Starburst
ENTERPRISE PRESTO

Starburst Data

Technical Solution Brief

In this Document

Presto | Starburst Completes Presto for the Enterprise

Presto: Open-source, fast and scalable distributed SQL engine.

Presto is a high performance distributed SQL engine for running fast analytic queries against various data sources ranging in size from gigabytes to petabytes. Architected for the separation of storage and compute, Presto scales on demand, and eliminates the time and cost of integrating disparate data into a single data warehouse.

Separation of Compute & Storage

Architected to separate storage and compute, Presto can scale up and down based on your analytics demand to access your data. There's no need to move your data and provisioning compute to the exact need results in significant cost savings.

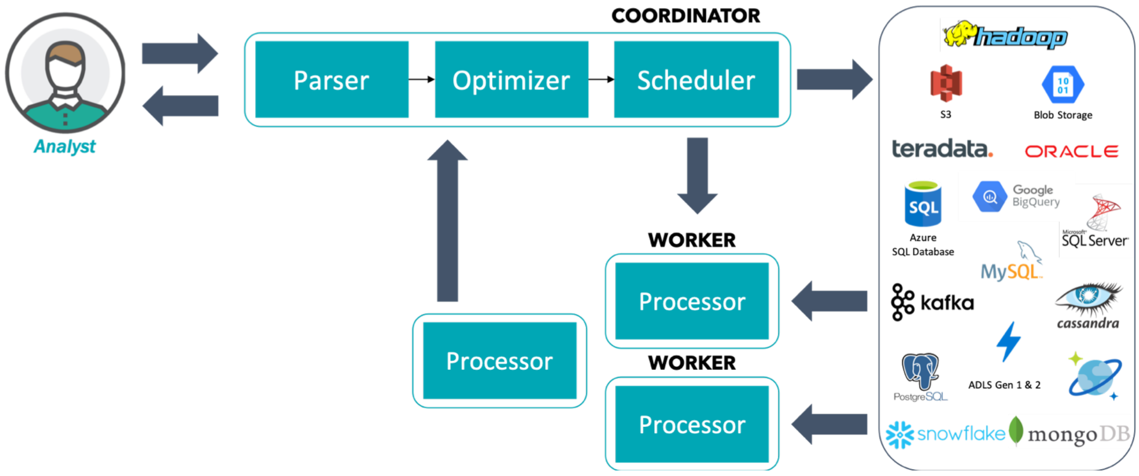
Analyze Anything

Allow your data scientists to use their favorite BI tools (e.g., Tableau, Qlik, Superset) to access and analyze virtually any data source. Presto's flexible architecture allows one to perform analytics federated across multiple data sources at the same time.

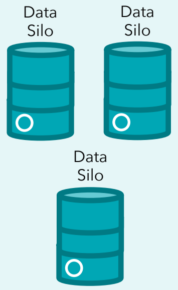
Deploy Anywhere

The separation of storage and compute allows Presto flexible deployment options. Deploy on public clouds such as AWS, Azure and GCP in addition to private clouds such as OpenStack, on premises on bare metal commodity hardware or a virtualized environment with a Kubernetes containerized deployment.

How Presto Works



Challenges | Integration in a Big Data Environment

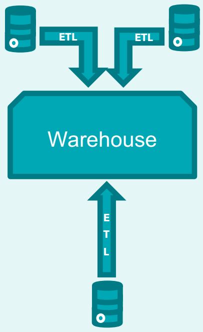


Data Access Difficulties

The Consumers – Several roles within a company require access to data in order to fulfill their function. Such roles include data analysts, data scientists and business intelligence.

The Procurers – Technical staff are responsible for making the data available when needed while at the same time meeting security requirements to ensure that only those individuals authorized to do so have access.

The Problem - These roles are needed to support a number of critical business functions. Data, however, exists in a variety of locations and often in different formats. A solution that provides integrated data access is needed.



The Traditional Data Warehouse Approach

Traditionally, the problem above is solved by building a data warehouse.

Advantages:

+ Aggregation of data from disparate sources into a single provisioned relational source for access.

Disadvantages:

- Costly to build and maintain
- ETL pipelines are required to migrate the data into the warehouse which itself is costly
- Time lag between when source data is accumulated before it is written to the warehouse
- No separation of compute and storage which means that you must ALWAYS overbuy what your anticipated requirements are.

The Modern Data Lake Approach

A more modern solution is to build a **data lake:**

- **On-premise** with Hadoop clusters
- **In the cloud** with unstructured object storage in a cloud provider such as Amazon AWS, Microsoft Azure or Google Cloud.

Advantages over warehouse:

+ Cheaper storage costs
+ Overall less costly to build.

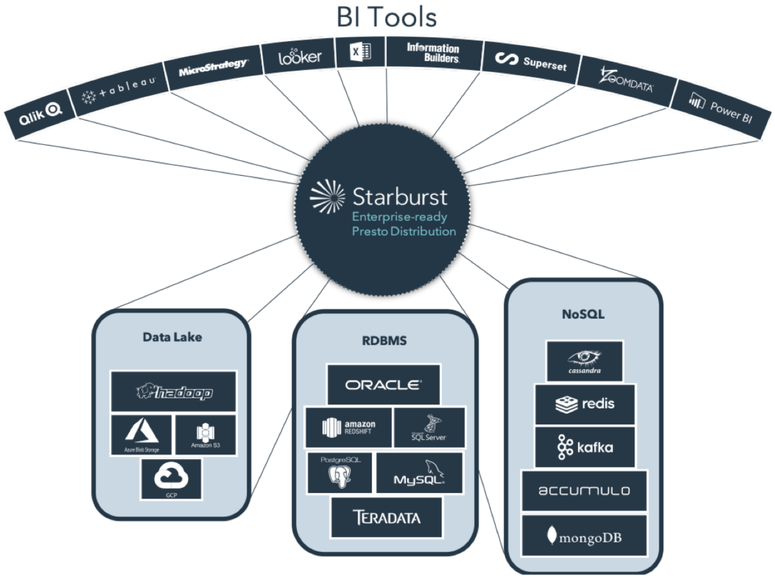
Challenges:

- Extracting the data in a meaningful way
- Making the right data available in the right form for the data consumers and data procurers.
  (Note that these were the key drivers for building a data-lake in the first place)

Solution | Integration in a Big Data Environment

Starburst Presto and Data Virtualization

Starburst Presto acts as a middle layer or “consumption layer” to provide your data consumers with federated virtual access to your data sources.



For Your Data Consumers:

- Access to data appears to be in a relational database irrespective of where the data actually resides.
- Data from disparate sources, or stored in different formats can be joined and operated on as if it was in the same database.
- Starburst Presto abstracts the pain points of warehouses and data lakes so your data consumers can continue to operate in their favorite BI tool or SQL client of choice.

How Starburst Presto Works:

- Data sources are accessed through a set of connectors that are configured to serve respective data up to the client through a SQL interface.
- The source data itself is never actually stored within Starburst Presto.
- Once a connector has been registered, the data in the source *immediately* becomes available through an ODBC/JDBC connection or command line interface.
- Presto is ANSI SQL compliant to support all queries and reporting requirements.

There are several typical use cases for which this approach is advantageous.

Use Cases | Integration in a Big Data Environment

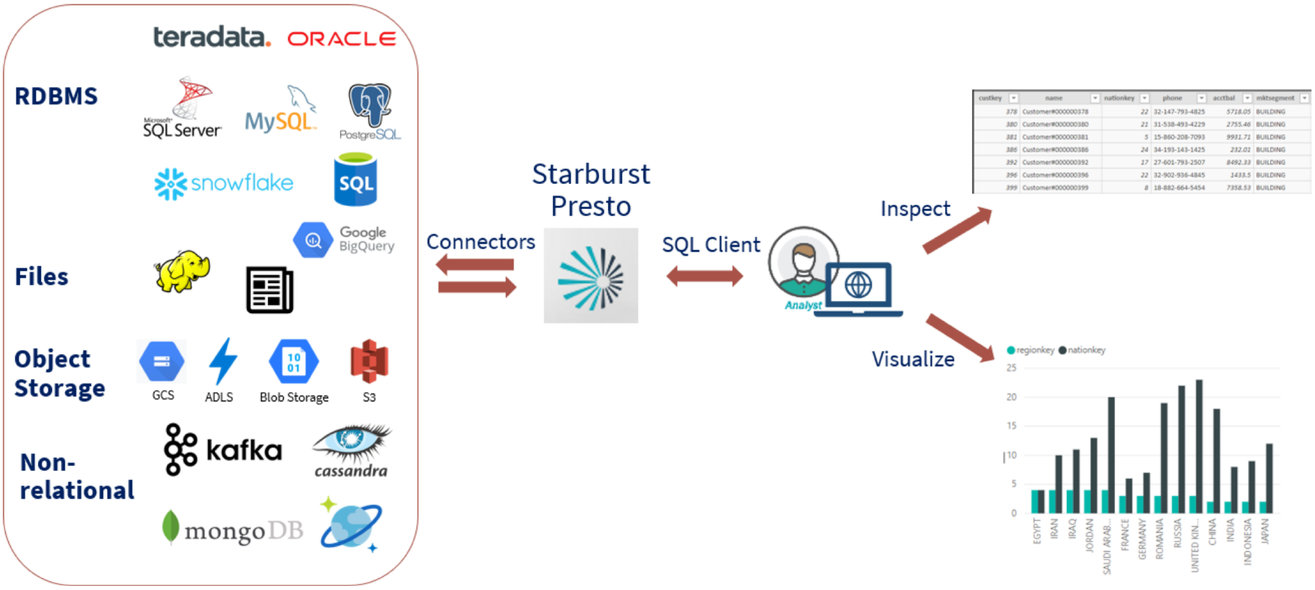
Use Case #1: Interactive Data Investigation

Rapid ad-hoc interactive queries and analysis is important for several data consumer roles.

They require *fast* interactive communication with their data to get an initial understanding before writing reports, performing analysis, writing predictive models, or investigating data and data profiles of data in sources as DBAs may need to do.

With Starburst Presto, consumers can query underlying sources from their SQL or BI tool of choice. A typical example would be using your favorite SQL client such as SQuirrel or Toad or PowerBI to return and inspect raw data.

Data can be queried rapidly from single source or combined through federated joins.



custkey	name	nationkey	phone	acctbal	mktsegment
378	Customer#000000378	22	32-147-793-4825	5718.05	BUILDING
380	Customer#000000380	21	31-538-493-4229	2755.46	BUILDING
381	Customer#000000381	5	15-860-208-7093	9931.71	BUILDING
386	Customer#000000386	24	34-193-143-1425	232.01	BUILDING
392	Customer#000000392	17	27-601-793-2507	8492.33	BUILDING
396	Customer#000000396	22	32-902-936-4845	1433.5	BUILDING
399	Customer#000000399	8	18-882-664-5454	7358.53	BUILDING

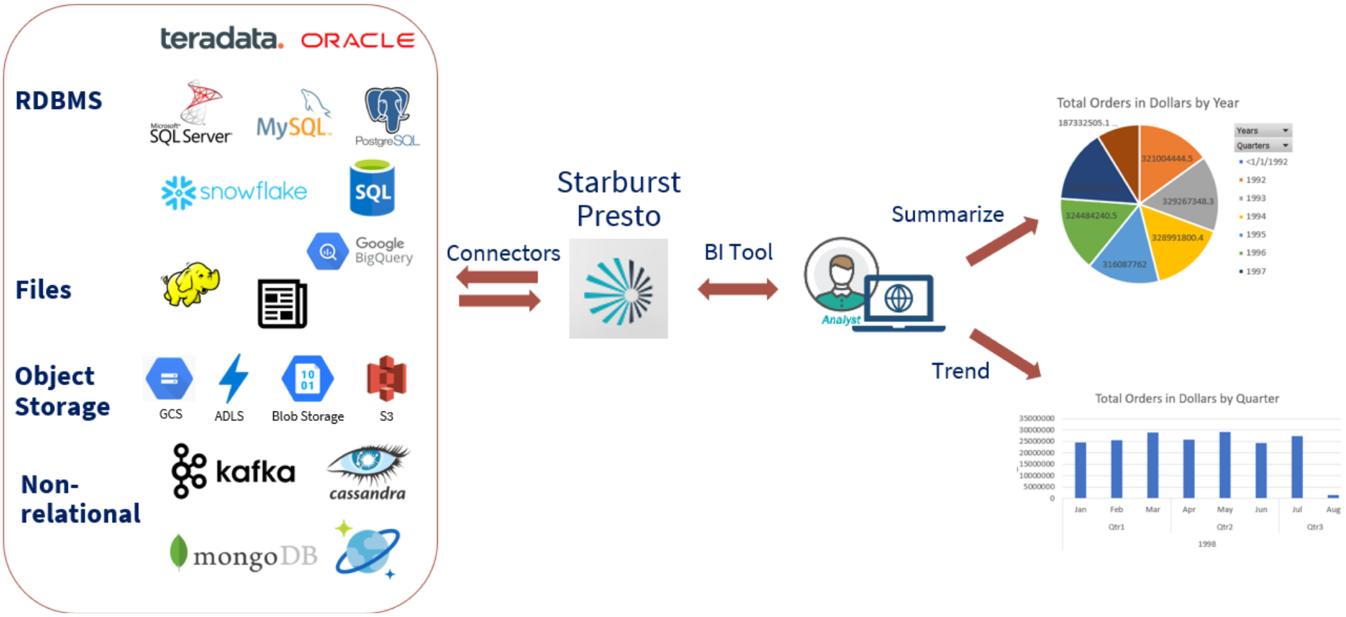
Use Case #2: Business Intelligence Dashboards and Reports

Business intelligence dashboarding and reporting requires rapid access to data in either single source or federated.

Data consumers need data access for a variety of reporting fuctions which includes risk and regulatory, marketing and financial reporting.

With Starburst Presto, we already know that consumers can work in their favorite BI tool of choice such as Tableau, Microstrategy or Qlik for dashboarding and reporting. Dashboards may need to be highly interactively with responsiveness on the order of a second, while some may have longer times and can respond overnight depending on SLAs.

Starburst Presto's separation of compute and storage gives you the flexibility to build your cluster to meet your needs. Additionally, auto-scaling functionality means that the cluster can expand and contract interactively to meet changing capacity requirements to minimize cost (more on this later).



Use Case #3: Data Science

Data scientists need access to data for model development and machine learning purposes to support a variety of lines of business such as marketing and customer segmentation and attrition risk, financial risk, fraud detection and data quality.

Data scientists work with a variety of tools and languages such as Python, R and Spark to name a few. They need rapid ad-hoc access to data sources while constructing and investigating new data models. Once a model is in production they also require access in much larger data volumes. Additionally, historical data from archival sources often needs to be analyzed with current data sources, which means the ability to federate sources is required.

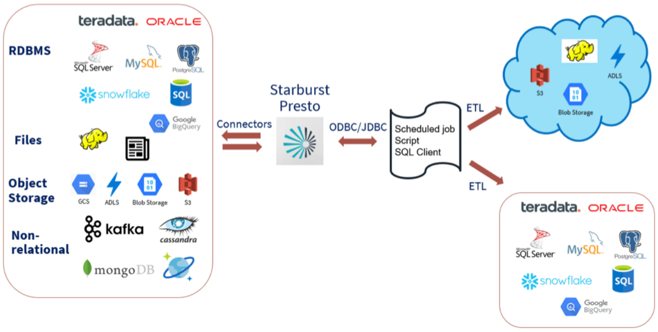
A solution that provides rapid access to single source data like relational or unstructured blob storage, or federated access is needed.

Starburst Presto fulfills these requirement by making it possible for data scientists to rapidly access large volumes of source data into their tool or language of choice through standard a ODBC/JDBC package interface.

Use Case #4: ETL Workloads

In addition to reporting and dashboarding, SQL is often used for ETL data migration and transformation.

Starburst Presto is ANSI SQL compliant for support of create table, update, insert and delete statements. This means that Starburst Presto can act as the SQL engine for ETL jobs, providing a single platform for both query and data migration needs.



Starburst Presto Advantages

Advantages

Performance and flexibility | Timeliness | Cost Reduction

Performance and Flexibility

Features

- **Separation of Compute & Storage**

  Starburst Presto separates your compute layer from storage. This allows you to scale compute independently to meet the service requirements of individual data consumers (time, capacity, concurrencies).

- **"Query Where it Lies"**

  With data virtualization and a "query it where it lies" methodology, data consumers have access to real-time data spread out across multiple locations. They also have data access during its journey to the data lake or cloud.

- **Storage Optionality**

  With federated data access you can choose which storage method best fits a given scenario. A typical example is hot vs cold storage where archival data can be kept in cheaper storage such as a data lake while current hot data is maintained in a relational DB.

- **Flexibility**

  Flexibility means being able to respond when use cases and requirements change. This also means avoiding vendor and data lock-in.

  Starburst Presto can be deployed in the public cloud (AWS, Azure and GCP), on a private cloud or Hadoop data lake or on on-premise VMs or bare metal.

  By providing access to data in any source in any format, if your data exists in one location or format today and in another tomorrow you will always be in a position to provide data access.

Starburst Presto Advantages

Advantages

Performance and flexibility Timeliness Cost Reduction

Performance and Flexibility

How is this achieved?

Starburst Presto's execution engine was built from the ground up to:

- Handle the largest data volumes
- Handle high concurrency and complex SQL statements
- Achieve rapid response times on the order of seconds

This is done through an MPP style **pipelined in-memory** execution architecture. It scales linearly when additional nodes or processing power is added to meet workload demand

Other architectural features that result in high performance:

- **Columnar** and **vectorized** data processing
- Runtime query **bytecode compilation**
- **Memory efficient** data structures
- Multi-threaded **multi-core** execution
- Optimized readers for **columnar formats** (ORC and Parquet)
- Predicate and column projection **pushdown**
- **Cost-Based Optimizer (more on this later)**

Starburst Presto Advantages

Advantages

Performance and flexibility

Timeliness

Cost Reduction

Timeliness

Sometimes analytics needs to be performed on recent data, for example in real time dashboarding.

- **Immediate data access** as soon as it is written by registering data in connectors in a federated manner.
- **Eliminate lag** associated with ETL migration when querying directly from the source.
- **Reduce time to market** by eliminating “data access” as a bottleneck in the process of furnishing reports and dashboards.

Cost Reduction

Data redundancy is expensive and risky to both produce and maintain.

- **Virtualization** makes it possible to reduce (or eliminate) your data warehouse footprint.
- **Reduce costs** of ETL development and maintenance associated with data migration into a data warehouse by querying the data where it lies.
- **Leverage storage savings** by storing “cold” data in low-cost options like a data lakes and “hot” data in a typical relational or non-relational database. Maintain simultaneous access to both with federated query capability.

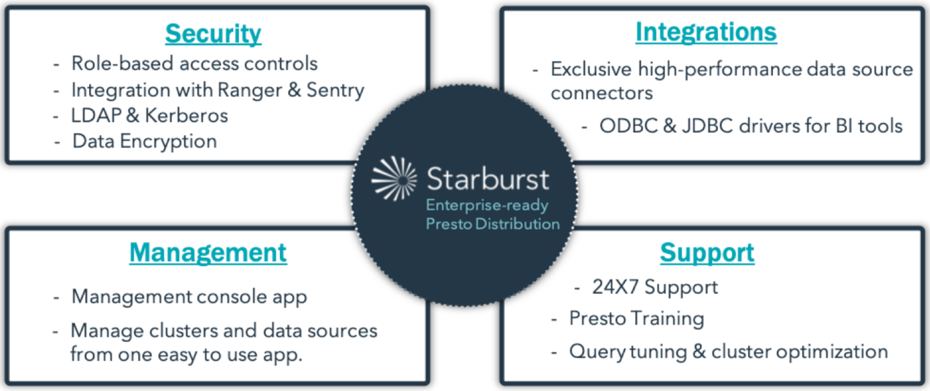
Starburst Presto Enterprise Edition

Starburst Presto Enterprise Distribution

First and foremost, Starburst is committed to the open source Presto project. Our engineers frequently contribute back to Presto's core functionality, core performance, stability and bug fixes. **These contributions can be found in Starburst Presto Community Edition (free to download from our website).**

Starburst also provides an **Enterprise Distribution**, which contains additional features not found in the Community Edition:

- Additional Certified Connectors and Drivers
- The Latest Cost-Based Query Optimization Features
- Custom Platform Support & Integrations
- Mission Control (Management Console)
- Additional Security Controls & Integrations



Security
- Role-based access controls
- Integration with Ranger & Sentry
- LDAP & Kerberos
- Data Encryption

Integrations
- Exclusive high-performance data source connectors
  - ODBC & JDBC drivers for BI tools

Starburst
Enterprise-ready
Presto Distribution

Management
- Management console app
- Manage clusters and data sources from one easy to use app.

Support
- 24X7 Support
- Presto Training
- Query tuning & cluster optimization

Support

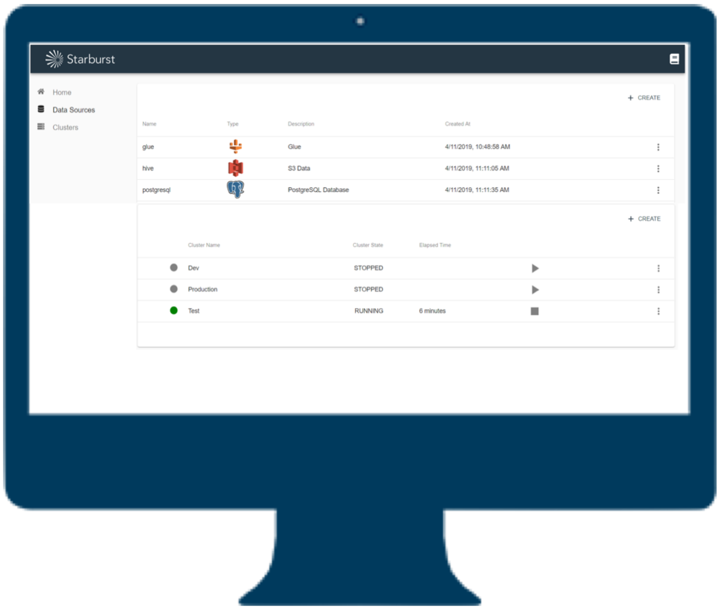
We are the Presto experts. With Starburst Presto Enterprise we offer:

- 24x7 support
- Rapid severity 1 response times
- Contact channels over email, webex and telecon

Starburst Presto Enterprise Edition

Mission Control

Starburst Mission Control is a management console that allows you to easily create, access, and manage Presto clusters. Mission Control makes it simple to connect data sources to Presto clusters allowing you to seamlessly access the data you need, no matter its location.



Name	Type	Description	Created At
glue		Glue	4/11/2019, 10:48:58 AM
hive		S3 Data	4/11/2019, 11:11:05 AM
postgresql		PostgreSQL Database	4/11/2019, 11:11:35 AM

	Cluster Name	Cluster State	Elapsed Time
●	Dev	STOPPED	
●	Production	STOPPED	
●	Test	RUNNING	6 minutes

Mission Control Features:

Web UI based management tool

Enables data architects to easily create, access, and manage their Starburst Presto deployment from a single, unified, easy-to-use UI.

Cluster management

Create new Presto clusters and point to the data sources that should be included.

Also, select the desired resources such as:

- Coordinator and worker node instance types
- Number of worker nodes
- Security group
- Access credentials
- Configure for coordinator high availability
- Launch Apache Superset.

Simple to connect to and configure data sources.

Select the desired data sources that are relevant for your Starburst Presto deployments such as S3 or relational databases and fill in configuration details.

Cloud formation template (CFT) configuration for AWS

Simplify the configuration and selection of resources for your CFT. Mission control will create a CFT according to your specifications, hook up and configure access to the data sources that have been requested and spin up the cluster with the parameters that have been set. Repeat the process as needed for any number of clusters.

Starburst Presto Enterprise Edition

Certified Connectors

The Enterprise Edition provides additional, certified proprietary connectors not found in the Community Edition

The list includes **Parallel Teradata**, **Parallel Snowflake**, **Oracle**, **BigQuery**, **MapR, and DB2**. Additionally, **certified ODBC and JDBC driv**ers are included as is **AWS Glue metastore integration** which is used to maintain data sources and statistics for cost based query optimization.

Security

Providing access to enterprise data means ensuring that the correct security measures are in place. The enterprise distribution enables you to control who has access to what data sources.

Permissions

Specify permission through integration with LDAP and Kerberos in addition to Apache Ranger and Sentry for role based access control. That includes both column level masking and row level filtering.

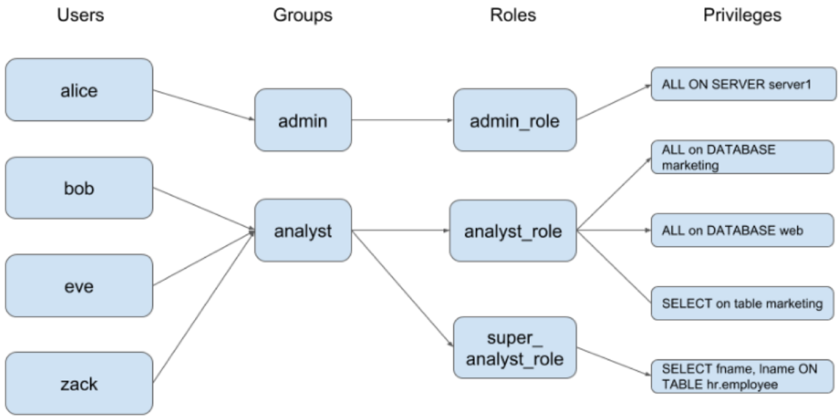
Audit Logs

Presto keeps an audit log that details the time a query was submitted, the user who submitted the query and the details of the query.

Ranger and Sentry Integration

Presto enforces the policies maintained in Ranger and Sentry. Authorization and authentication credentials are maintained in your source system of choice and altering those policies must be done through the Ranger or Sentry UI.

When a query is submitted, Presto parses and analyzes the query to understand the privileges needed to access the required data.   Presto communicates with the Apache Ranger or Sentry service to determine if the request is valid. If the request is valid, the query continues to execute. If the request is invalid, an error is returned to the user.



Starburst Presto Enterprise Edition

Cost-Based Query Optimization

A crucial component of Starburst Presto is the speed with which results are returned.

Query Planning

Different query plans can have a large (orders of magnitude large) impact on performance. Identifying and choosing an optimal query plan helps to meet response requirements while keeping cluster size down and reducing costs. Presto will automatically push down SQL into the source DB where appropriate to avoid transferring an unnecessary amount of data - for example when a filter is applied in the query.

Query Optimization

This feature takes advantage of table statistics maintained in a metastore such as Hive or Glue. Statistics are harvested from the output of analyze table commands that can be made partition specific. Table statistics are provided to the query planner by connectors for query optimization. Examples of optimization include Join reordering and Join type selection.

Join Execution

Data transfer over the network is a common bottleneck in execution times. With optimal join ordering the amount of data transferred over the network can be minimized, for example by joining two large tables later in execution. Starburst Presto will automatically pick the join order with lowest computed cost using table statistics. Joins can also be accomplished via repartitioning or broadcasting.

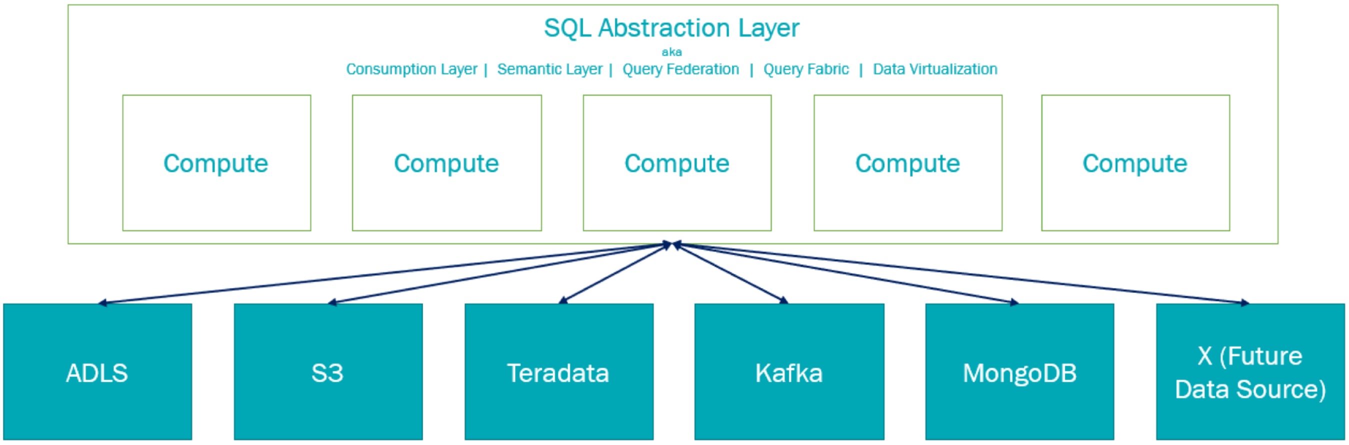
Broadcasting Join

All of the data for a given table is replicated and distributed to each node in the cluster - this works well when one of the tables is relatively small, can fit in memory, and is being joined to a much larger table.

Partitioned Join

Both tables are re-distributed based on a hash of the join key. Partitioned joins are effective for larger join sizes and larger tables.

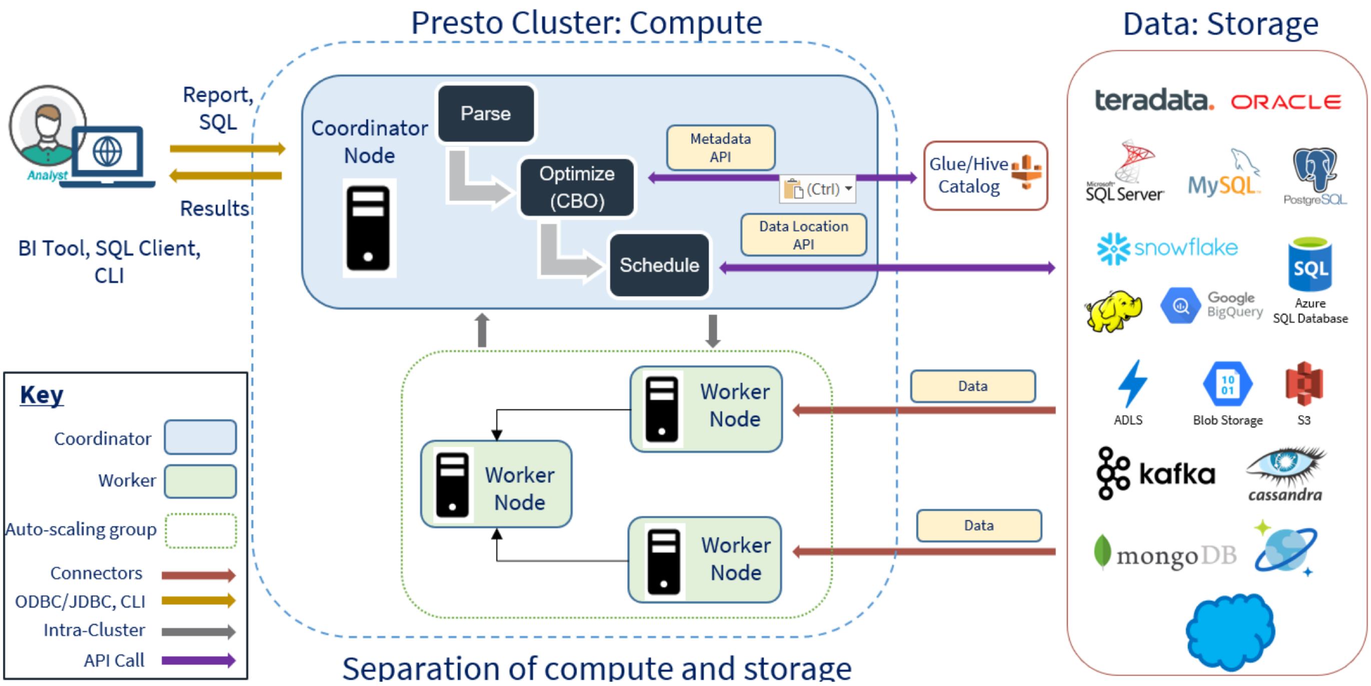
The cost based optimizer will choose the more efficient method depending on the acquired table statistics. Query Optimization is continuously being improved and the latest CBO improvements are included in the enterprise edition.



Architecture

How It Works: Architecture and Execution Environment

The Presto cluster is separate from the storage infrastructure and can be scaled independently to meet the requirements of individual use cases. Presto maintains only the information needed to connect to the source data through connectors and never stores the data itself.



Architecture

Presto Execution Process

Presto Clusters

A cluster is composed of a coordinator node and one or more worker nodes. In the most simple environment a single machine can act as both coordinator and worker but for practical production purposes a capacity planning exercise should inform the number of worker nodes and their machine type (memory and CPU requirements). The coordinator node acts as the brain of the cluster and is the host that the user connects to - either by ODBC/JDBC for BI tool or SQL client use or through SSH for command line work.

Query Execution

The coordinator intercepts the SQL query and executes a parse - optimize - schedule sequence of events. The query is initially parsed before going through a query planning and optimization step. The optimizer uses data source statistics maintained in in a metadata catalog such as Hive or Glue to generate an optimal query plan. The scheduler interrogates the location of the various data sources requested from the query plan before the coordinator assigns work to the worker nodes in the cluster.

Connecting with Data Sources

Data source specific connectors unload and stream data from the sources to the worker nodes, the data is not landed to disk. Where possible, SQL is pushed down to the source systems so that only the required amount of data is unloaded.

For example, if a filter is applied in the SQL query, the filter is executed in the source DB so that unnecessary data is not unloaded and transferred to the Presto cluster. The worker nodes in the cluster execute the query plan in a series of stages before communicating the result back to the user either to the BI Tool/SQL client or command line interface.

Additional Features

- For security purposes encryption is supported for both intra-cluster data streaming and data at rest.
- Processing is done in memory for optimal performance.
- In order to avoid failing a query it is possible to spill to disk when exceeding available memory. Spill to disk parameters make it possible to configure memory thresholds and spill behavior such as the number of spill threads to suit your needs.

For performance reasons we strongly recommend deploying your Presto cluster local to the data to minimize the impact of network transfer. As an example, for cloud deployments, spin up the Presto cluster in a region local to the data such as S3 or Azure blob and for and for on-premise deploy the cluster in a datacenter local to the Hadoop cluster.

For best performance, machine types should be selected with enough memory and CPU to suit your use cases. **Our consulting services can assist in capacity planning exercises to help you both appropriately size the cluster and select the correct instance types for cloud deployments to get the most from your cluster.**

Starburst

Starburstdata.com | hello@starburstdata.com

Twitter: @StarburstData