Apache Kafka

History, Pub/Sub messaging, queue systems, Kafka architecture, producers, consumers, topics,ISR

Sairam Elangovan

History

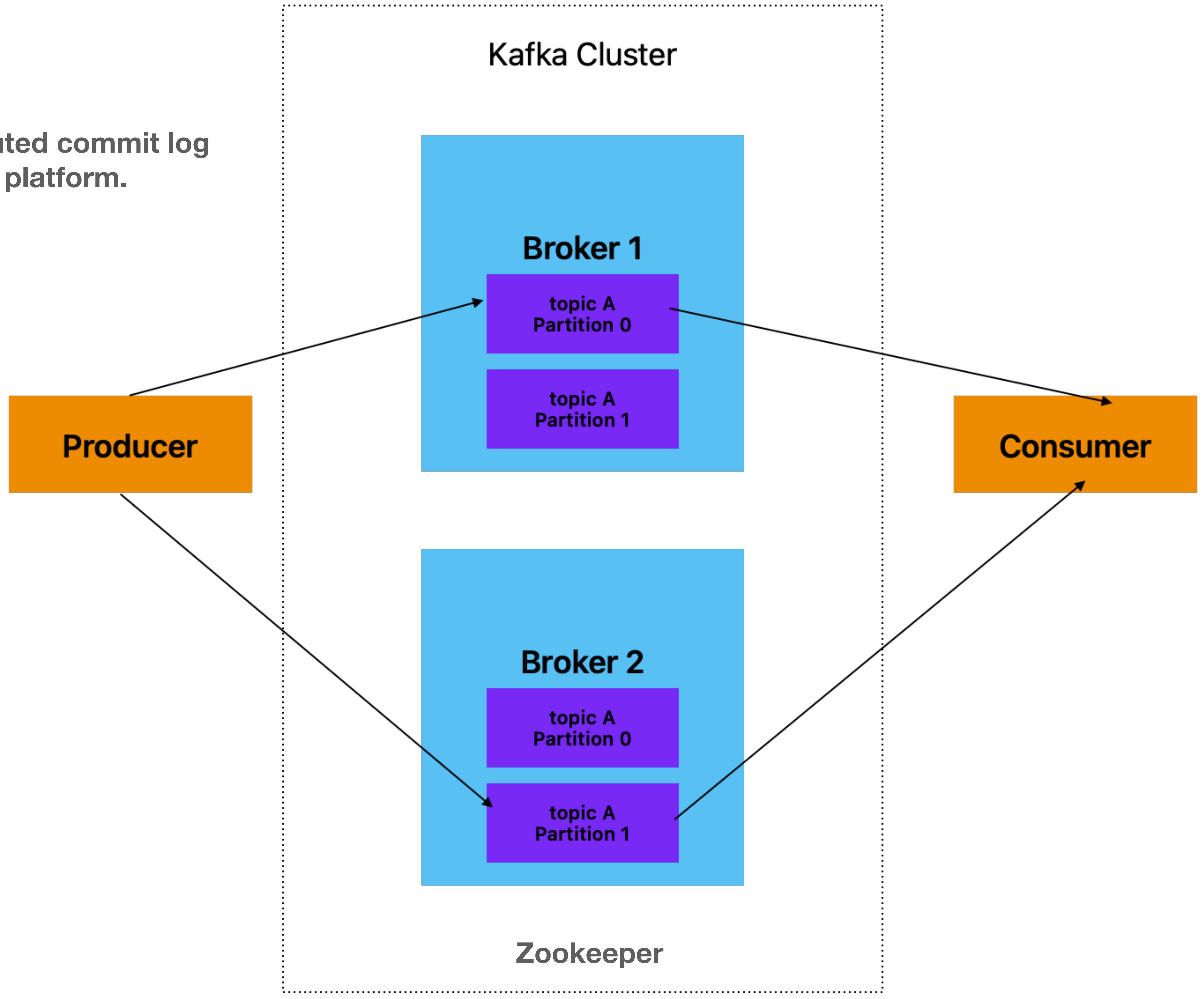
- LinkedIn’s system for collecting system and application metrics had performance issues and the data collection was inconsistent across different systems.
- The systems would break continuously due to the ever changing schemas and tracking was based on hourly batching, i.e. it could not be used in real time.

Primary goals of Kafka

- Decoupled messaging system (push-pull model using producers and consumers)
- Data persistence (multiple consumers consuming the same data for different purposes)
- High Throughput
- Horizontal scaling of system

Kafka Architecture

Kafka is often described as distributed commit log or as a distributed streaming platform.



Kafka installation

- Tarball local development installation
- Installation via package managers (apt, yum, rpm etc)
- Docker containers

Kafka installation through docker compose

- git clone https://github.com/confluentinc/cp-all-in-one
- cd cp-all-in-one
- cd cp-all-in-one/
- docker-compose up -d zookeeper broker control-center

Tarball local development installation

- $ curl -O http://packages.confluent.io/archive/7.3/confluent-7.3.1.tar.gz
- $ tar xzf confluent-7.3.1.tar.gz
- Configuring .bashrc file in linux for enabling custom env variable cods (Optional):
    - $ export CONFLUENT_HOME=${HOME}/confluent-7.3.1 \ && echo "export CONFLUENT_HOME=$CONFLUENT_HOME" >> ~/.bashrc
    - $ echo "export PATH=$CONFLUENT_HOME/bin:${PATH}" >> ~/.bashrc
    - $ ~/confluent-7.3.1/bin/confluent completion bash | sudo tee /etc/bash_completion.d/confluent \ && echo "source /etc/bash_completion.d/confluent" >> ~/.bashrc \ && source ~/.bashrc

Kafka installation via package managers

- $ sudo apt-get install openjdk-11-jre-headless
- $ wget -qO - https://packages.confluent.io/deb/7.1/archive.key | sudo apt-key add -
- $ sudo add-apt-repository \ "deb [arch=amd64] https://packages.confluent.io/deb/7.1 stable main" && \ sudo apt-get update
- $ sudo apt-get install -y \ confluent-platform \ confluent-security
- $ sudo systemctl start confluent-zookeeper (systemctl enable for running services even if your linux server is rebooted)
- $ sudo systemctl start confluent-server

Zookeeper's role in kafka

- ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.
- All of these kinds of services are used in some form or another by distributed applications.
- ZooKeeper is used for
  - Controller election
  - Cluster membership
  - Topic configuration
  - Access control lists
  - Quotas
  - Maintaining consumer offsets (post 0.9v consumer offsets are stored under __consumer_offsets topic)

Kafka brokers

- A computer instance or container running Kafka process
- Manage partitions
- Handle read and write requests
- Manage replication of partitions

Topic creation

- Topics are logs that hold messages or events in a logical order.
- ISR- All reads/writes for a specific partition happens through 'Leader' of the partition and 'Follower' get in-sync with 'Leader' for updates.
- $ bin/kafka-topics.sh --create --topic new-topic --bootstrap-server localhost:9092
- $ bin/kafka-topics.sh --describe --topic quickstart-events --bootstrap-server localhost:9092
- bin/kafka-topics.sh --list --zookeeper localhost:2181

Producers

- A producer partitioner maps each message to a topic partition, and the producer sends a produce request to the leader of that partition
- The partitioners shipped with Kafka guarantee that all messages with the same non-empty key will be sent to the same partition.
- $ bin/kafka-console-producer.sh --topic quickstart-events --bootstrap-server localhost:9092

Consumers

- Consumers read data from Kafka topics.
- Consumers can be standalone or a part of consumer group
- Offset management (from-beginning, latest and none)
- $ bin/kafka-console-consumer.sh --topic quickstart-events --from-beginning --bootstrap-server localhost:9092

Docker exec commands

- docker exec broker \ kafka-topics --bootstrap-server broker:9092 \ --create \ --topic new-topic
- docker exec --interactive --tty broker kafka-console-producer --bootstrap-server broker:9092 --topic new-topic
- docker exec --interactive --tty broker kafka-console-consumer --bootstrap-server broker:9092 --topic new-topic —from-beginning
- docker exec broker kafka-topics --bootstrap-server broker:9092 —list