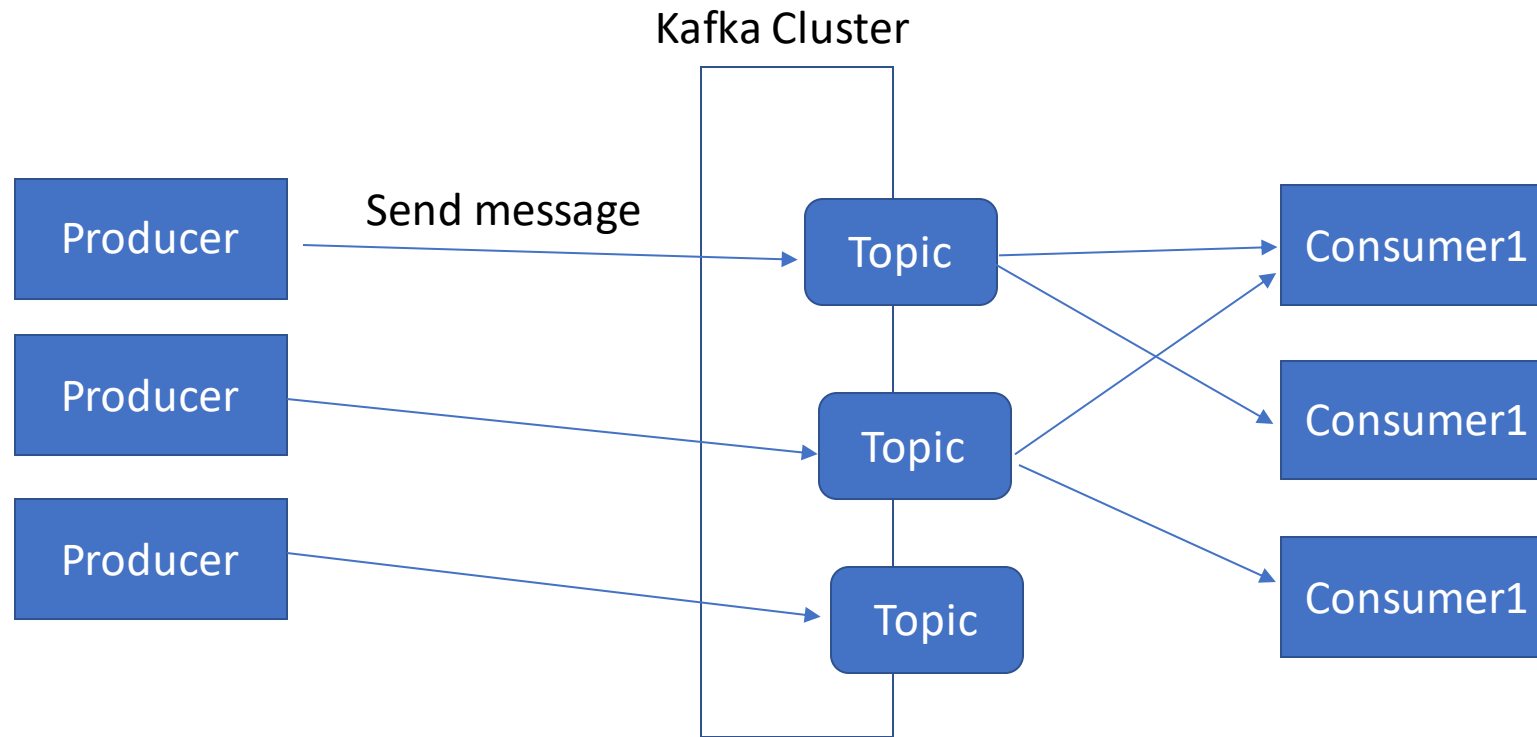


Kafka as Dumb Pipe

-Pratik Das

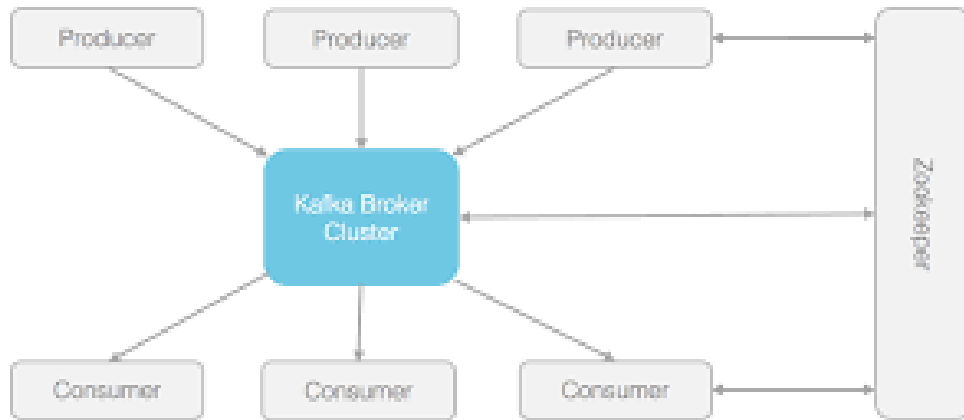
Kafka-Publish-Subscribe



Capabilities

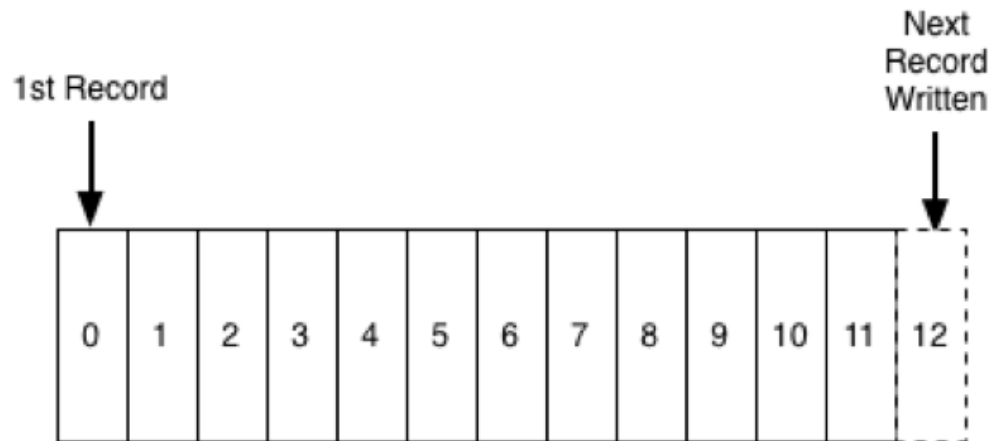
- Distributed and redundant System
- Event ledger
- Clustering is core
- Durability and ordering guarantee

Producer-Broker-Consumer



- Broker – Node in cluster
- Producer writes records to a broker
- Consumer reads records from broker
- Leader election for cluster distribution

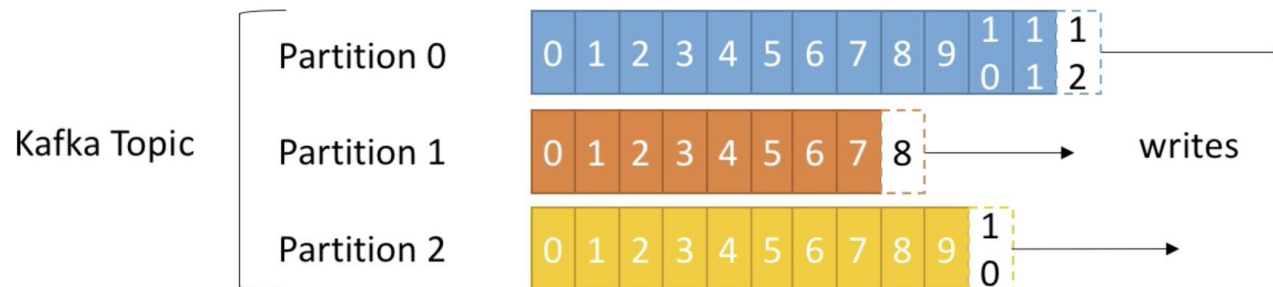
Producer Message



- Key, value, timestamp
- Immutable
- Append only
- Persisted to disk not memory

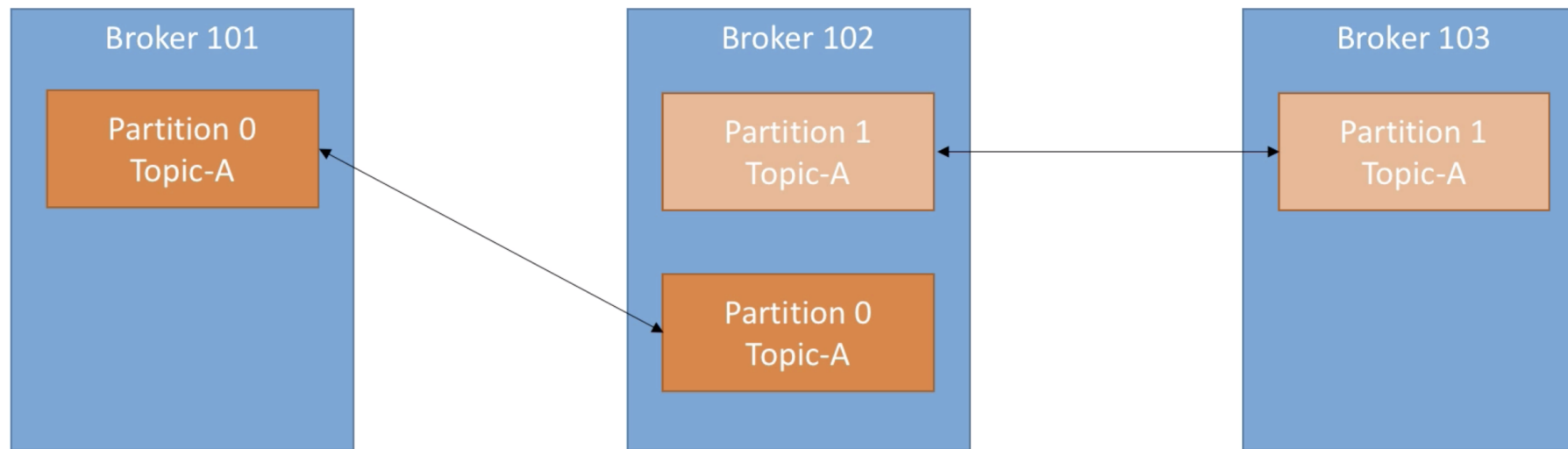
Topic, Partitions and offsets

- Topic is a particular stream of data
 - Similar to a table in a database
 - Any number of topics can be created
 - Identified by its name
- A topic is split into Partitions
 - Each partition is ordered. Order is guaranteed only within a partition
 - Each message within a partition gets an incremental id called an Offset
 - Data written to a partition is immutable and deleted after a period of time
 - Data is assigned randomly to a partition unless a key is provided



Kafka Cluster - Brokers

- A Kafka Cluster is composed of multiple brokers (servers)
- The partitions in a topic reside in one of the brokers
- Each partition is replicated in other brokers depending on its Replication factor. Replication is managed using Leader-follower algorithm.
- Clients connect to a cluster by connecting to 1 broker (bootstrap broker)



Partition 0 is replicated
between broker 101 and 102

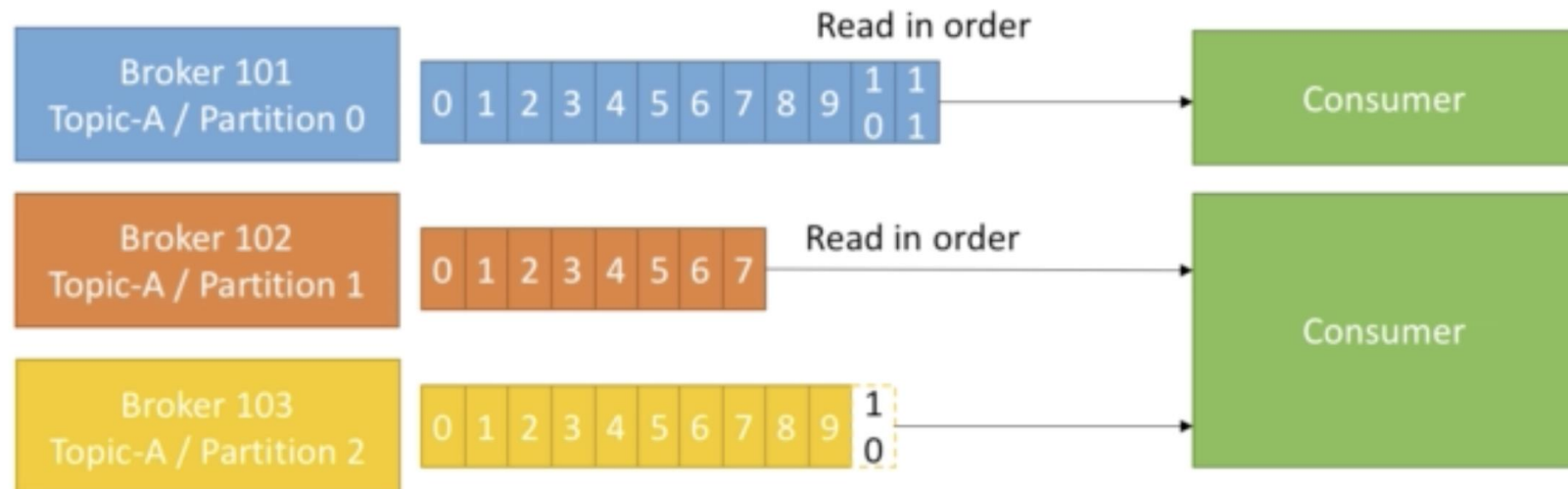
Partition 1 is replicated
between broker 102 and 103

Producers – Get data into Kafka

- Producers send data to topic
- Producers choose to receive acknowledgements-
 - Acks=0 (producer will not wait for ack)
 - Acks=1 (producer will wait for leader ack)
 - Acks=2 (producer will wait for ack from leader and all replicas)
- Producer can send a message with a key. Messages sent with same key are written to same partition.
- If key is not sent, partition is selected on a round-robin basis.
- Ordering within partition is guaranteed using key hashing

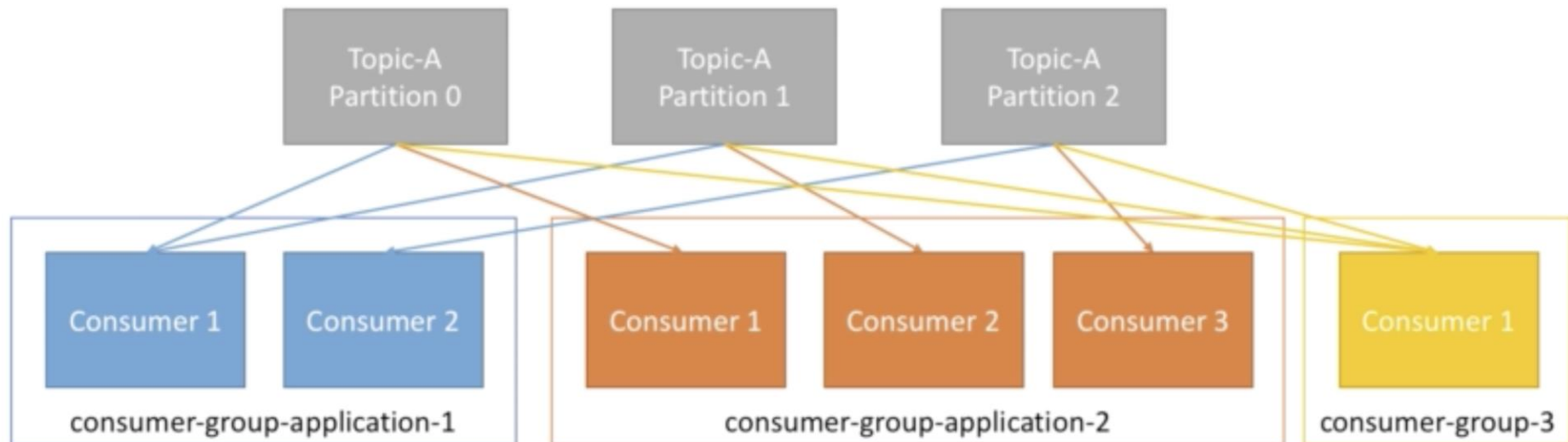
Consumers

- Consumers read data from topic
- Consumer know which data to read from
- Data is read in order within each partition
- Consumer can also read from multiple partitions

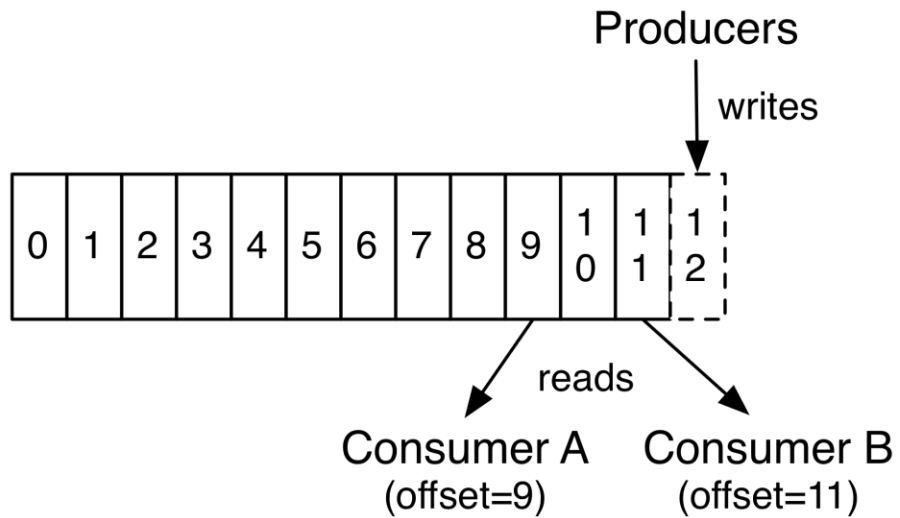


Consumer Groups

- Consumer read data in consumer groups
- Each consumer within a group reads from exclusive partitions
- If number of consumers are more than number of partitions, some consumers will be inactive.



Consumer Offsets

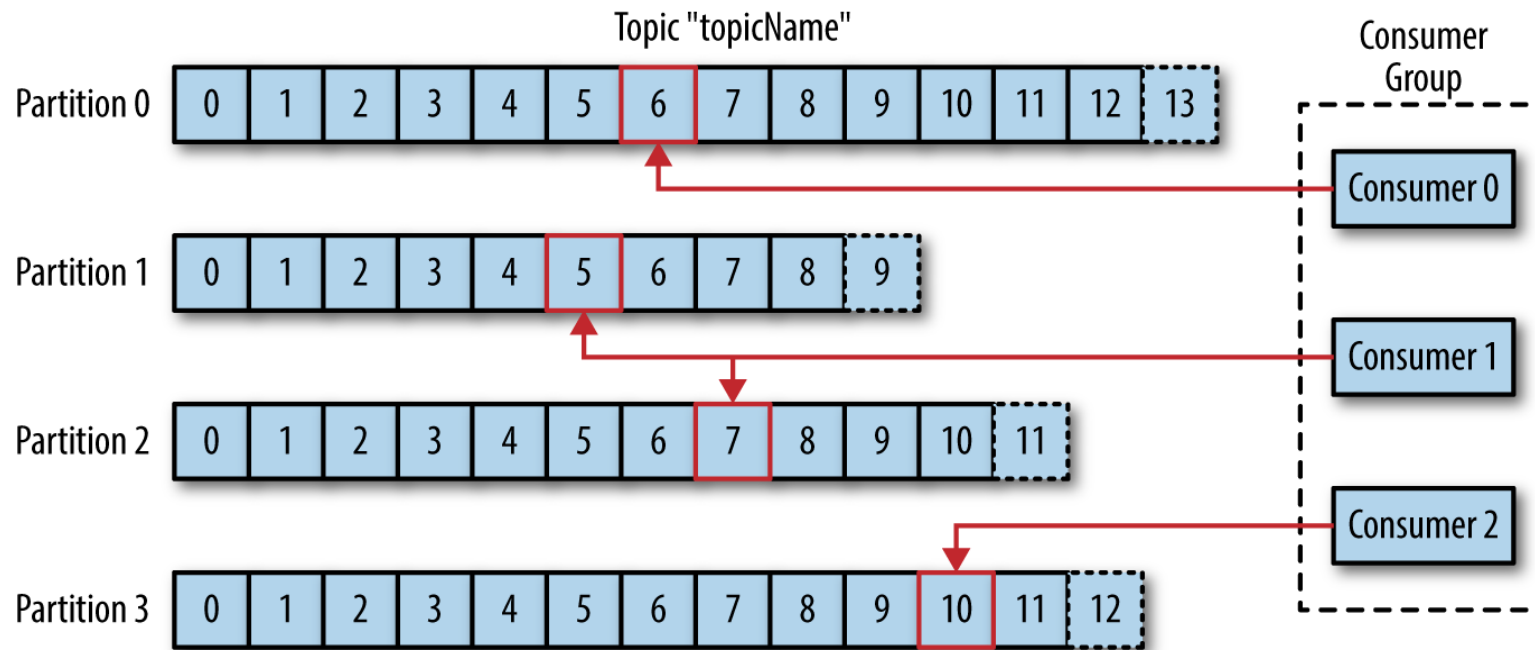


- Offset is a Unique sequential Identifier per partition
- Consumer keep track of offsets
- Consumers commits the offset after processing the data read from partition.
- The committed offsets are stored in a Kafka topic

Broker Discovery

- Consumer connect to any 1 broker called bootstrap server
- Each broker knows about all other brokers, topics and partitions

Parallelism using Partitions, offsets and consumer groups



- Writes are made to leader of a partition
- Partition can be manual or based on a key
- Replication factor is for topic
- Auto-rebalancing

Delivery Guarantees

- Producer

- Async – no guarantee
- Committed to leader
- Committed to leader and quorum

- Consumer

- At-least once(Default)-offsets are committed when message is processed.
- At-most once-offsets are committed when message is received
- Exactly once -for kafka to kafka workflows or kafka to external idempotent consumer

Other features

- Log compaction
- Disk not heap
- Pagecache to socket
- Balanced partitions and leaders

Differences with Messaging System

- Ordering
- Horizontal Scaling
- Push

Events(Not Tables) as Single Source of Truth

