Apache Iceberg

Ryan Blue
2019 Big Data Orchestration Summit

NETFLIX

Netflix's Data Warehouse

5-year Challenges

- Smarter processing engines
  - CBO, better join implementations
  - Result set caching, materialized views
- Reduce manual data maintenance
  - Data librarian services
  - Declarative instead of imperative

Problem Whack-a-mole

- Unsafe operations are everywhere
  - Writing to multiple partitions
  - Renaming a column
- Interaction with object stores causes major headaches
  - Eventual consistency to performance problems
  - Output committers can’t fix it
- Endless scale challenges

What is Iceberg?

Iceberg is a scalable format for tables with a lot of best practices built in.

A format?

We already have Parquet, Avro and ORC . . .

A table format.

A table format

- File formats help you modify or skip data in a single file
- Table formats do the same thing for a collection of files

- To demonstrate this, consider Hive tables . . .

Hive Tables

- Key idea: **organize data in a directory tree**

```
date=20180513/
 |- hour=18/
 |    |- ...
 |- hour=19/
 |    |- part-000.parquet
 |    |- ...
 |    |- part-031.parquet
 |- hour=20/
 |    |- ...
```

Hive Tables

- Filter: WHERE **date = '20180513' AND hour = 19**

```
date=20180513/
 |- hour=18/
 |    |- ...
 |- hour=19/
 |    |- part-000.parquet
 |    |- ...
 |    |- part-031.parquet
 |- hour=20/
 |    |- ...
```

Hive Metastore

- Problem: **too much directory listing** for large tables
- Solution: use HMS to track partitions

```
date=20180513/hour=19 -> hdfs:/.../date=20180513/hour=19
date=20180513/hour=20 -> hdfs:/.../date=20180513/hour=20
```

- The file system still tracks the files in each partition . . .

Hive Tables: Problems

- State is kept in both the **metastore** and in a **file system**
- Changes are not atomic without locking
- Requires directory listing
    - O(n) listing calls, n = # matching partitions
    - Eventual consistency breaks correctness

Hive Tables: Benefits

- Everything supports Hive tables*
  - Engines: Hive, Spark, Presto, Flink, Pig
  - Tools: Hudi, NiFi, Flume, Sqoop
- **Simplicity** and **ubiquity** have made Hive tables indispensable
- The whole ecosystem uses the same at-rest data!

Iceberg

Iceberg’s Goals

- **An open spec and community for at-rest data interchange**
  - Maintain a clear spec for the format
  - Design for multiple implementations across languages
  - Support needs across projects to avoid fragmentation

NETFLIX

Iceberg’s Goals

- **Improve scale and reliability**
    - Work on a single node, scale to a cluster
    - All changes are atomic, with serializable isolation
    - Native support for cloud object stores
    - Support many concurrent writers

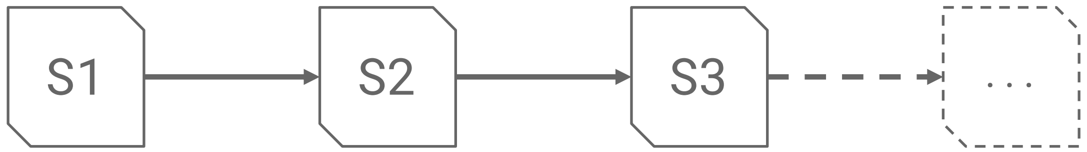
Iceberg's Goals

- **Fix persistent usability problems**
  - In-place evolution for schema and layout (no side-effects)
  - Hide partitioning: insulate queries from physical layout
  - Support time-travel, rollback, and metadata inspection
  - Configure tables, not jobs

- Tables should have **no unpleasant surprises**

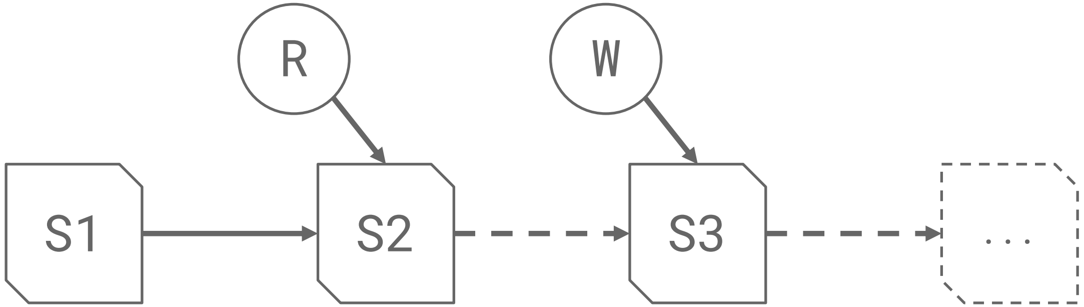
Iceberg's Design

- Key idea: **track all files in a table** over time
    - A **snapshot** is a complete list of files in a table
    - Each write produces and commits a new snapshot



Iceberg’s Design

- Readers use the current snapshot
- Writers optimistically create new snapshots, then commit



NETFLIX

In reality, it’s a bit more complicated.

Iceberg Design Benefits

- All changes are atomic
- No expensive (or inconsistent) file system operations
- Snapshots are indexed for scan planning on a single node
- CBO metrics are reliable
- Versions for incremental updates and **materialized views**

Iceberg at Netflix

Scale

- Production tables: **tens of petabytes**, **millions of partitions**
    - Scan planning fits on a single node
    - Advanced filtering enables more use cases
    - Overall performance is better
- Low latency queries are faster for large tables

Concurrency

- Production Flink pipeline writing in 3 AWS regions
- Lift service moving data into a single region
- Merge service compacting small files

Usability

- Rollback is popular
- Metadata tables
    - Track down the version a job read
    - Find the process that wrote a bad version

Future Work

- Spark vectorization for faster bulk reads
  - Presto vectorization already done

- Row-level delete encodings
  - MERGE INTO
  - ID equality predicates

Thank you!
Questions?

NETFLIX

Ryan Blue
rblue@netflix.com