Read from Kafka topic using multiple Consumers

CoderInBoots

Questions in mind

How to scale consumers ?

- Need to read messages in parallel using multiple consumers

How to tackle the following problems while using multiple consumers ?

- Will the consumers read duplicate messages ?
- How the order of messages are maintained while reading it from multiple consumers ?

Partitions & Consumer Groups

Partitions

- a storage unit that divides a topic log into multiple logs and distributes them across a Kafka cluster
- Partitions allow multiple consumers to read from different partitions at the same time, which enables Kafka to parallelize data processing
- Messages are stored in the order they were produced within a partition. Messages with the same partition key are routed to the same partition, ensuring they are processed in the correct order.
- Each partition can have multiple replicas spread across different brokers, which guarantees fault tolerance and data redundancy.
- The number of partitions and the strategy used to assign messages to them play a crucial role in optimizing Kafka's performance.

Consumer Groups

- A consumer group in Apache Kafka is a collection of consumers that work together to process events from a topic in parallel
- Consumers are assigned a subset of partitions from a topic, and the group rebalances partitions when members join or leave
- The number of partitions in a topic limits the number of consumers that can consume messages at any given time. If there are more consumers than partitions, the extra consumers will be idle.
- Multiple consumer groups can read from the same topic, but each group maintains its own offsets and receives messages independently.

Key points to understand

If a message key is provided

Kafka uses a hash-based algorithm to map the key to a partition. This ensures that messages with the same key are always in the same partition. The default Kafka partitioner uses the murmur2 algorithm to hash keys

If no message key is provided

Kafka uses a round-robin approach to assign messages to partitions. This strategy distributes messages evenly across all available partitions. It's often used when messaging order isn't important

Consumer Group

a collection of consumers that work together to process events from a topic in parallel.

Partition distribution

The group's partitions are divided among the consumers. When a consumer group has one consumer, it receives all messages from all partitions. When there are more consumers than partitions, the extra consumers are idle and don't receive messages.

Multiple consumer groups

Multiple consumer groups can read from the same topic, but each group maintains its own set of offsets and receives messages independently.

Scalability

Consumer groups enable consumers to read from many events at once, which is important for scalability