Apache ICEBERG

Concepts and explanations

Shwetank Singh
GritSetGrow - GSGLearn.com

Shwetank Singh
GritSetGrow - GSGLearn.com

DATA LAKEHOUSE

A data architecture pattern that combines the flexibility and scalability of data lakes with the data management and performance capabilities of data warehouses.

Integrating Apache Iceberg with a data lake to enable ACID transactions and schema evolution while maintaining the low-cost storage of a data lake.

Shwetank Singh
GritSetGrow - GSGLearn.com

APACHE ICEBERG

An open table format designed for large analytic datasets, providing features like schema evolution, partitioning, and time travel queries.

Using Apache Iceberg to manage petabyte-scale data stored in Amazon S3, enabling efficient queries and data management.

Shwetank Singh
GritSetGrow - GSGLearn.com

ICEBERG ARCHITECTURE

The structural design of Apache Iceberg, including data files, metadata files, manifest files, and the catalog. These components work together to enable efficient data management and querying.

Iceberg's architecture separates metadata from data, allowing for efficient query planning and execution.

DATA FILES

The actual files where the data is stored, typically in columnar formats like Parquet or ORC.

A Parquet file storing sales data in an Iceberg table.

Shwetank Singh
GritSetGrow - GSGLearn.com

METADATA FILES

Files that store metadata information such as schema, partitioning, and snapshots. These files help manage and query the data efficiently.

An Iceberg metadata file might include the schema definition and partitioning information for a table.

Shwetank Singh
GritSetGrow - GSGLearn.com

MANIFEST FILES

Metadata files that list data files, their statistics, and partition information. They are used to manage data efficiently and support operations like time travel and rollback.

A manifest file in Iceberg listing all the Parquet files and their row counts for a table.

follow for more content

Shwetank Singh
GritSetGrow - GSGLearn.com

CATALOG

A service that tracks the metadata of tables, such as schemas, versions, and locations, allowing users to find and interact with tables.

Using the Hive Metastore as the catalog for Iceberg tables, enabling integration with existing Hive infrastructure.

Shwetank Singh
GritSetGrow - GSGLearn.com

HOW ICEBERG WORKS

Iceberg manages large datasets by using a combination of metadata, manifest files, and a catalog to provide efficient data management and querying. It supports features like ACID transactions, schema evolution, and partitioning.

Creating an Iceberg table and performing queries using Apache Spark.

Shwetank Singh
GritSetGrow - GSGLearn.com

EFFICIENCY OF ICEBERG

Iceberg is efficient due to its use of metadata and manifest files, which allow for optimized query planning and execution. It reduces the need for expensive operations like full table scans by using partitioning and data skipping techniques.

Using Iceberg's time travel feature to query historical data without scanning the entire table.

Shwetank Singh
GritSetGrow - GSGLearn.com

SCHEMA EVOLUTION

The ability to change the schema of a table (e.g., add or remove columns) without affecting existing data.

```
ALTER TABLE my_table ADD COLUMN new_column STRING;
```

Shwetank Singh
GritSetGrow - GSGLearn.com

PARTITIONING

Dividing a table into smaller, more manageable pieces based on a specified column or set of columns.

```
CREATE TABLE sales (id INT, amount
DOUBLE, date STRING) PARTITIONED BY
(date);
```

Shwetank Singh
GritSetGrow - GSGLearn.com

TIME TRAVEL QUERIES

Queries that allow you to access data as it existed at a specific point in time.

```
SELECT * FROM my_table
FOR SYSTEM_TIME AS OF '2022-01-01
00:00:00';
```

ACID TRANSACTIONS

A set of properties that ensure reliable processing of database transactions: Atomicity, Consistency, Isolation, Durability.

Ensuring that a batch update operation on an Iceberg table is completed fully and consistently, even in the case of failures.

METADATA LAYER

A layer that manages metadata about the data files, such as schema, partitioning, and file locations, enabling efficient query planning and execution.

Apache Iceberg's metadata layer includes manifest files that list all data files and their statistics.

Shwetank Singh
GritSetGrow - GSGLearn.com

MANIFEST LISTS

Files that contain lists of manifest files, providing a way to efficiently manage large numbers of data files.

A manifest list in Iceberg might include references to several manifest files for a large table.

Shwetank Singh
GritSetGrow - GSGLearn.com

SNAPSHOTS

Point-in-time representations of a table's state, allowing for time travel queries and rollback operations.

```
SELECT * FROM my_table.snapshots;
```

MERGE-ON-READ

A data writing strategy where updates are written to delta files, which are merged with base files at read time.

```
MERGE INTO target_table USING source_table
ON target_table.id = source_table.id WHEN
MATCHED
THEN UPDATE
SET target_table.value = source_table.value
WHEN NOT MATCHED
THEN INSERT (id, value) VALUES
(source_table.id, source_table.value);
```

COPY-ON-WRITE

A data writing strategy where updates are written by creating new versions of data files, replacing the old files.

Using copy-on-write to handle batch updates to a fact table in Iceberg, ensuring consistent and optimized query performance.

COMPACTION

The process of combining smaller data files into larger ones to improve read performance.

```
CALL
iceberg.system.compact_table('my_table');
```

Shwetank Singh
GritSetGrow - GSGLearn.com

Z-ORDERING

A data sorting technique that improves query performance by clustering related data together.

```
ALTER TABLE my_table ORDER BY ZORDER
(column1, column2);
```

HIDDEN PARTITIONING

A technique where partitions are defined internally by the system rather than explicitly by the user, improving flexibility and performance.

Iceberg's hidden partitioning automatically manages partitions based on data characteristics, simplifying schema management.

DATA SKIPPING

A query optimization technique that avoids reading irrelevant data based on metadata information.

Iceberg uses data skipping to avoid reading files that do not match query predicates, reducing I/O and improving query performance.

Shwetank Singh
GritSetGrow - GSGLearn.com

BLOOM FILTERS

A space-efficient probabilistic data structure used to test whether an element is present in a set. It helps to optimize data skipping by quickly identifying relevant data files.

```
CREATE TABLE my_table USING iceberg
OPTIONS (bloom_filter_columns = 'id');
```

Shwetank Singh
GritSetGrow - GSGLearn.com

DATAFILE BLOOM FILTERS

Bloom filters applied to data files in Iceberg to quickly identify whether a file contains relevant data for a query.

A Bloom filter in an Iceberg data file helps skip irrelevant files during a query, reducing I/O.

Shwetank Singh
GritSetGrow - GSGLearn.com

CREATE TABLE

The operation to define a new table in Iceberg, specifying its schema and partitioning strategy.

```
CREATE TABLE my_table
(id INT, name STRING)
PARTITIONED BY (name);
```

INSERT DATA

The operation to add new data to an Iceberg table.

```
INSERT INTO my_table
VALUES (1, 'Alice'), (2, 'Bob');
```

UPDATE DATA

The operation to modify existing data in an Iceberg table.

```
UPDATE my_table
SET name = 'Charlie' WHERE id = 1;
```

DATA ENGINEERING - APACHE ICEBERG

Shwetank Singh
GritSetGrow - GSGLearn.com

DELETE DATA

The operation to remove data from an Iceberg table.

```
DELETE FROM my_table WHERE id = 2;
```

follow for more content

Shwetank Singh
GritSetGrow - GSGLearn.com

TIME TRAVEL

The ability to query data as it existed at a specific point in time.

```
SELECT * FROM my_table
FOR SYSTEM_TIME AS OF '2023-01-01';
```

Shwetank Singh
GritSetGrow - GSGLearn.com

PARTITION EVOLUTION

The ability to change the partitioning strategy of a table over time.

```
ALTER TABLE my_table
DROP PARTITION FIELD old_field
ADD PARTITION FIELD new_field;
```

Shwetank Singh
GritSetGrow - GSGLearn.com

TRANSACTION ISOLATION

Ensuring that concurrent transactions do not interfere with each other, maintaining data consistency.

Using snapshot isolation to ensure that each transaction operates on a consistent view of the data.

follow for more content

Shwetank Singh
GritSetGrow - GSGLearn.com

DATA VERSIONING

Keeping track of changes to the data over time, enabling rollback and historical queries.

Creating new snapshots for each data update to track versions over time.

DATA ENGINEERING - APACHE ICEBERG

follow for more content

Shwetank Singh
GritSetGrow - GSGLearn.com

BRANCHING

Creating separate branches of the table metadata to support different versions or experiments.

```
CALL
iceberg.system.create_branch('my_table',
branch = 'test_branch');
```

DATA ENGINEERING - APACHE ICEBERG

TAGGING

Assigning tags to specific snapshots for easy identification and access.

```
CALL iceberg.system.tag_snapshot
('my_table', snapshot_id = 12345,
tag = 'release_v1.0');
```

Thank you!

Shwetank Singh
GritSetGrow - GSGLearn.com

follow me for more content