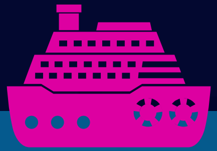
trino ICEBERG



St Louis - Apr 24, 2024

Lester Martin: Educational Engineer @ Starburst

Connection Before Content

Lester Martin - https://about.me/lestermartin

- Educational Engineer @ Starburst
  - Build the content
  - Teach the class
  - Repeat
- 30 years of technology experience
  - Started my journey on a TRS-80 Model III
  - Played most every role, but consider myself a programmer at my core
  - Half of career in transactional systems and the second half in analytical processing
  - A DECADE of "big data" experience to include
    - Trino/Starburst, Hadoop, Hive, Spark
    - NiFi, Kafka, Storm, Flink
    - HBase, MongoDB

trino

Agenda

1. How did we get here?
2. What is Trino?
3. What is Iceberg?
4. Modern data lake architecture
5. Trino & Iceberg current state
6. DEMO

trino

Big data keeps getting bigger

- More data = need for better, faster tech
- Hadoop and Hive were originally the superstars
- But eventually even their performance was too slow
  - Data scientists at Facebook were limited to as few as 10 queries a day
  - Enter Presto (soon to be Trino)

Let’s go Trino

The best query engine in the world

trino

What is Trino?

Trino *(formerly known as Presto)* is a fast distributed SQL query engine designed to query large data sets distributed over one or more heterogeneous data sources.

Open Source

- User-driven development
- Large and active community
- Huge variety of users
- Apache license, version 2.0

Highly Performant

- ANSI SQL MPP Query Engine
- Cost-Based Query Optimizer
- High Concurrency
- Proven Scalability

Flexible Query Engine

- Scale storage and compute independently
- No ETL or data integration necessary to get to insights from multiple sources
- Powerful and capable with ETL
- SQL-on-Anything

No Vendor Lock-In

- No Hadoop distro vendor lock-in
- No storage engine vendor lock-in
- No data/file format lock-in
- No cloud vendor lock-in
- No database lock-in

https://trino.io

trino

Where should I use it?

Interactive data analytics

Enter a SQL query for Trino to process and return results as quickly as possible.

- Query large amounts of data
- Test hypotheses
- Run A/B testing
- Build visualizations

High performance data lake analytics

Trino enables users to run SQL based analytics on HDFS/Hive and cloud object storage

- Run petabyte scale analytics
- Scale and performance benefits

Federated analytics

Create a single point of access by using Trino to query disparate data sources.

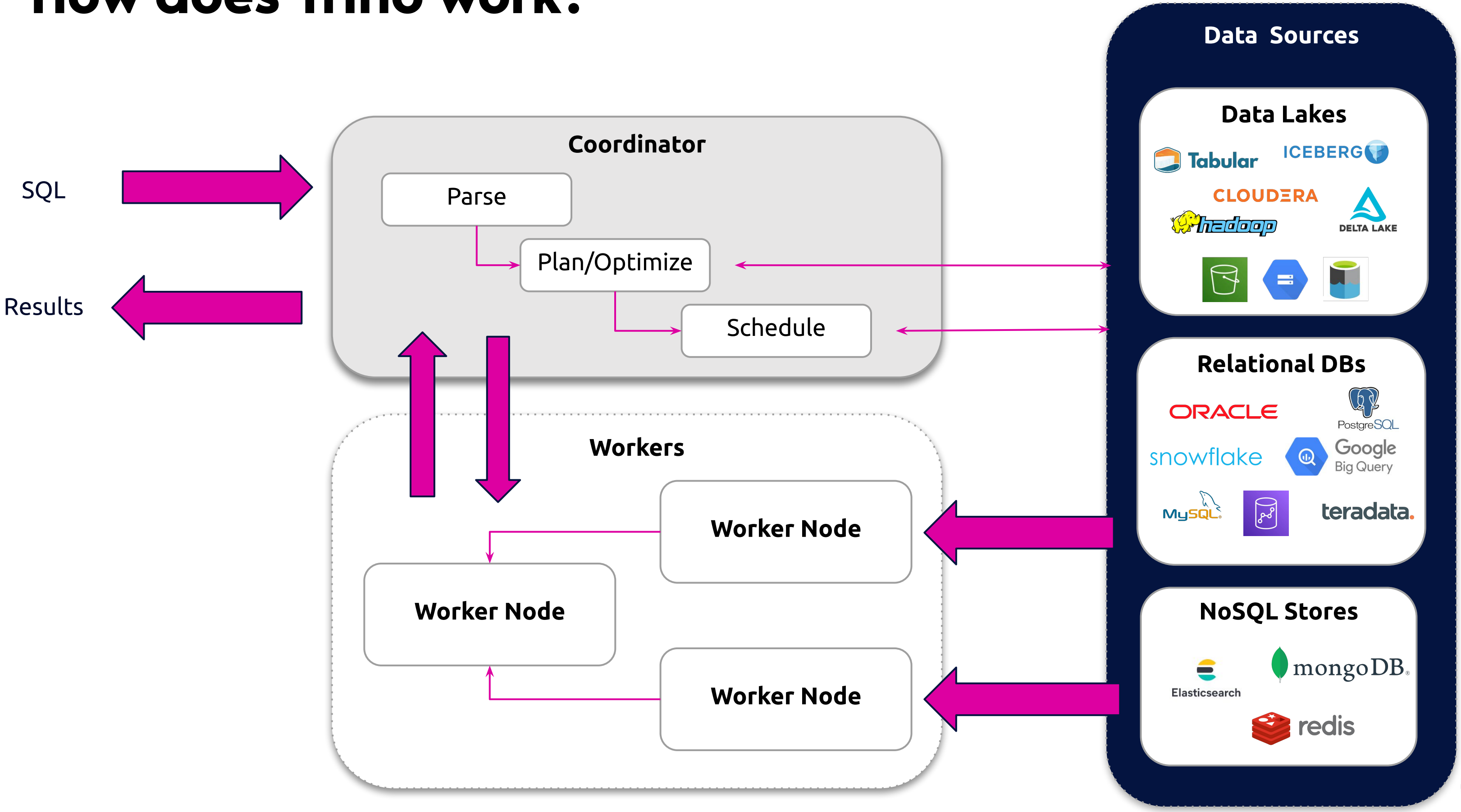
- Object storage
- Relational systems
- Streaming systems
- NoSQL systems

Batch ETL processing

Run resource intensive ETL processes in batches without fear of failure with Trino.

- Use SQL with every data source
- Work with numerous data sources and targets all in the same system
- Ensure speed and reliability

How does Trino work?



SQL

Results

Coordinator

Parse

Plan/Optimize

Schedule

Workers

Worker Node

Worker Node

Worker Node

Data Sources

Data Lakes

Tabular ICEBERG CLOUDERA hadoop DELTA LAKE

Relational DBs

ORACLE PostgreSQL snowflake Google Big Query MySQL teradata.

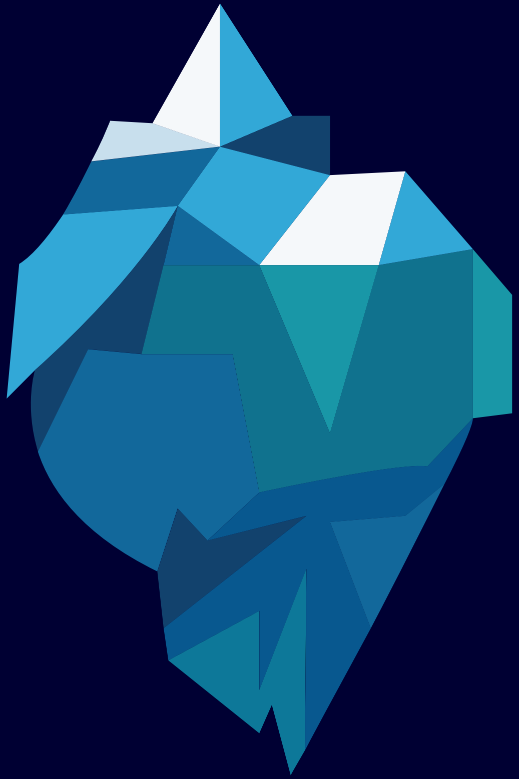
NoSQL Stores

Elasticsearch mongoDB. redis

trino

Iceberg Origins

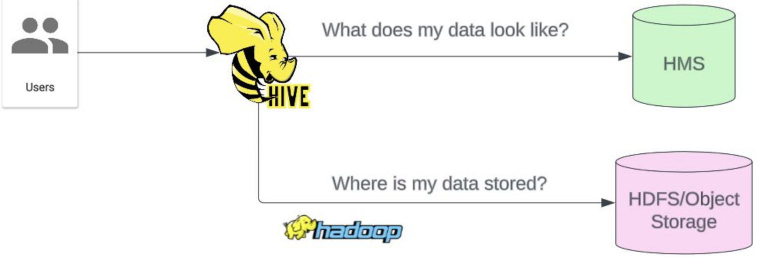
What came before?



trino

The challenges of the invisible Hive "spec"

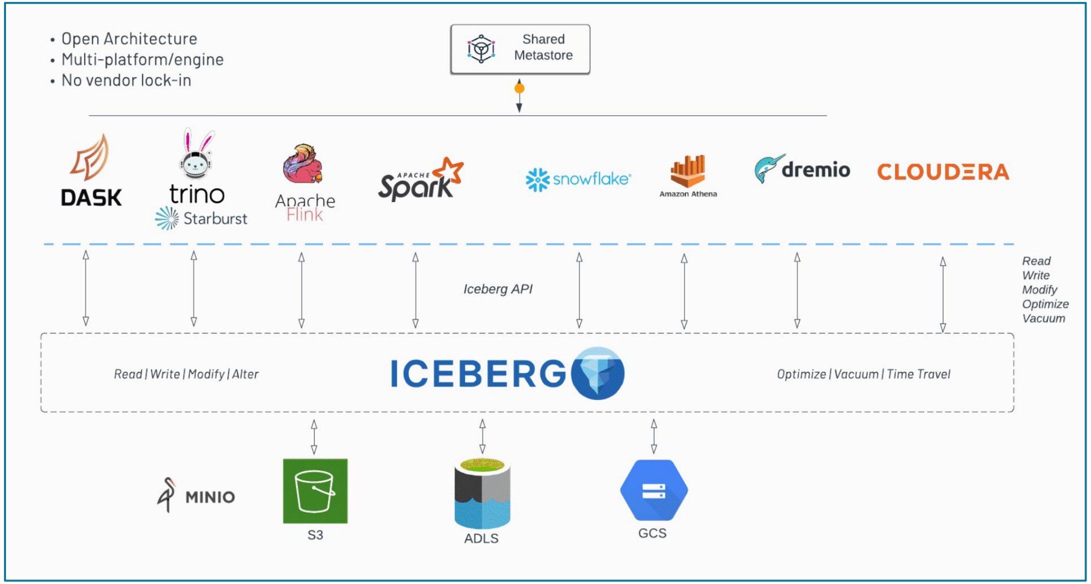
- **Partitioning based on column names at the end of the table** which match directory names on the file system (users must know this)
- **Partitions are rigid**
- **Partial schema evolution**
- Transactional/ACID has always been squirrelly (inconsistency, correctness issue)
- Not optimized for object storage (need list of folders + scan all files in each folder)



Apache Iceberg

- Created by Ryan Blue & Daniel Weeks at **Netflix** in 2017.
- Solve the challenges of performance, data modification and schema evolution in the lake.
- Uses open data concepts (ORC, Parquet, Avro) and architecture.
- Seen enormous interest and adoption over the last 3 years.

Multi-Engine Platform



- Open Architecture
- Multi-platform/engine
- No vendor lock-in

Iceberg should be invisible

Behaves like a warehouse

Avoid unpleasant surprises

- No zombie data
- Performance is not mysterious
- Reduced metastore reliance

Doesn't steal attention

- Fast metadata operations
- Automate the boring stuff
- Fix problems without migration

Optimistic Concurrency

- Allows multiple writes simultaneously, checks for conflicts before final commit

Universal open standard

ICEBERG

trino

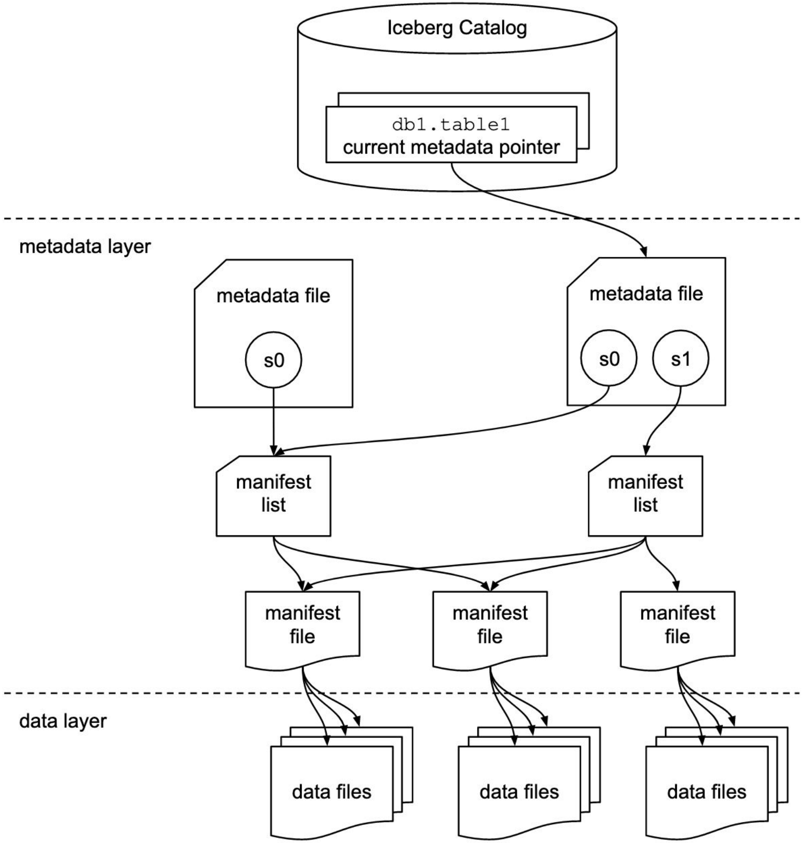
Architecture

Comprised of a **hierarchy of metadata files** to accommodate constant changes to a table (insert, delete, update, schema migration, partition changes).

Think of a **database transaction log but using an object store for the storage.**

Metadata:

- **Iceberg catalog** (HMS/Glue/JDBC) - Stores the file path for the "current" metadata file.
- **Metadata file** (json) - Stores information about table (schema/partition/etc) at a given point in time and details + pointers to snapshots (manifest list).
- **Manifest list** (avro) - Contains statistics for a collection of files that represent a single snapshot.
- **Manifest file** (avro) - List of data files (orc, parquet, avro), pruning by partition and column stats.



ICEBERG

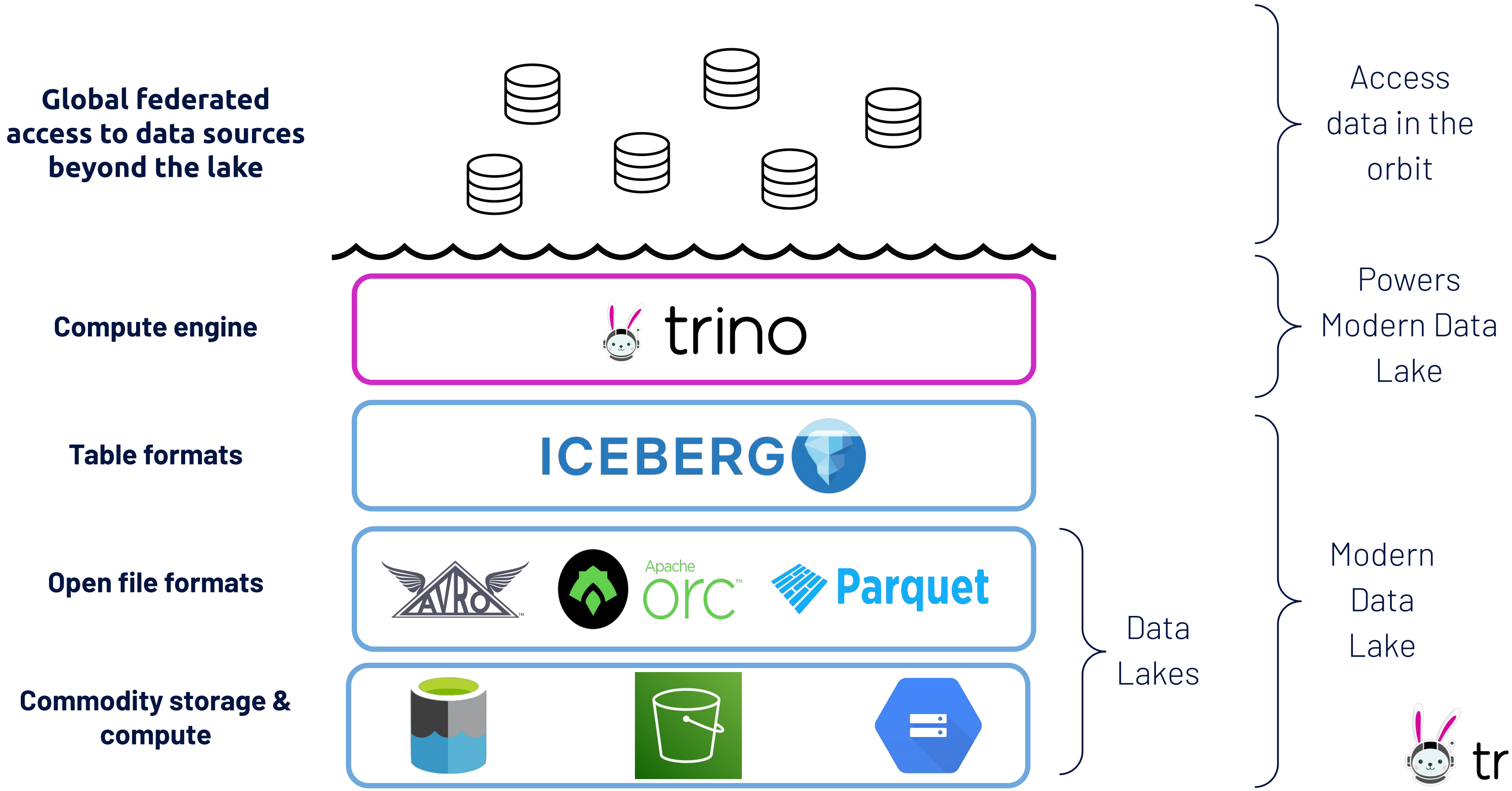
trino

Build a modern data lake

Trino + Iceberg = Happiness

trino

The Modern Data Lake



Global federated access to data sources beyond the lake

Compute engine

Table formats

Open file formats

Commodity storage & compute

Access data in the orbit

Powers Modern Data Lake

Data Lakes

Modern Data Lake

Modern Data Lake Benefits

Data Warehouse Benefits

- ACID transactions
- Fined grained access control
- Data quality
- High performance and concurrency
- Highly curated data
- Typically proprietary systems
- Best for business intelligence use cases

Data Lake Benefits

- Separation of storage & compute
- Large scale
- Cost efficient
- Open formats
- Structured and unstructured data
- Open source options
- Best for data science and data engineering use cases

trino

Features & Advantages

Full Table History

- Every operation is tracked
- Time Travel
- *SELECT … FOR VERSION AS OF TIMESTAMP '…';*

Real Schema Evolution

- No zombie data! (a classic Hive problem)
- Add/Remove/Rename/Retype columns for free
- *ALTER TABLE … ADD/DROP/RENAME/ALTER COLUMN…;*

Reduced Metastore Reliance

- Metastore ops are slow
- File System reads support high parallelism

Partition Evolution

- Data volumes aren't static
- Handle changes in data over time
- *ALTER TABLE … SET PROPERTIES (partitioning = …)*

Features & Advantages

Partition Transforms

- Partition on the year of a timestamp: *partitioning = ARRAY[year(orderdate)]*
- Partitioning is "hidden", users don't need to care
- Pick a new transform later

Performance

- No file listing
- Partition pruning
- Fine grain data skipping
- Automatic JOIN ordering

Iceberg DEMO

Running on Starburst Galaxy

Starburst Galaxy

Built for the cloud

Fully managed cloud data lake analytics built and supported by the creators of Trino

Available on leading public clouds

aws

Download SQL for your own testing…
https://raw.githubusercontent.com/lestermartin/events/main/2024-04-24_STL-TUG/sql.txt

trino