SDS Assignment 4

Siddharth Menon

November 2025

1 System Architecture - Spark

The streaming data pipeline consists of three main components: a data generator, a Kafka cluster, and a Spark Structured Streaming processor.

The data generator continuously produces ad events, which are pushed to the Kafka topic `ad-events-2`. The Spark processor reads the events from this topic, parses the incoming JSON data, and assigns event-time timestamps with watermarks to handle out-of-order events. It then performs 10-second tumbling window aggregations to compute the number of views and clicks for each `campaign_id`.

Finally, the aggregated results, including CTR and the maximum Kafka timestamp per window, are written back to the Kafka topic `results` for downstream consumption.

The flow is as follows:

A dataframe is first created from the raw JSON data which applies a watermark of 1 second. It then splits it into 2 dataframes for aggregating view and click events in 10 second windows (view_counts and click_counts).

A final dataframe is then created using view_counts and click_counts to compile their data and calculate CTR for the incoming data.

2 System Architecture - Flink

The Flink streaming data pipeline also consists of three main components: a data generator, a Kafka cluster, and a Flink SQL processor.

The data generator continuously produces ad events, which are pushed to the Kafka topic `ad-events-2`. The Flink SQL processor reads the events from this topic by defining a source table in Flink with the appropriate schema and a 1-second watermark on the event timestamp (`event_ts`) to handle out-of-order events.

A 10-second tumbling window aggregation is then performed using Flink SQL, computing the number of clicks and views per `campaign_id`. The Click-Through Rate (CTR) is calculated per window as:

Finally, the aggregated results, including CTR and the maximum production timestamp per window, are written to the Kafka topic `results`.

The flow is as follows:

A source table is created in Flink that reads the raw JSON events from Kafka and converts the event time into a timestamp column. Flink SQL then computes windowed aggregates directly over this table, calculating clicks, views, and CTR for each `campaign_id` in 10-second tumbling windows. The results are inserted into the sink Kafka table `results` for downstream consumption.