databricks

Introduction to Streaming

LECTURE

Streaming from Delta Lake

Streaming from Delta Lake covers how Delta tables can be loaded as streaming sources, with transactional logs tracking new data, support for append-only operations, configurable micro-batch rates, and stateful streaming for tasks like deduplication, aggregations, and stream joins

Streaming from Delta Lake

Using a Delta table as a streaming source

- Each committed version represents new data to stream. Delta Lake transactions logs identify the version's new data files
- Structured Streaming assumes append-only sources. Any non-append changes to a Delta table causes queries streaming from that table to throw exceptions.
  - Set `delta.appendOnly = true` to prevent non-append modifications to a table.
  - Use Delta Lake change data feed to propagate arbitrary change events to downstream consumers (discussed later in this course).

When you load a Delta table as a stream source and use it in a streaming query, the query processes all of the data present in the table as well as any new data that arrives after the stream is started.

You can load both paths and tables as a stream.

- Each committed version represents new data to stream. Delta Lake transactions logs identify the version's new data files
- SS assumes append-only sources. Any non-append changes to a Delta table causes queries streaming from that table to throw exceptions.
  - Set table property delta.appendOnly = true to prevent non-append modifications to a table.
  - Use Delta Lake **Change Data Feed** to propagate arbitrary change events to downstream consumers (discussed later in this course).

Streaming from Delta Lake

Using a Delta table as a streaming source

- You can limit the input rate for micro-batches by setting DataStreamReader options:
  - maxFilesPerTrigger: Maximum files read per micro-batch (default 1,000)
  - maxBytesPerTrigger: Soft limit to amount of data read per micro-batch (no default)
  - Note: Lakeflow Spark Declarative Pipelines auto-tune options for rate limiting, so you should avoid setting these options explicitly for your pipelines.

You can limit the input rate for micro-batches by setting DataStreamReader options:

- maxFilesPerTrigger: Maximum files read per micro-batch (default 1,000)
- maxBytesPerTrigger: Soft limit to amount of data read per micro-batch (no default)
- Note: Lakeflow Spark Declarative Pipelines auto-tune options for rate limiting, so you should avoid setting these options explicitly for your pipelines.

Streaming to Delta Lake

Using a Delta table as a streaming sink

- Each micro-batch written to the Delta table is committed as a new version.
- Delta Lake supports both append and complete output modes.
  - Append is most common.
  - Complete replaces the entire table with each micro-batch. It can be used for streaming queries that perform arbitrary aggregations on streaming data.

As soon as each micro-batch is written, it is committed. This means the data has been successfully ingested and a new micro-batch can proceed.

As mentioned previously, you have a choice of Output modes. When doing Stateful streaming, typically will use Output mode = Complete.
This replaces the entire table with current micro-batch along with contents from previous micro-batches.
Examples of Stateful streaming include:

- Dropping duplicate rows
- Aggregations
- Stream - Stream joins

Thank you for completing this lesson and continuing your journey to develop your skills with us.