Kafka Fundamentals

An Introduction to Apache Kafka for Java Developers

Luis Castillo

30-04-2023

Agenda

1. Overview of Kafka
2. Basic Concepts of Kafka
3. Kafka APIs for Java

Section 1

Overview

Core concepts

- **Producer**: Application that sends messages to Kafka
- **Consumer**: Application that reads messages from Kafka
- **Topic**: A stream of records, categorized by name
- **Partition**: A single, ordered, immutable sequence of records in a topic
- **Offset**: Unique identifier for each record within a partition

Architecture and Components

- **Broker**: Kafka server that stores and manages messages
- **Zookeeper**: Coordination service for Kafka cluster
- **Schema Registry**: Stores and manages Avro schema versions

APIs for Java

- **Producer API**: Send messages to Kafka
- **Consumer API**: Read messages from Kafka
- **Streams API**: Process streams of records in real-time
- **Connect API**: Integrate Kafka with other systems

Section 2

Kafka fundamentals

Producer

- Sends messages to Kafka topics
- Can choose the partition for each message
- Handles **serialization** of messages

Consumer

- Reads messages from Kafka topics
- Commits the **offset** of the last consumed message
- Handles **deserialization** of messages

Offset

- A unique identifier for each record within a partition
- Maintained by Kafka to track the position of a consumer in a partition
- Consumers commit the offset of the last consumed message to remember their position

Partitioning

- Topics are divided into **partitions**
- Each partition is an ordered, immutable sequence of records
- Partitions enable parallelism and provide fault tolerance

Consumer Groups

- Group of consumers working together to read messages from a topic
- Each consumer in a group reads messages from a unique partition
- Enables parallelism and load balancing

Rebalancing

- Process of redistributing partition ownership among consumers in a consumer group
- Occurs when consumers are added or removed, or when partitions are added or removed

Delivery Semantics

Defines the guarantee of message delivery in Kafka

Semantics	Description	Characteristics
At Least Once	Ensures messages are delivered at least once	Possibility of duplicate messages
At Most Once	Ensures messages are delivered at most once	Possibility of message loss
Exactly Once	Ensures messages are delivered exactly once	No duplicates, no message loss

Section 3

Kafka APIs

Key Feature of Producer API

- **Asynchronous** message sending
- **Batching** of messages to optimize throughput
- **Serialization** support for various data formats

Key Features of Consumer API

- **Offset** management and committing
- **Deserialization** support for various data formats
- **Rebalancing** for consumer groups

Key Features of Streams API

- **Stateful** stream processing
- **Windowing** support for time-based operations
- **Joining** streams to create complex processing topologies

Key Features of Connect API

- **Source Connectors**: Import data from external systems into Kafka
- **Sink Connectors**: Export data from Kafka to external systems
- **Configuration-based** integration with minimal coding

References I

1. “What is apache kafka?” n.d.

2. “Apache kafka: An introduction,” n.d.

3. “Introduction to apache kafka,” n.d.

4. “What is kafka,” n.d.

5. “Welcome to kafka! We’re glad you’re here,” n.d.

6. “Fundamentals for apache kafka,” n.d.

7. “Kafka internals fundamentals,” n.d.

8. “Event streaming platform,” n.d.

References II

9. “Creating your first apache kafka producer application,” n.d.

10. “KafkaProducer JavaDoc,” n.d.

11. “How to implement kafka producer,” n.d.

12. “Creating your first apache kafka producer application with confluent,” n.d.

13. “Kafka-python documentation,” n.d.