بسم الله الرحمن الرحیم

تکنولوژی کامپیوتر

جلسه‌ی بیست و چهارم

ادامه پردازش جریان – کافکا

جلسه گذشته

مفاهیم

- Stream
- Topic
- Producer
- Consumer

Messaging Systems

- Direct messaging from producer to consumer
- Message Broker
  - *Queue-Based Messaging Systems*
  - *Log-Based Messaging Systems*

Handling Multiple consumers

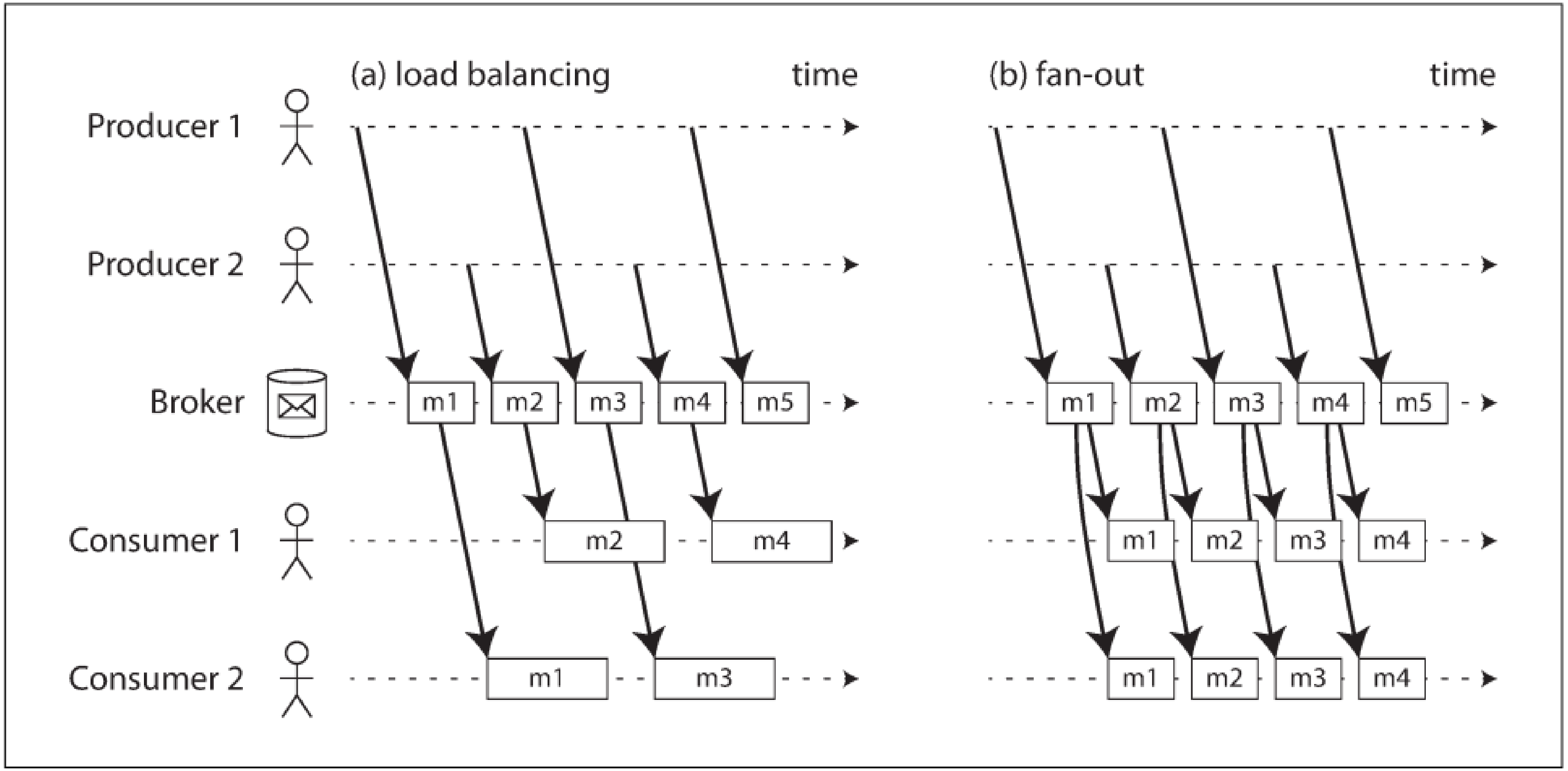


Figure 11-1. (a) Load balancing: sharing the work of consuming a topic among consumers; (b) fan-out: delivering each message to multiple consumers.

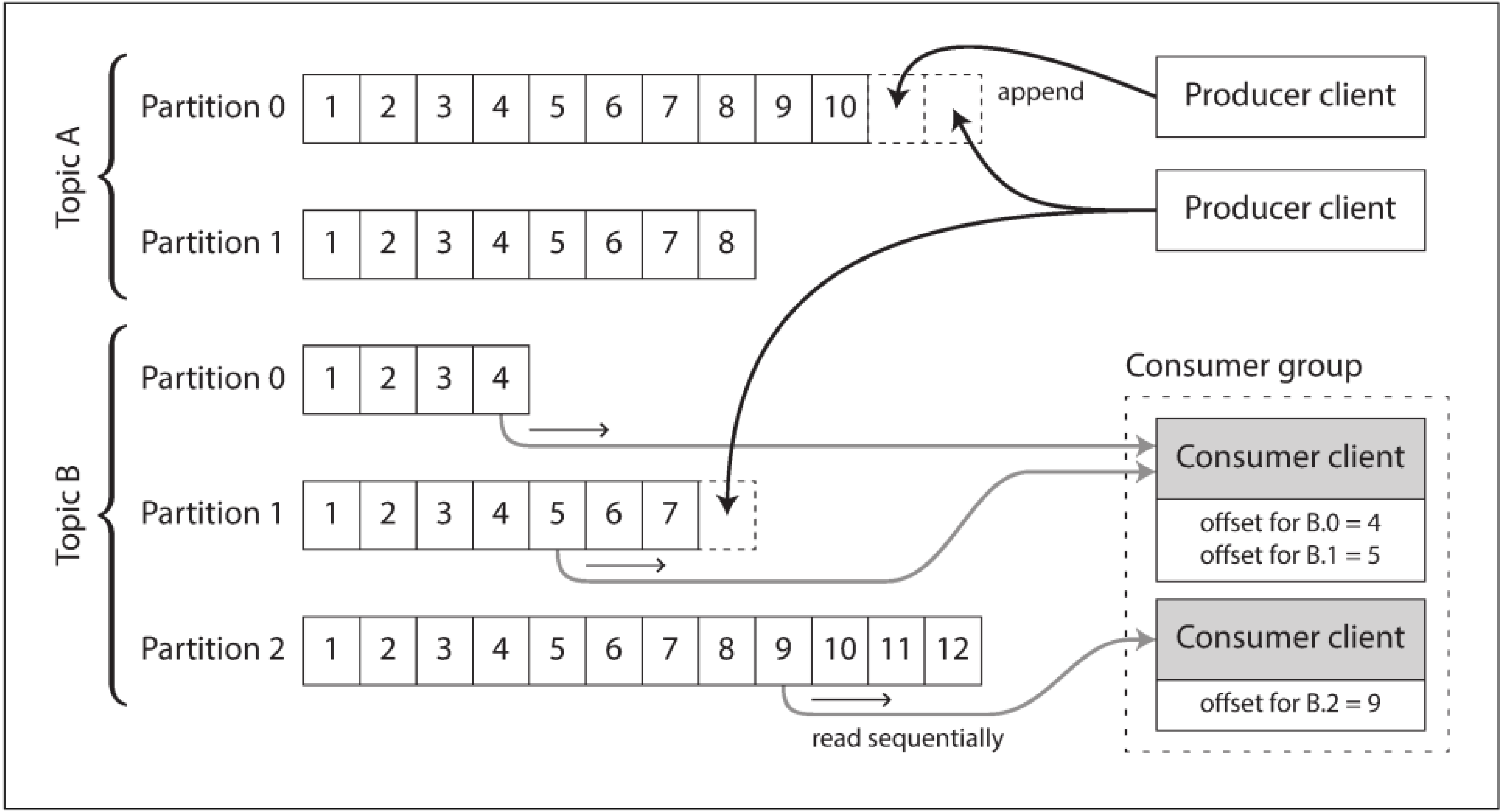


Figure 11-3. Producers send messages by appending them to a topic-partition file, and consumers read these files sequentially.

Implementing change data capture

- Like one leader
- Capture changes
  - *SQL Trigger*
  - *Read logical replication log*

Event sourcing

- A technique in Domain Driven Design community
- Event Sourcing Vs. Change data capture
- Event as source of truth
  - *Then transform it into application state*
- Event vs Commands

جلسه جدید

State, Streams, and Immutability

- Advantages of immutable events
- Limitations of immutability

Uses of Stream Processing

- Monitoring and alerting
- Complex event processing
- Stream analytics
- Maintaining materialized views
- Search on streams
- Message passing and RPC

PROCESSING STREAMS

Options in Stream Processing

- You can take the data in the events and write it to a database, cache, search index, or similar storage system, from where it can then be queried by other clients.
- You can push the events to users in some way, for example by sending email alerts or push notifications
- You can process one or more input streams to produce one or more output streams. Streams may go through a pipeline consisting of several such processing stages before they eventually end up at an output (option 1 or 2).

Reasoning About Time

- the average over the last five minutes.
- the last five minutes

- Event time versus processing time

Reasoning About Time

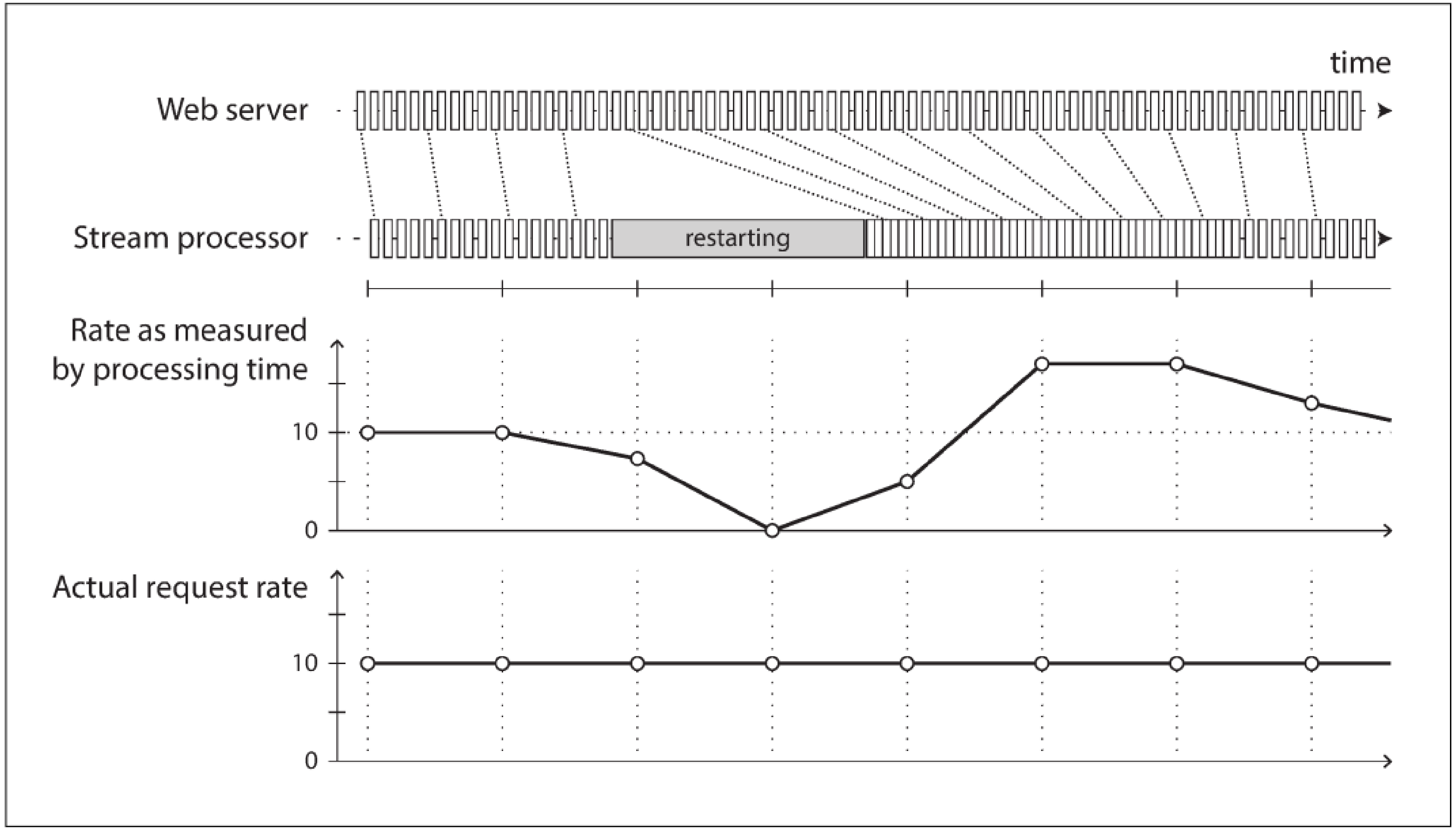


Figure 11-7. Windowing by processing time introduces artifacts due to variations in processing rate.

Calculating events on a given window?

- you can never be sure when you have received all of the events for a particular window, or whether there are some events still to come.

Calculating events on a given window?

- Options:
  - *Ignore the straggler events, as they are probably a small percentage of events in normal circumstances. You can track the number of dropped events as a metric, and alert if you start dropping a significant amount of data.*
  - *Publish a correction, an updated value for the window with stragglers included. You may also need to retract the previous output.*

Whose clock should be used?

- To adjust for incorrect device clocks, one approach is to log three timestamps:
  - *The time at which the event occurred, according to the device clock*
  - *The time at which the event was sent to the server, according to the device clock*
  - *The time at which the event was received by the server, according to the server clock*

Types of window

- Tumbling window
- Hopping window
- Sliding window
- Session window

Stream Joins

- Stream-stream join (window join)
- Stream-table join (stream enrichment)
- Table-table join (materialized view maintenance)

Problem: Time-dependence of joins

Fault Tolerance

- What should we do when streaming job failed?

Fault Tolerance

- Microbatching and checkpointing

Fault Tolerance - External side effects

- Atomic commit and distributed transaction

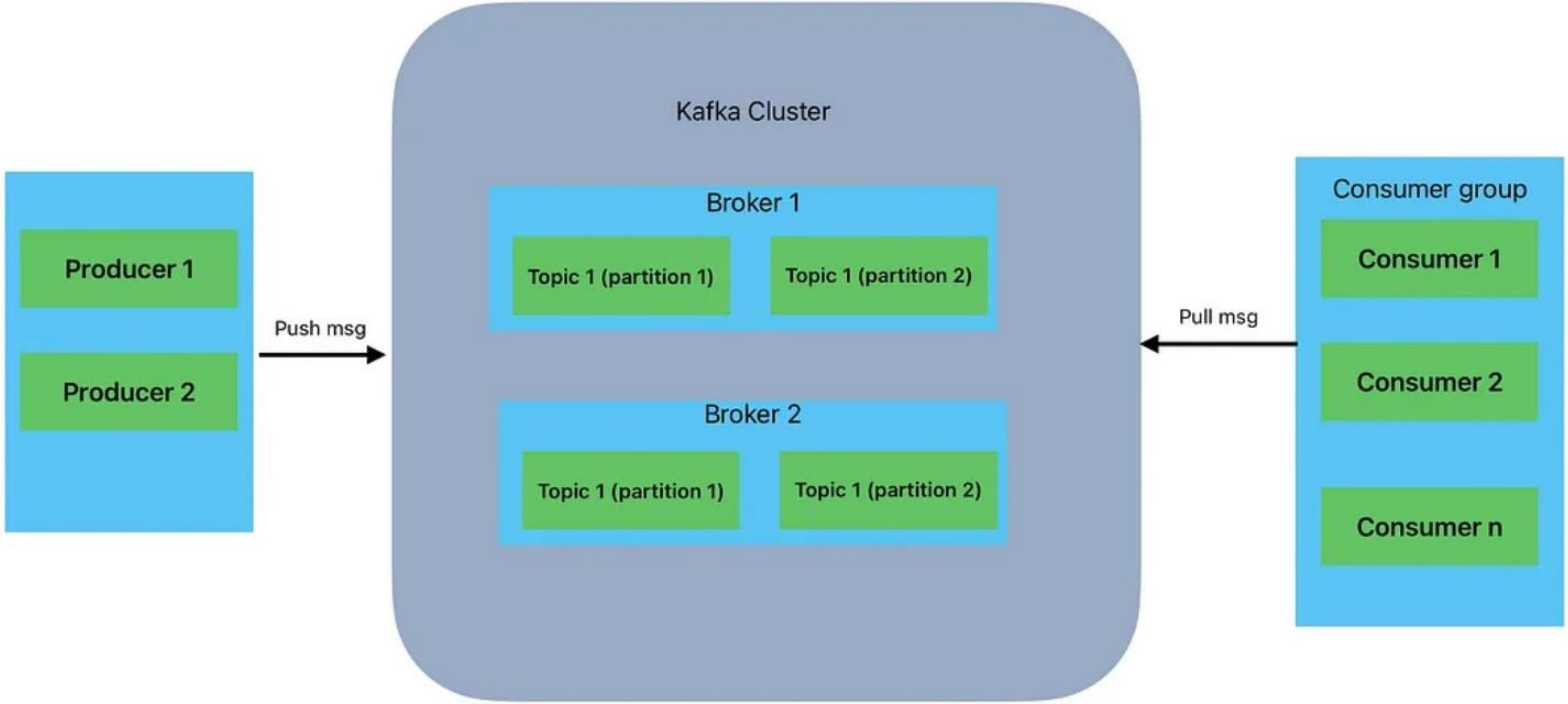
Fault Tolerance - External side effects

- Idempotence

Fault Tolerance

- Problem: Rebuilding state after a failure

APACHE KAFKA


Kafka Cluster
Producer 1
Producer 2
Push msg
Broker 1
Topic 1 (partition 1)
Topic 1 (partition 2)
Broker 2
Topic 1 (partition 1)
Topic 1 (partition 2)
Pull msg
Consumer group
Consumer 1
Consumer 2
Consumer n

Kafka Terminology

- Producer
- Consumer
- Broker
- Controller
- Message
- Topic
- Partition
- Offset

Kafka Broker in Cluster

- Cluster Metadata
- Leader Election
- Scalability

Single Leader Replication

- Leader
- Followers
- ISR (In-Sync Replica)

Data Distribution Across Partitions

- Round-robin partitioning
- Key-based partitioning

Choosing the Right Number of Partitions

Retention

Kafka Producer

- Asynchronous Writes
- Partitioning
- Acks (Acknowledgments)
- Compression

Kafka Producer Idempotence

How Kafka Consumers Work

- Subscribe
- Polling Data
- Process Data
- Commit Offsets
  - *Automatic Offset Commit*
  - *Manual Offset Commit*

Consumer Group

Offset Management

Kafka Transactions

PRACTICE WITH KAFKA