- *Average execution time for 99 queries* with non-partitioned data is 1.75 times that of partitioned data for both 1 and 5 concurrent users.
- With 10 concurrent users, some of the queries failed due to lack of sufficient memory. This indicates that as concurrent workload increases, we need to increase the hardware capacity – either by increasing the number of worker nodes or by having larger size of memory/ vCPUs per worker node.
- The variations in the *Average execution time* per query across the derived 99 TPC-DS queries is due to the difference in the complexity of the queries and the volume of data being analyzed.
- Compared to the smaller cluster size of three (3) worker nodes, the query response times improved with the bigger cluster size of seven (7) worker nodes.

10.3 Validation Environment 3

The third validation environment consisted of a medium size cluster with medium memory/vCPUs node configuration:

- SEP Cluster: One (1) Coordinator plus seven (7) worker PowerEdge R740xd nodes; each node having 256 GB memory and 64 vCPUs.
- Data stored in ECS EX500.

10.3.1 Results

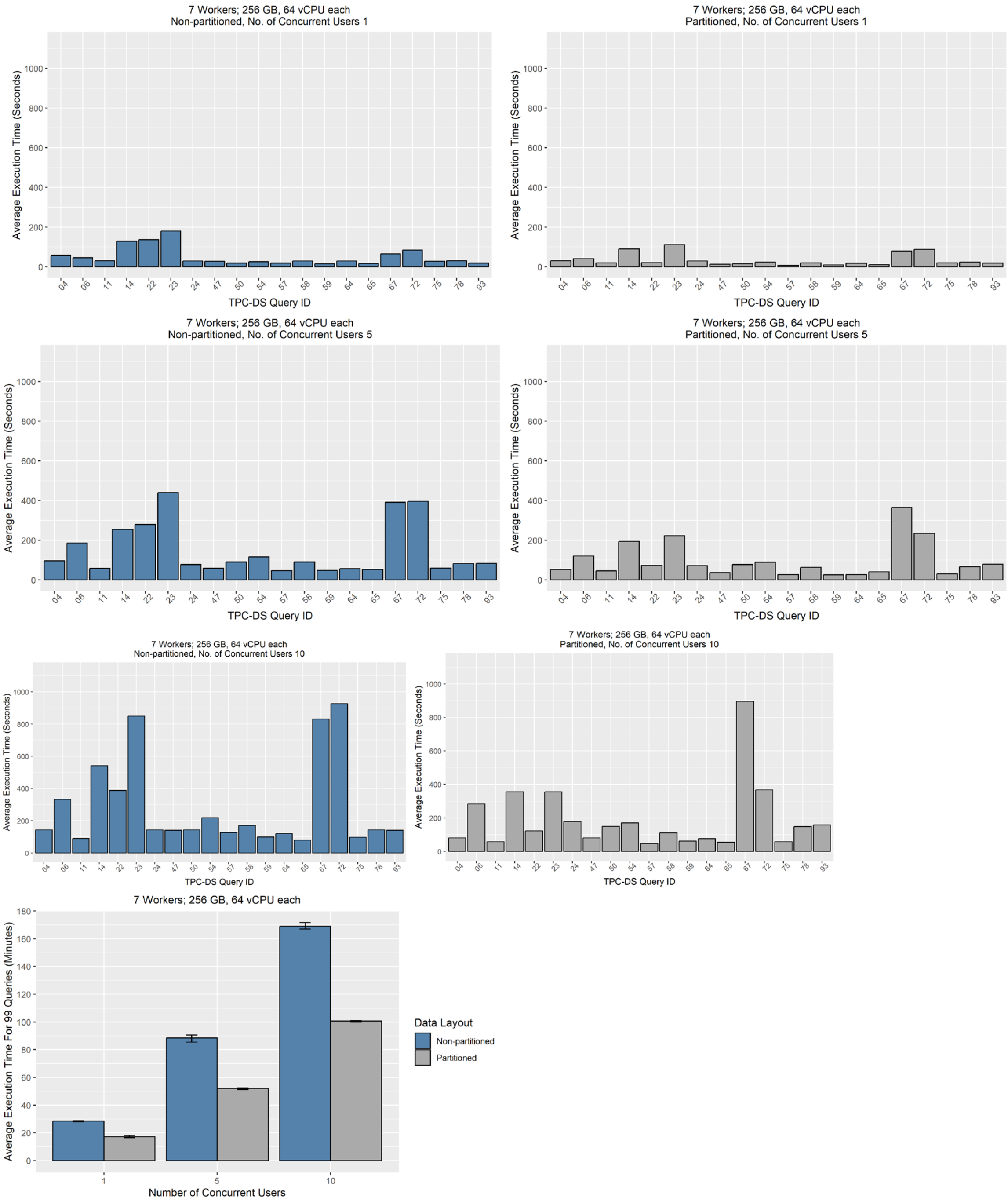


Figure 12 Validation Environment 3 Results

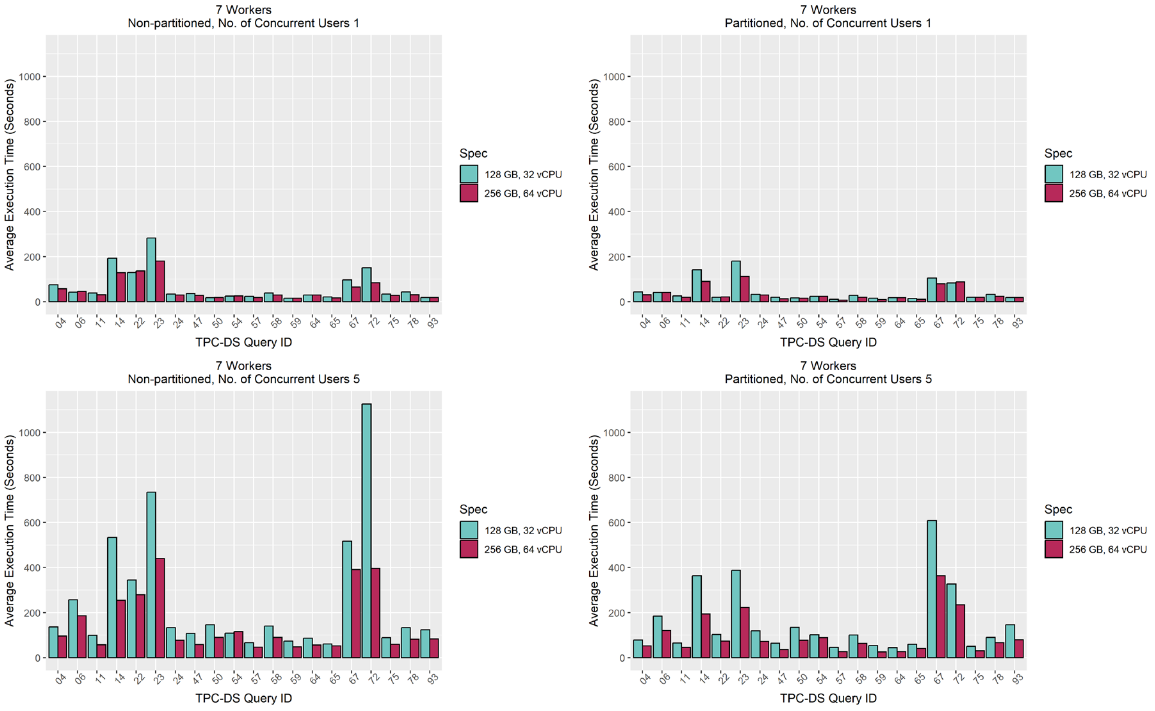


Figure 13 128 GB/ 32 vCPUs versus 256 GB/ 64 vCPUs

10.3.2 Observations

The bar charts above display the 20 queries that consumed the most time, arranged in order by their respective query IDs.

- Three (3) iterations of the derived 99 TPC-DS queries executed successfully for 1, 5 and 10 concurrent users.
- *Average execution time for 99 queries* with non-partitioned data are over 1.75 times that of partitioned data for 1, 5, and 10 concurrent users.
- *Average execution time for 99 queries* indicates a linear increase as the number of concurrent users querying the system.
- The variations in the *Average execution time* per query across the derived 99 TPC-DS queries is due to the difference in the complexity of the queries and the volume of data being analyzed.
- Compared to the lower configuration of 128 GB/ 32 vCPUs nodes, the query response times improved with the higher configuration of 256 GB/ 64 vCPUs nodes.

10.4 Validation Environment 4

The fourth validation environment consisted of a multi cluster SEP deployment:

- Two (2) SEP clusters connected using Stargate connector.
- SEP Cluster 1 connected to data source ECS EX500 with Scale Factor 1000 derived TPC-DS data.
- SEP Cluster 2 connected to Postgres DB with Scale Factor 1000 derived TPC-DS data.

10.4.1 Observations

- Single source TPC-DS derived queries were executed from SEP Cluster 2 against the ECS data source in remote SEP Cluster 1.
- Federated TPC-DS derived queries were executed from SEP Cluster 2 against data sources in both Cluster 1 and Cluster 2.
- In both the above cases the advantage of Stargate connector is that Cluster 1 would handle the pre-aggregation and filtering and only send the results back to Cluster 2.
- To enable multicloud or multi region analytics, one could potentially set up SEP Cluster 1 on-prem and SEP Cluster 2 on a public cloud or on-premises in another region and connect them using Stargate connector.

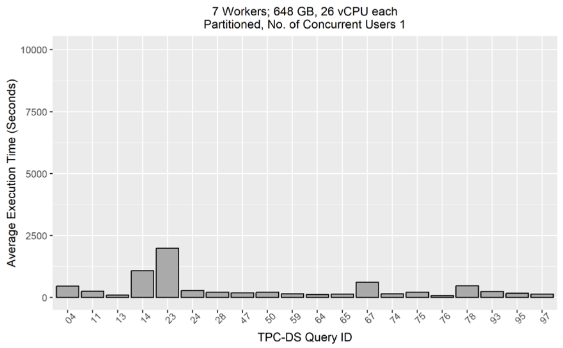
10.5 Validation Environment 5

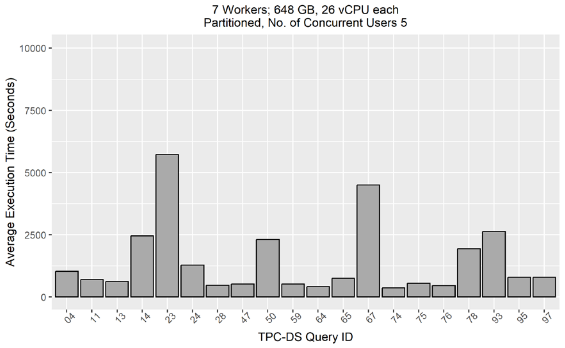
The fifth validation environment consisted of a medium size cluster with high memory/vCPUs node configuration:

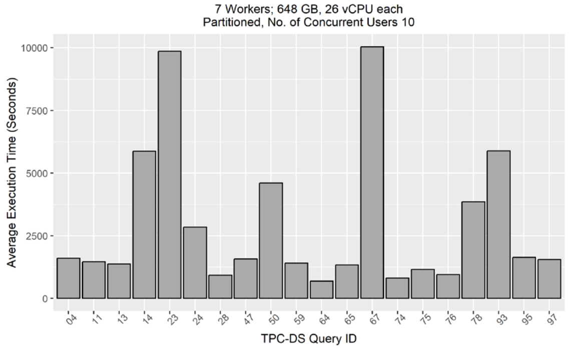
- SEP Cluster: One (1) Coordinator plus seven (7) worker PowerEdge R740xd nodes; each node having 648 GB memory and 26 vCPUs.
- Data stored in ECS EX500.
- The environment being referred to here is different from Validation Environments 1, 2, 3 and 4. In this environment, we use a derived **TPC-DS Scale Factor of 10000, which corresponds to a dataset size of 10 TB, as opposed to 1 TB in other environments**. The data is partitioned by date for all Fact tables and is also compacted. The Dimension tables are not partitioned. For more information on how to generate this data, refer to Appendix C TPC-DS derived Data Generation Steps.
- Iceberg connector configuration parameter "`iceberg.experimental.extended-statistics.enabled`" set to "`true`". This parameter is set to ensure Cost-based optimizations uses the statistics to take appropriate decisions about the query plan as indicated in the Starburst documentation for Iceberg connector. These options enable a more efficient use of resources, preventing worker nodes from running out of memory while executing queries against SF10000 data. For more information, refer to Iceberg connector configurations.

  It should be noted that this configuration parameter for the iceberg connector is experimental for SEP 402-e.2 LTS; however, in later STS releases of SEP, for example 406-e STS, this parameter is no longer experimental.

10.5.1 Results







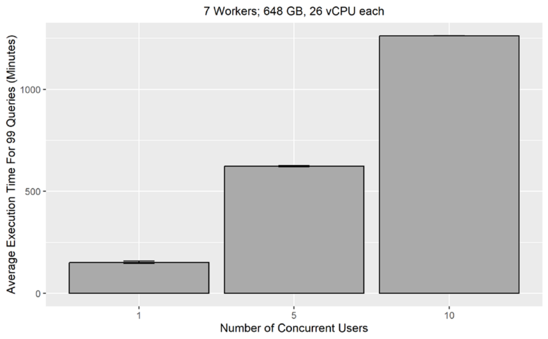


Figure 14 Validation Environment 5 Results

10.5.2 Observations

The bar charts above display the 20 queries that consumed the most time, arranged in order by their respective query IDs. The following observations were made:

- All three (3) iterations of derived 99 TPC-DS queries executed successfully for each of the three concurrency levels, namely 1, 5, and 10 concurrent users.
- **Memory:** For 10 TB data, a minimum of 648 GB per worker node is necessary to successfully execute all 99 queries over three iterations. Using less than 648 GB memory results in inconsistent success, with certain combinations of heavy queries causing insufficient cluster memory errors in some of the three iterations.
- **vCPUs:** For 10 TB data, we used 26 vCPUs per worker node, and all 99 queries executed successfully over three iterations. For queries processing high volumes data and in the presence of high concurrency, it is critical to calibrate the storage provisioning accordingly, in our case, an ECS instance. The performance of such workloads can be impacted when storage is not properly provisioned. Seek guidance from Dell on how to provision your storage system according to your workload.
- The variations in the *Average execution time* per query across the derived 99 TPC-DS queries is due to the difference in the complexity of the queries and the volume of data being analyzed.
- Overall, to improve execution times for 10 TB, we expect that increasing the capacity of ECS by adding more nodes enhances its ability to handle a greater number of concurrent requests. As a result, a larger number of ECS nodes would enable higher vCPUs on the SEP workers, leading to improved end-to-end concurrency and faster execution times.

10.6 Validation Environment 6

The sixth validation environment consisted of a multi cluster SEP deployment in a multicloud environment, in this case the AWS US East region:

- Two (2) SEP clusters connected using Stargate connector.
- SEP Cluster 1 on-premises connected to data source ECS EX500 with Scale Factor 1000 derived TPC-DS data.
- SEP Cluster 2 in the AWS US East region on Amazon EKS connected to Amazon S3 with Scale Factor 1000 derived TPC-DS data.
- Using Starburst, users can generate views that incorporate tables from multiple data sources. For additional details on this feature, refer to Create View. We created a view for each derived TPC-DS Dimension and Fact table in the Starburst Cluster 1 instance according to Figure 7. Each view maps to a select query that retrieves all the data. Views of Dimension tables point to the ECS on premises while views of Fact tables point to the Starburst instance deployed on the AWS US East region. TPC-DS derived queries (#1 and #2) were subsequently executed on this view to achieve query federation.

10.6.1 Observations

- Single source derived TPC-DS queries were executed from SEP Cluster 1 against the Amazon S3 data source in remote SEP Cluster 2 in the AWS US East region.
- Federated derived TPC-DS queries were executed from SEP Cluster 1 against data sources in both Cluster 1 and Cluster 2 in the AWS US East region.
- In both the above cases the advantage of Stargate connector is that SEP Cluster 2 in AWS would handle the pre-aggregation and filtering and only send the results back to SEP Cluster 1 thus reducing latency and egress costs.

11.0 Conclusion

Data will remain highly distributed in most enterprises. Without capabilities to access data where it resides, enterprises resort to data copies, data movement, and additional transformations that add to the complexity of the data landscape. Starburst is designed to enable quick access to data no matter where it resides, across customer-owned data centers and public cloud providers. Combined with Dell PowerEdge servers and Dell ECS object storage, it powers high performance analytics for petabyte-scale data, helping unlock valuable insights from the enterprise data assets.

Data users can accelerate data exploration without waiting for expensive ETL pipelines to be set up, while IT teams can continue to focus on simplifying and modernizing the data landscape. A critical step toward a modern data landscape is adopting data lake storage technology, such as Dell ECS, using open data formats to unlock analytics over the same data with a multitude of analytics engines, and without data copies or movement. In many organizations, different teams and user personas opt for different tools based on the specific workloads – all of them trying to access the same data. The Multicloud Data Analytics solutions empowers such users to extract business value out of their data in a simple and effective manner.

12.0 Terminology

Term	Definition
AWS	Amazon Web Services
CTAS	Create Table as Select
ECS	Elastic Cloud Storage
ELT	Extract-Load-Transform
ETL	Extract-Transform-Load
PoCs	Proof of Concepts
RHCOS	Red Hat Enterprise Linux CoreOS
SEP	Starburst Enterprise Platform

13.0 Support

If there are any issues with the Starburst components, contact Starburst support Starburst Support.

If there are issues with any of the Dell Technologies components, contact Dell support Dell.support.com.

14.0 Related Resources

The following are links to additional documentation resources:

- Starburst Enterprise 402-e.2 LTS reference documentation
- Dell PowerEdge R740xd Server
- ECS EX500 Storage
- PowerScale File Storage Family
- PowerSwitch S5248F-ON Documentation
- Red Hat OpenShift Container Platform 4.10 Documentation
- PostgreSQL 15.1 Documentation
- TPC-DS Benchmarking
- Apache JMeter
- TPC-DS Benchmarks fair use

15.0 Dell Technologies Customer Solution Centers

Our global network of dedicated Dell Technologies Customer Solution Centers are trusted environments where world class IT experts collaborate with customers and prospects to share best practices; facilitate in-depth discussions of effective business strategies using briefings, workshops, or Proof of Concepts (PoCs); and help business become more successful and competitive.

Dell Technologies Customer Solution Centers reduce the risks that are associated with new technology investments and can help improve the speed of implementation.

All services of the Customer Solution Centers are available to all Dell Technologies customers at no charge. Contact your account team today to submit an engagement request.

16.0 Dell Technologies InfoHub

The Dell Technologies InfoHub is your one-stop destination for the latest information about Dell Solutions products. New material is frequently added, so browse often to keep up to date on the expanding Dell portfolio of cutting-edge products and solutions.

17.0 Contact Information for Additional Questions

For more information, including sizing guidance, technical questions, or sales assistance, email analytics.assist@dell.com, or contact your Dell Technologies or authorized partner sales representative.

18.0 Footnotes

[1] The derived results presented herein are derived from the TPC-DS Queries Benchmark and as such are not comparable to published TPC-DS Queries Benchmark results, as the derived results presented herein do not comply with the TPC-DS Queries Benchmark Specification.

[2] Based on Dell analysis comparing cyber-security software capabilities offered for Dell ECS vs. competitive products, September 2022.

This Reference Architecture is for informational purposes only and may contain typographical error and technical inaccuracies. The content is provided AS IS, without express or implied warranties of any kind.

A Connectors and Configuration

Starburst provides access to over 50 enterprise data sources ranging from data lakes and warehouses to streaming systems, relational database systems, and more. Access and connectivity of our connectors are also bolstered with performance and security features that meet the demands of enterprise scale analytics. For more information on the list of available connectors, refer to Starburst Connectors.

A.1 Iceberg

Documentation: https://docs.starburst.io/latest/connector/iceberg.html

Example:

```
iceberg: |
    connector.name=iceberg
    iceberg.file-format=PARQUET
    iceberg.max-partitions-per-writer=6000
    hive.metastore.uri=thrift://hive:9083
    hive.metastore-timeout=2400s
    hive.s3.path-style-access=true
    hive.s3.ssl.enabled=false
    hive.s3.endpoint=http://s3-service:80
    hive.s3.aws-access-key=starburst
    hive.s3.aws-secret-key=${ENV:S3_AWS_SECRET_KEY}
```

A.2 Hive

Documentation: https://docs.starburst.io/latest/connector/hive.html

Example:

```
hive: |
    connector.name=hive
    hive.metastore.uri=thrift://hive:9083
    hive.metastore-timeout=2400s
    hive.s3.path-style-access=true
    hive.s3.ssl.enabled=false
    hive.s3.endpoint=http://s3-service:80
    hive.s3.aws-access-key=starburst
    hive.s3.aws-secret-key=${ENV:S3_AWS_SECRET_KEY}
    hive.iceberg-catalog-name=iceberg
    materialized-views.enabled=true
    materialized-views.namespace=ns
    materialized-views.storage-schema=views_cache_storage
    cache-service.uri=http://coordinator:8180
```

A.3 Stargate

Starburst Stargate enables Starburst Enterprise customers to link catalogs and data sources supported by one Starburst cluster to other catalogs and data sources in remote Starburst clusters. Stargate becomes a gateway for unlocking data access across geographies while ensuring access controls are in place, and data residency requirements are honored.

Documentation: https://docs.starburst.io/latest/connector/starburst-stargate.html

Example:

```
stargate: |
    connector.name=stargate
    connection-url=jdbc:trino://starburst-route-
default.apps.ocp.abr.sdp.hop.lab.emc.com:80/postgres
    connection-user=admin
    connection-password=pwd
    ssl.enabled=true
```

A.4 Create a schema with views from multiple data sources and run derived TPC-DS queries to achieve query federation

1. Example of a remote catalog of a data source in aws via stargate

   ```
   stargate-ambar-iceberg-aws: |
     connector.name=stargate
       connection-url=jdbc:trino://<remote fqdn or ip address of starburst in aws>/iceberg
     connection-user=dv-admin
   ```

2. Example of a local iceberg catalog of an on-prem data source.

   ```
   iceberg-local-ecs: |
     connector.name=iceberg
     iceberg.file-format=PARQUET
     iceberg.max-partitions-per-writer=101
     hive.metastore.uri=thrift://hive-moss-svc.hive-moss:9083
     hive.metastore-timeout=2400s
     hive.s3.path-style-access=true
     hive.s3.ssl.enabled=false
     hive.s3.endpoint=http://moss-service.hive-moss:80
     hive.s3.aws-access-key=<s3 access key>
     hive.s3.aws-secret-key=<s3 secret key>
   ```

3. Created a schema named stargate_view_sf1k_aws_fact_local_dim by using:

   ```
   CREATE SCHEMA iceberg.stargate_view_sf1k_aws_fact_local_dim
   WITH (location = 's3://stargate-federated/stargate_view_sf1k_aws_fact_local_dim/');
   ```

4. Create Fact Table Views : The below views point to the fact tables from remote S3 connected via the stargate:

   ```
   CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.store_sales AS
   SELECT * FROM "stargate-ambar-iceberg-aws".sf1000_partition.store_sales;

   CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.store_returns AS
   SELECT * FROM "stargate-ambar-iceberg-aws".sf1000_partition.store_returns;

   CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.catalog_sales AS
   SELECT * FROM "stargate-ambar-iceberg-aws".sf1000_partition.catalog_sales;
   ```

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.catalog_returns AS SELECT * FROM "stargate-ambar-iceberg-aws".sf1000_partition.catalog_returns;
```

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.web_sales AS SELECT * FROM "stargate-ambar-iceberg-aws".sf1000_partition.web_sales;
```

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.web_returns AS SELECT * FROM "stargate-ambar-iceberg-aws".sf1000_partition.web_returns;
```

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.inventory AS SELECT * FROM "stargate-ambar-iceberg-aws".sf1000_partition.inventory;
```

5. Create Dimension Table Views : The below views point to the dimension tables from local S3 connected via iceberg:

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.store AS SELECT * FROM "iceberg-local-ecs".ambar_local_sf1k_partitioned.store;
```

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.call_center AS SELECT * FROM "iceberg-local-ecs".ambar_local_sf1k_partitioned.call_center;
```

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.catalog_page AS SELECT * FROM "iceberg-local-ecs".ambar_local_sf1k_partitioned.catalog_page;
```

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.web_site AS SELECT * FROM "iceberg-local-ecs".ambar_local_sf1k_partitioned.web_site;
```

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.web_page AS SELECT * FROM "iceberg-local-ecs".ambar_local_sf1k_partitioned.web_page;
```

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.warehouse AS SELECT * FROM "iceberg-local-ecs".ambar_local_sf1k_partitioned.warehouse;
```

```
CREATE VIEW "iceberg-local-ecs".stargate_view_sf1k_aws_fact_local_dim.customer AS SELECT * FROM "iceberg-local-ecs".ambar_local_sf1k_partitioned.customer;
CREATE VIEW "iceberg-local-
ecs".stargate_view_sf1k_aws_fact_local_dim.customer_address AS
SELECT * FROM "iceberg-local-
ecs".ambar_local_sf1k_partitioned.customer_address;

CREATE VIEW "iceberg-local-
ecs".stargate_view_sf1k_aws_fact_local_dim.customer_demographi
cs AS SELECT * FROM "iceberg-local-
ecs".ambar_local_sf1k_partitioned.customer_demographics;

CREATE VIEW "iceberg-local-
ecs".stargate_view_sf1k_aws_fact_local_dim.date_dim AS SELECT
* FROM "iceberg-local-
ecs".ambar_local_sf1k_partitioned.date_dim;

CREATE VIEW "iceberg-local-
ecs".stargate_view_sf1k_aws_fact_local_dim.household_demograph
ics AS SELECT * FROM "iceberg-local-
ecs".ambar_local_sf1k_partitioned.household_demographics;

CREATE VIEW "iceberg-local-
ecs".stargate_view_sf1k_aws_fact_local_dim.item AS SELECT *
FROM "iceberg-local-ecs".ambar_local_sf1k_partitioned.item;

CREATE VIEW "iceberg-local-
ecs".stargate_view_sf1k_aws_fact_local_dim.income_band AS
SELECT * FROM "iceberg-local-
ecs".ambar_local_sf1k_partitioned.income_band;

CREATE VIEW "iceberg-local-
ecs".stargate_view_sf1k_aws_fact_local_dim.promotion AS SELECT
* FROM "iceberg-local-
ecs".ambar_local_sf1k_partitioned.promotion;

CREATE VIEW "iceberg-local-
ecs".stargate_view_sf1k_aws_fact_local_dim.reason AS SELECT *
FROM "iceberg-local-ecs".ambar_local_sf1k_partitioned.reason;

CREATE VIEW "iceberg-local-
ecs".stargate_view_sf1k_aws_fact_local_dim.ship_mode AS SELECT
* FROM "iceberg-local-
ecs".ambar_local_sf1k_partitioned.ship_mode;

CREATE VIEW "iceberg-local-
ecs".stargate_view_sf1k_aws_fact_local_dim.time_dim AS SELECT
* FROM "iceberg-local-
ecs".ambar_local_sf1k_partitioned.time_dim;
```

6. Run the derived TPC-DS queries using the catalog and schema:

```
iceberg.stargate_view_sf1k_aws_fact_local_dim
```

B TPC-DS derived Queries Generation and Required Modifications

To generate derived TPC-DS queries, use the *dsdgen* utility with the following options:

```
tools/dsqgen \
-DIRECTORY query_templates \
-INPUT query_templates/templates.lst \
-VERBOSE Y \
-QUALIFY Y \
-SCALE 1000 \
-DIALECT netezza \
-OUTPUT_DIR /tmp
```

Syntax modifications are required in 18 of the 99 queries before being executed. These changes do not alter the target result set of queries, and they are not expected to offer any advantage to the query engine over the original queries. Specifically, the syntax changes pertain to:

- date type casting (from string)
- including 'interval' for a number that needs to be added to date

As an example, we have an original query of:

```
  d_date between '2002-4-01' and
         (cast('2002-4-01' as date) + 60 days)
```

and the modified query is:

```
  d_date between cast('2002-4-01' as date) and
         (cast('2002-4-01' as date) + interval '60' day)
```

C TPC-DS derived Data Generation Steps

This is an example on how to generate “`store_sales`” table CTAS statements to load the data:

```
create table iceberg.<dest_schema>.store_sales with (
    format = 'PARQUET',
    partitioning = ARRAY['ss_sold_date_sk']
) as
select * from iceberg.<source>.store_sales where ss_sold_date_sk
between 2450816 and 2450916;
insert into iceberg.<dest_schema>.store_sales select * from
iceberg.<source>.store_sales where ss_sold_date_sk
between 2450917 and 2451017;
insert into iceberg.<dest_schema>.store_sales select * from
iceberg.<source>.store_sales where ss_sold_date_sk
between 2451018 and 2451118;
.
.
.
insert into iceberg.<dest_schema>.store_sales select * from
iceberg.<source>.store_sales where ss_sold_date_sk is NULL;
```

This is an example of Alter table statements to compact the partitions:

```
alter table store_sales execute optimize(file_size_threshold
=> '1000MB') where ss_sold_date_sk between 2450816 and 2450916;
alter table store_sales execute optimize(file_size_threshold
=> '1000MB') where ss_sold_date_sk between 2450917 and 2451017;
alter table store_sales execute optimize(file_size_threshold
=> '1000MB') where ss_sold_date_sk between 2451018 and 2451118;
.
.
.
alter table store_sales execute optimize(file_size_threshold
=> '1000MB') where ss_sold_date_sk is NULL;
```