databricks

Introduction to Streaming

LECTURE

Aggregations, Time Windows, Watermarks

In this lecture, you will explore the differences between stateless and stateful stream processing, work with examples of stateful streaming operations like group-by and counting, and understand how time-based concepts such as aggregations, time windows, and watermarks are used to analyze and manage streaming data in real time.

Types of Stream Processing

Stateless vs. Stateful processing

- **Stateless**
  - Typically trivial transformations. The way records are handled do not depend on previously seen records.
  - Example: Data Ingest (map-only), simple dimensional joins
- **Stateful**
  - Previously seen records can influence new records
  - Example: Aggregations over time, Fraud/Anomaly Detection

There are two types of stream processing: stateless and stateful. Stateless processing is typically used for trivial transformations.
Records that are handled don't depend on previously seen records. An example of this is ingesting data and then joining it to a dimensional table.
We can think of this in the same way we would think of the total number of cars passing by a checkpoint, where we're joining that data onto the freeway we're on.

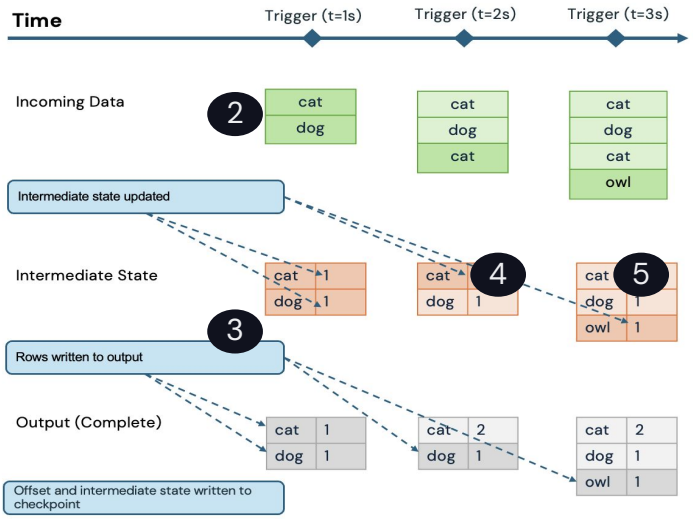
In stateful processing, previously seen records can influence new record aggregations over time. This is particularly useful in fraud and anomaly detection.

An analogy we can use to conceptualize this is the average speed of red cars that pass a checkpoint.

Stateful Stream

```
spark
.readStream
.<source info>
.groupBy(animal)
.count()
.writeStream
.mode("complete")
.<sink info>
.trigger("1s")
.start()
```

1



Let's review a basic example of a Stateful Stream:

1. Here we have an example of a readstream where a GroupBy Animal and a count is applied.
2. We can see in the first trigger with incoming data for two rows: one for a cat and a dog
3. In this same operation, the Intermediate State, the count is generated, in this case would be 1 cat and 1 dog and consequently writing as a complete mode.
4. For the next data arrival, another cat arrives and its intermediate state its updated with cat 2 and dog 1.
5. In the final data arrival, an owl arrives and now its 2 cats, 1 dog and 1 owl.

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