

Apache Kafka

Welcome to the World of Distributed
Messaging Queue

Agenda

- Course Curriculum
- Messaging Queue
- Distributed Messaging Queue
- Apache Kafka
- High Level Architecture
- Broker, Producer, Consumer
- Topic, Partition, Replication
- Real time Kafka use cases
- Kafka in market

COURSE CURRICULUM

- Introduction of Kafka
- Architecture of Kafka – Broker, Producer, Consumer, Topics, Partition, Replication
- Installation of Kafka on single node
- Installation of Kafka on cluster
- Comparison of Kafka with other messaging queue.
- Basic Kafka Operations
- Adding, removing and modifying topics
- Producer overview and its API
- Uniform data distribution between partition
- Consumer overview and its API
- Configuration of consumer, producer and brokers
- Monitoring Kafka using JMX and other administration tools
- Several POCs and project

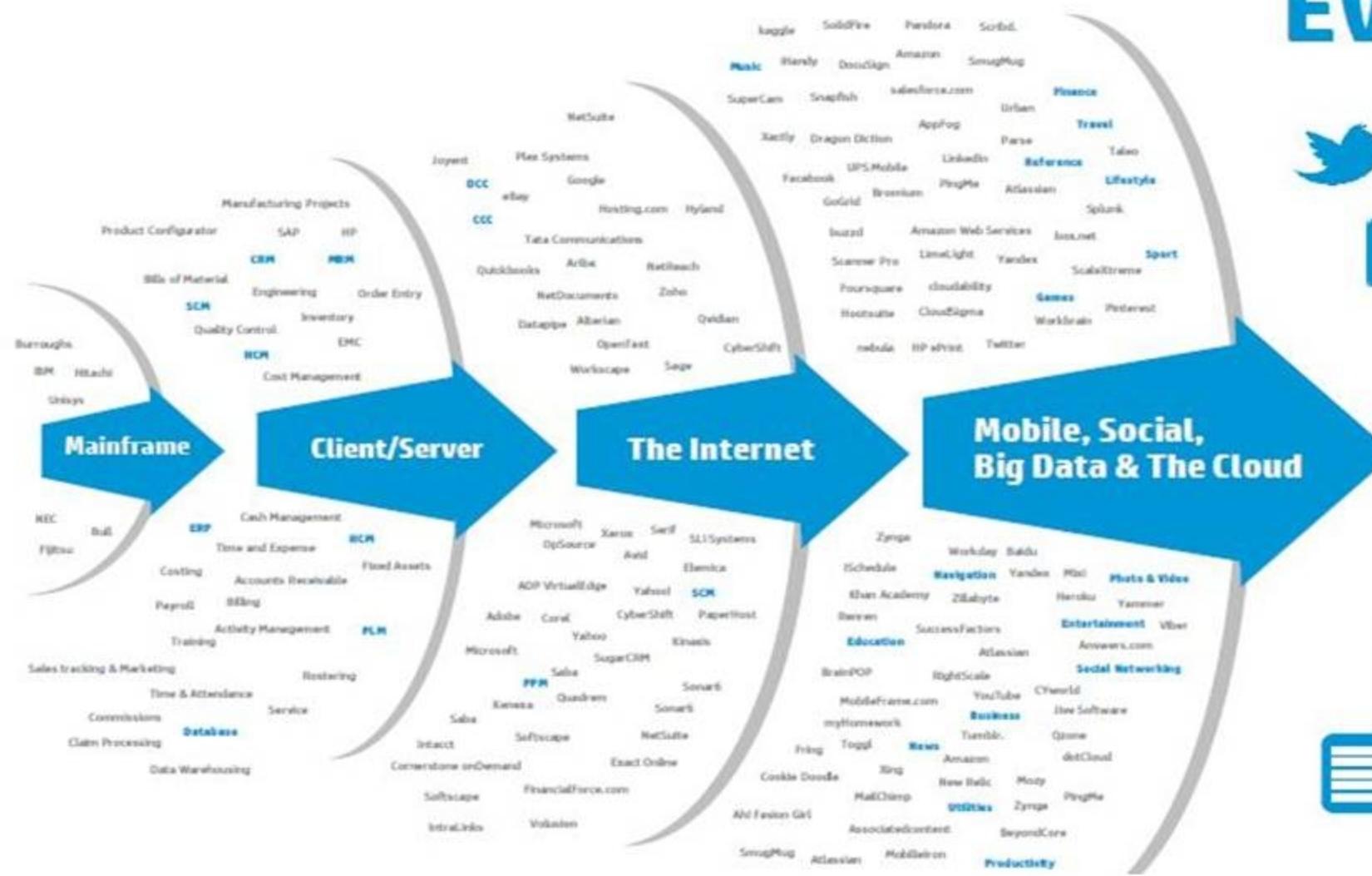


What is Messaging Queue?

- Message queues provide an asynchronous communications protocol.
- Asynchronous communications protocol meaning that the sender and receiver of the message do not need to interact with the message queue at the same time.
- Messages pushed onto the queue are stored until the client retrieves them.
- Message queues have implicit or explicit limits on the size.

Can messaging queue handle the big data?

A new style of IT emerging



Every 60 seconds

-  **98,000+** tweets
 -  **695,000** status updates
 -  **11million** instant messages
 -  **698,445** Google searches
 -  **168 million+** emails sent
 -  **1,820TB** of data created
 -  **217** new mobile web users

Can messaging queue handle the big data?



- No
- We would require a distributed messaging system.

- Features of Distributed Messaging Queue
 - Scalable:
 - Scale to hundreds of node
 - Fault tolerance:
 - System should work even some nodes in cluster goes down
 - System should support replication
 - No single point of failure:
 - System should work even the master node in a cluster goes down.
 - High Read/Write throughput
 - Should handle **million of messages** per second

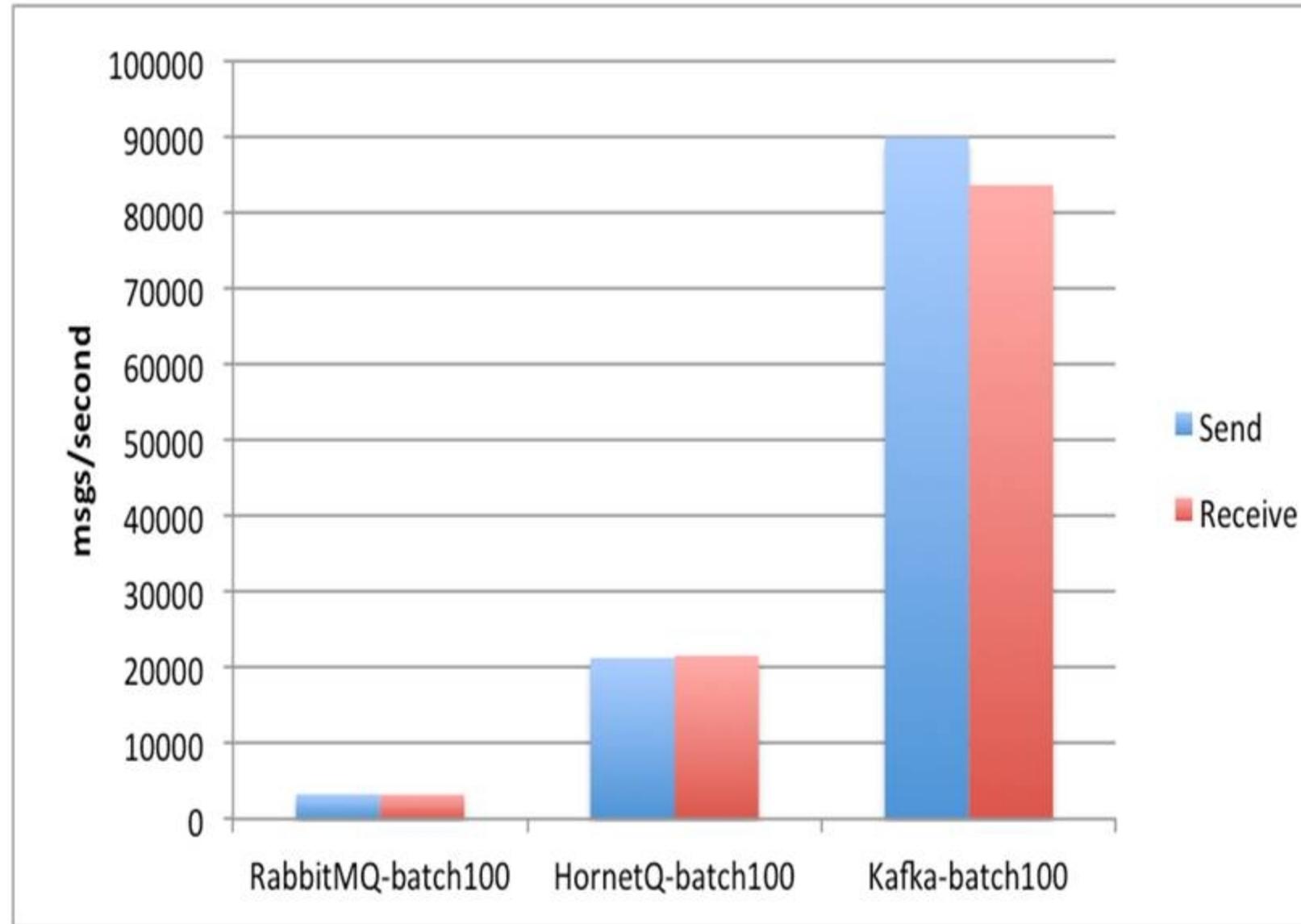
Available Messaging Queue in Market



Why Apache Kafka?

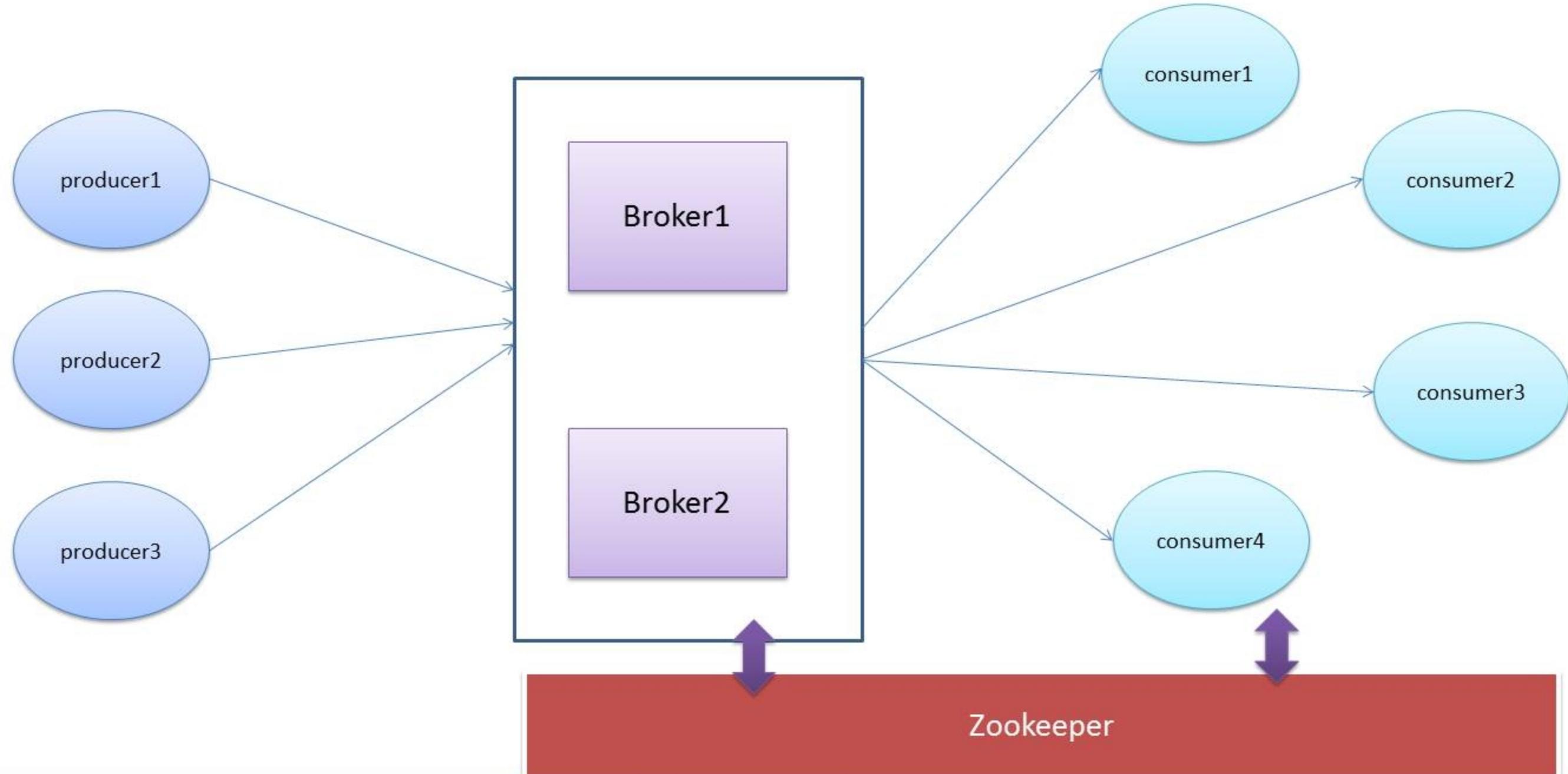


- Open sourced by LinkedIn in 2011.
- Apache 2.0 license
- Scalable
 - It can easily scale to hundreds of nodes by zero time.
- Durable
 - Messages are persisted on disk and replicated within the cluster to prevent data loss.
- Fault Tolerance
 - Support replication
- Write/Read throughput:
 - A single Kafka node can handle hundreds of megabytes of reads and writes per second from thousands of clients.
- No Single point of failure:
 - No master/slave architecture.
 - Same role is assigned to all the nodes in cluster. Guarantee ordering of messages
- Support compression
- Support the data retention



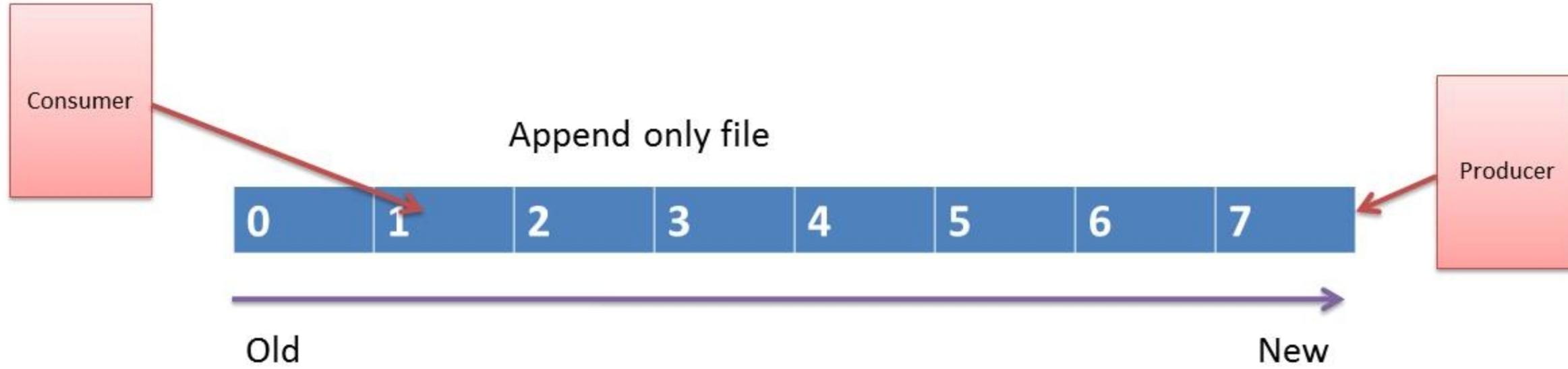
1. **RabbitMQ:** 2K msgs/sec
2. **HornetQ:** 20K msgs/sec
3. **Kafka:** 90K msgs/sec

High Level Architecture



- Broker receive message from producer (Push)
- Consumer read data from Broker (Pull)
- Broker is responsible to persist the messages on disk for some time.
- Broker is very lightweight
 - Just handle the TCP connection
 - Write data into file.
- Broker write the data into **append only log** files.
- Producer perform the sequential write on append only log files.
- Consumer perform the sequential reads.

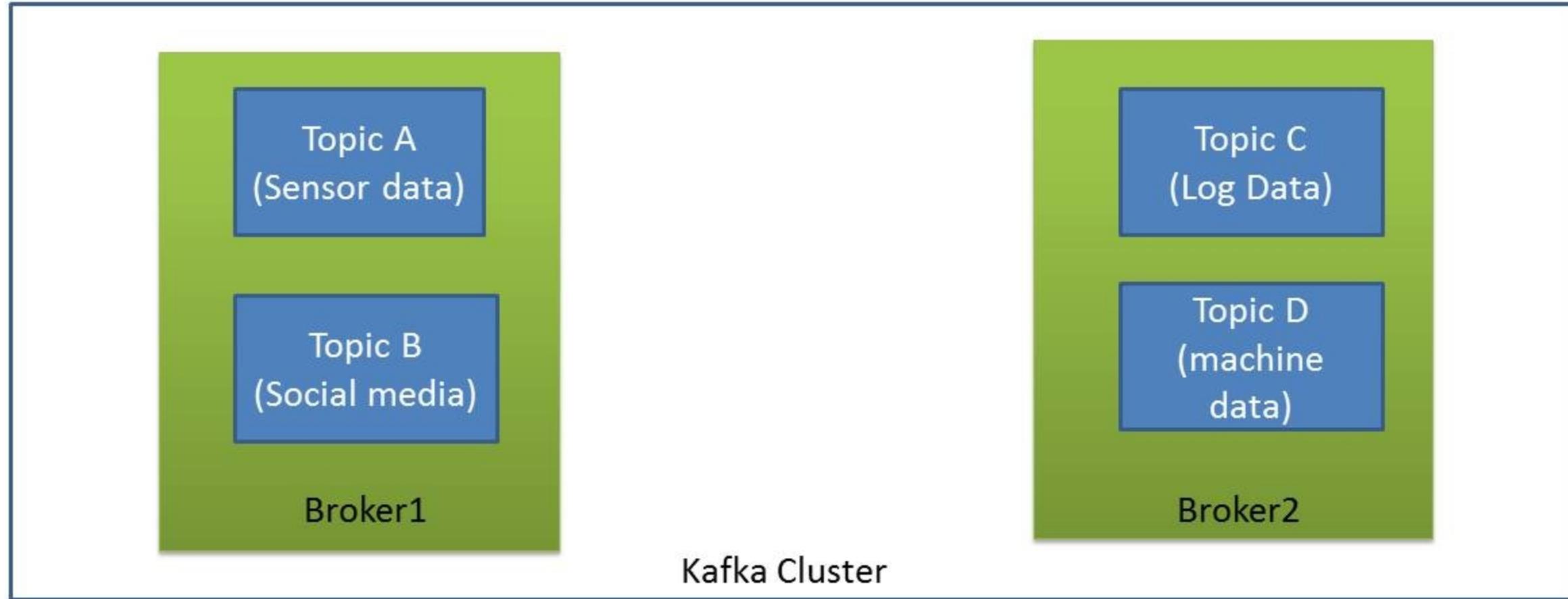
Append Only Log File



How data is stored?

- Data is stored in a topic
- What is topic?
 - Topic is a Queue
 - Different types of message should be insert into different topics.
 - A one broker can handle the multiple topics.
 - A one topic can assign to multiple brokers.

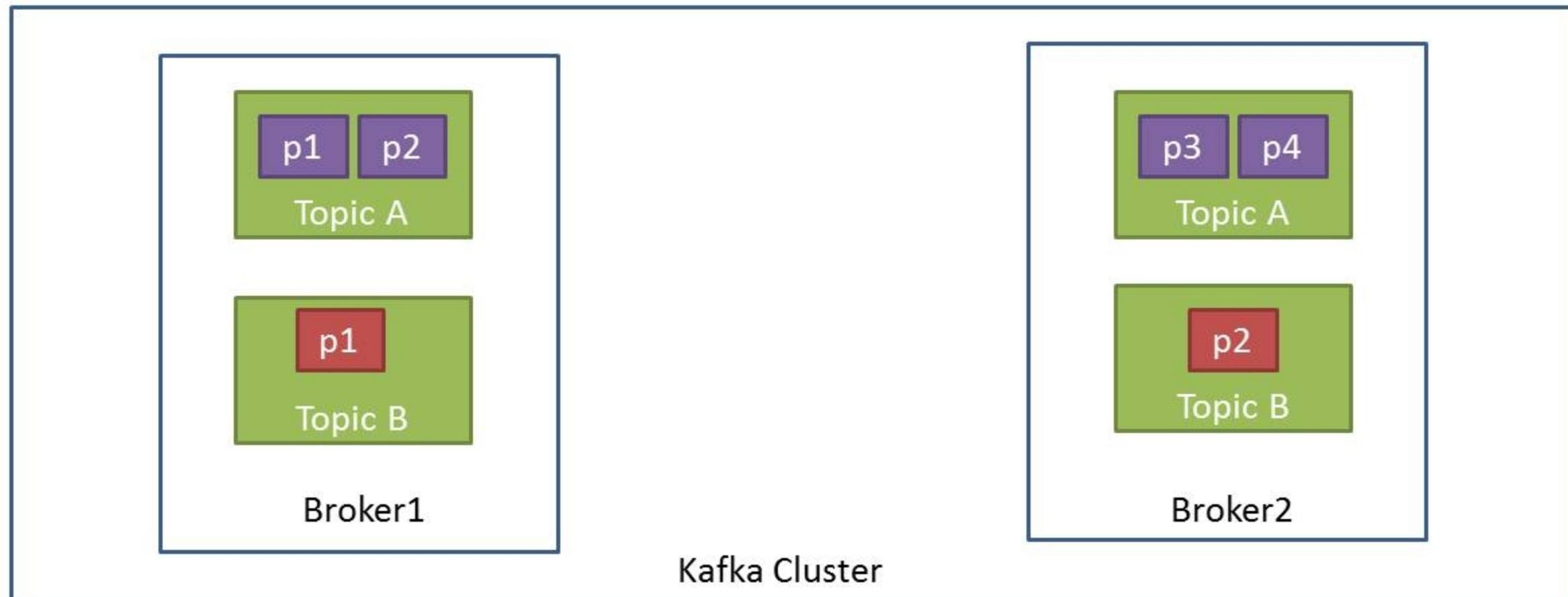
Topic Distribution





What will happen if
topic size reach to
Broker disk size?

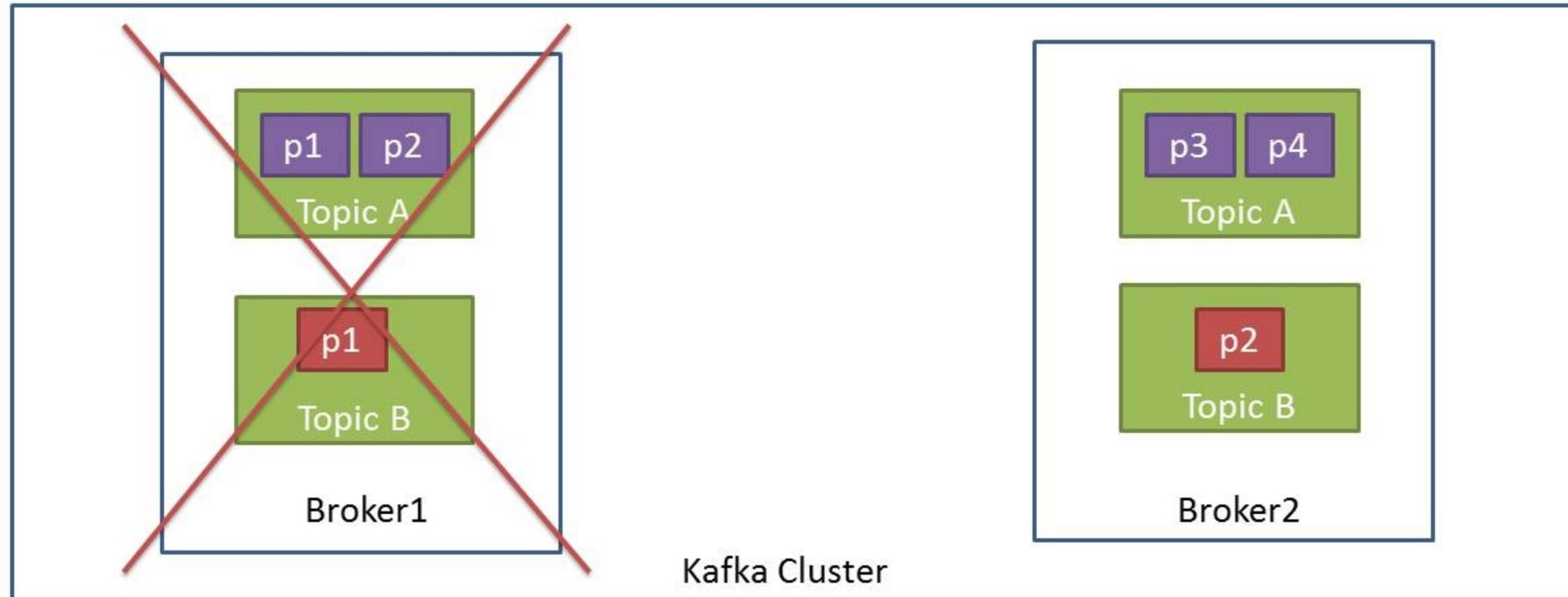
- Partition is the logical distribution of topic.
- A broker can contains multiple partitions of a topic.





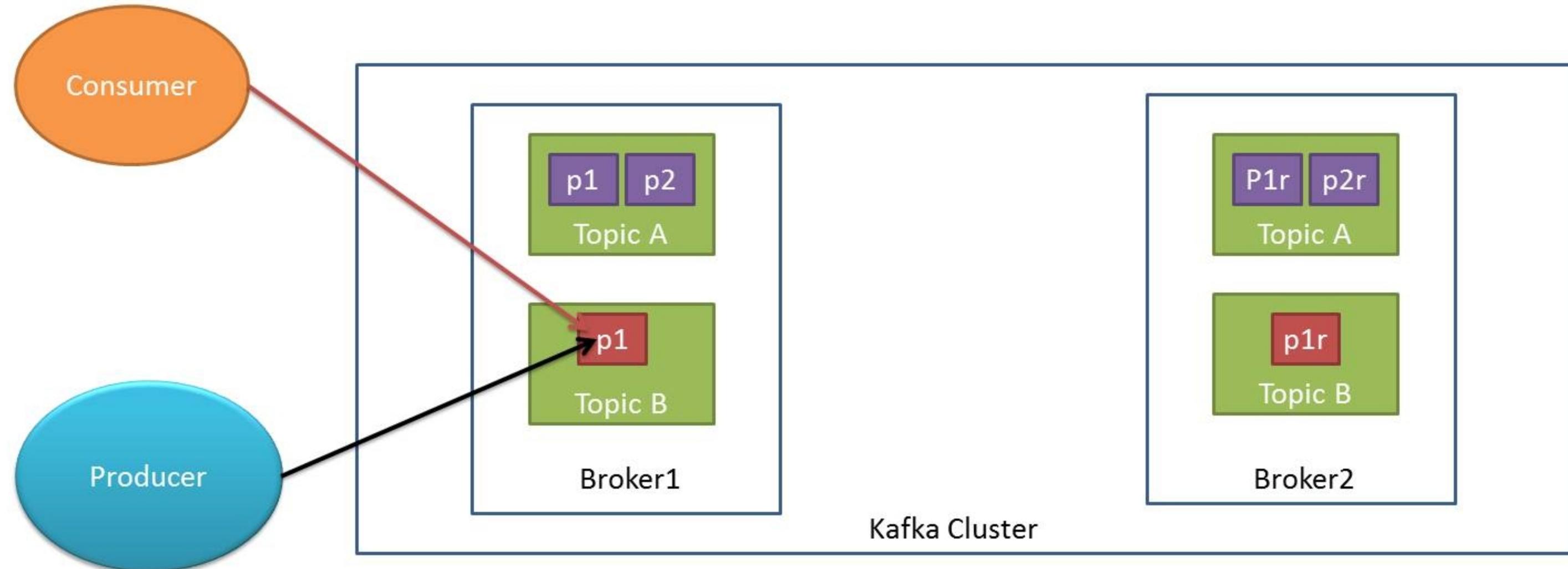
What will happen,
if my broker goes
down ??

Broker machine down

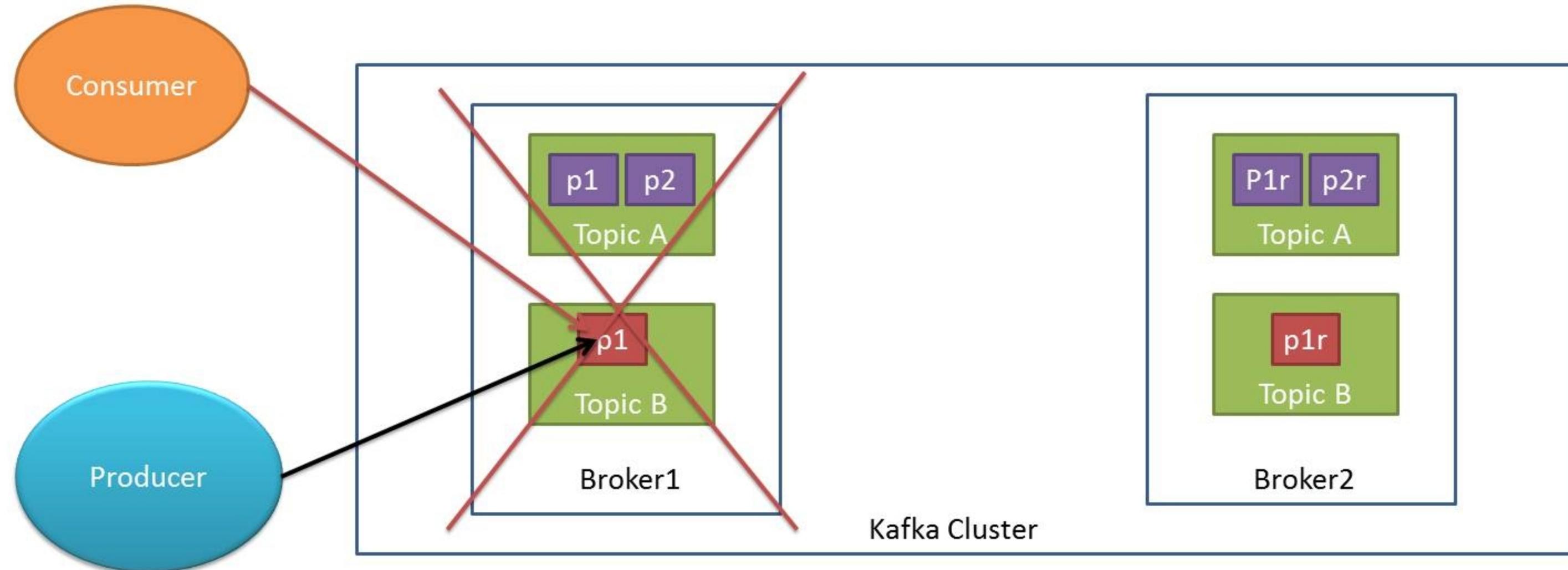


- Partitions of a topic are replicated.
- One partition is the leader of partition(master partition) and replica of same partition is treated as follower.
- All write/read happens on leader partition.
- Replication provides fault tolerance.

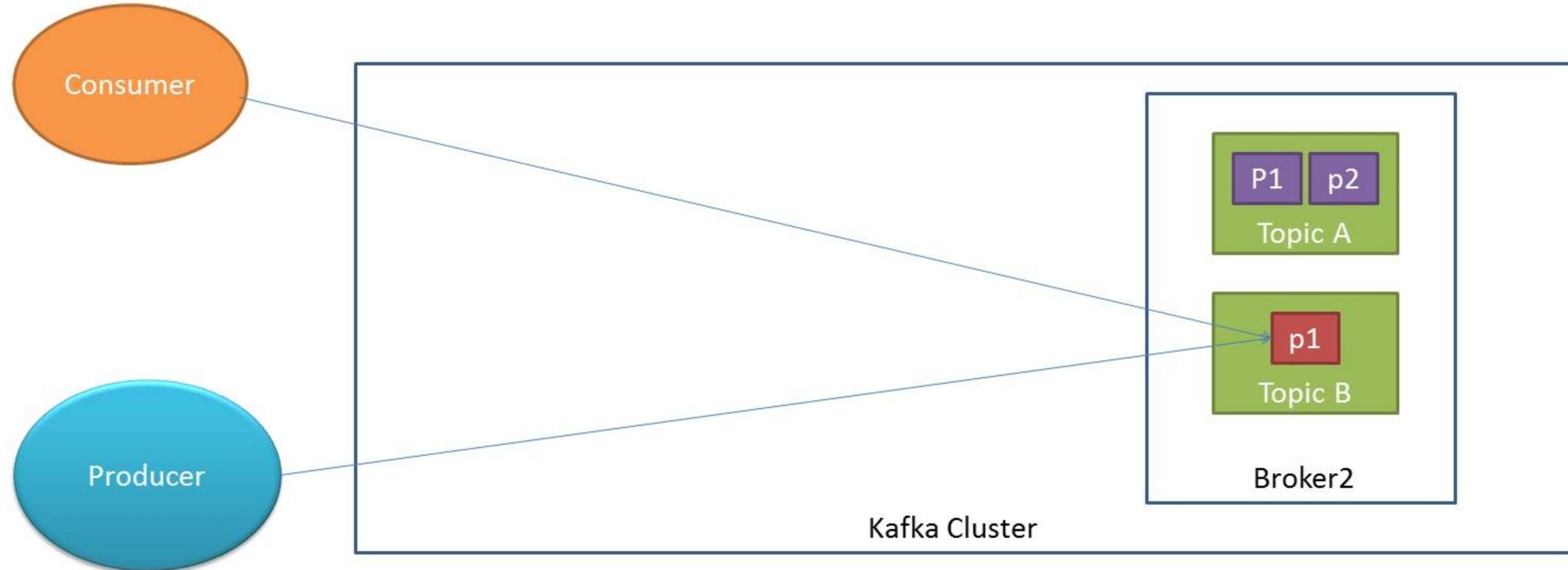
Replication Distribution



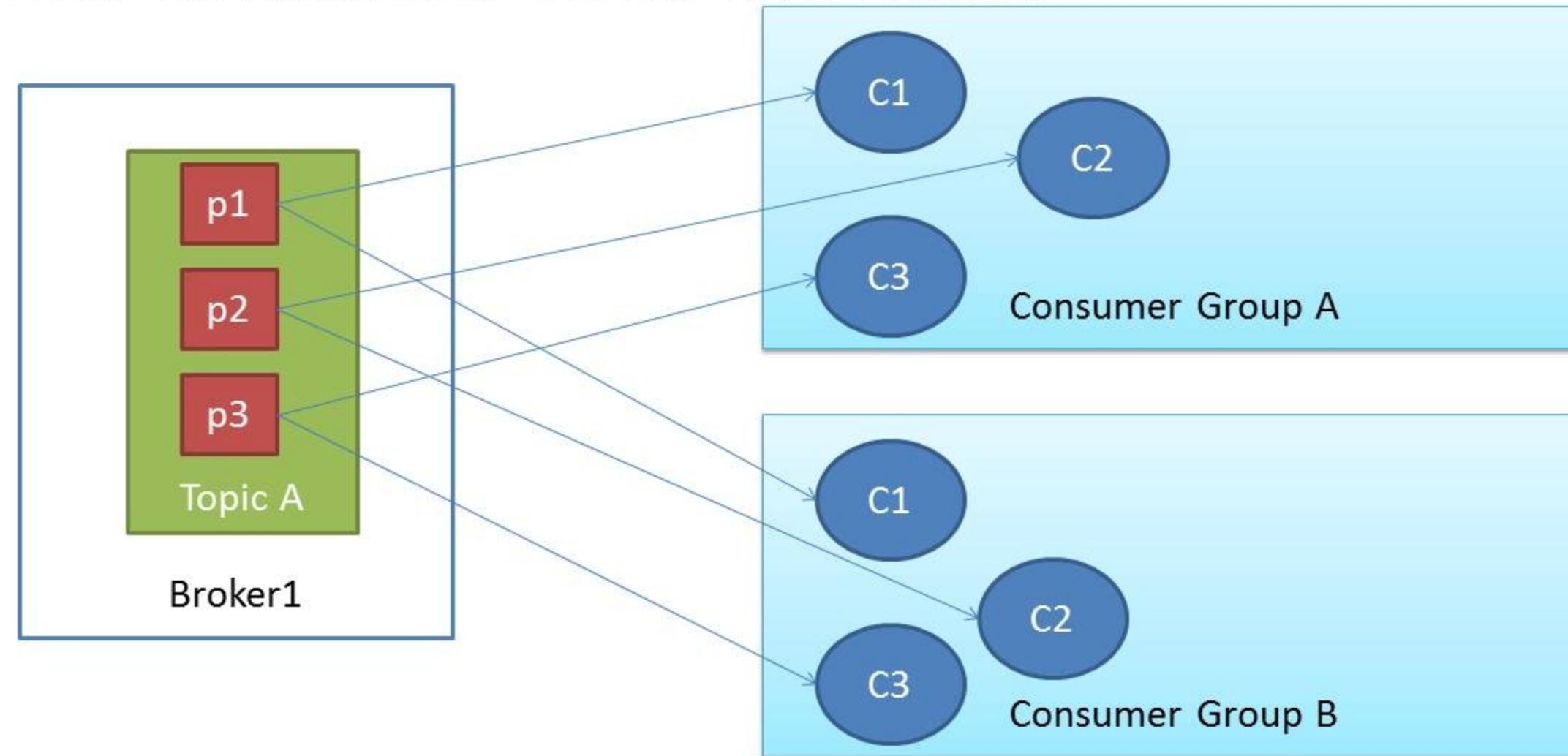
Replication Distribution



Replication Distribution

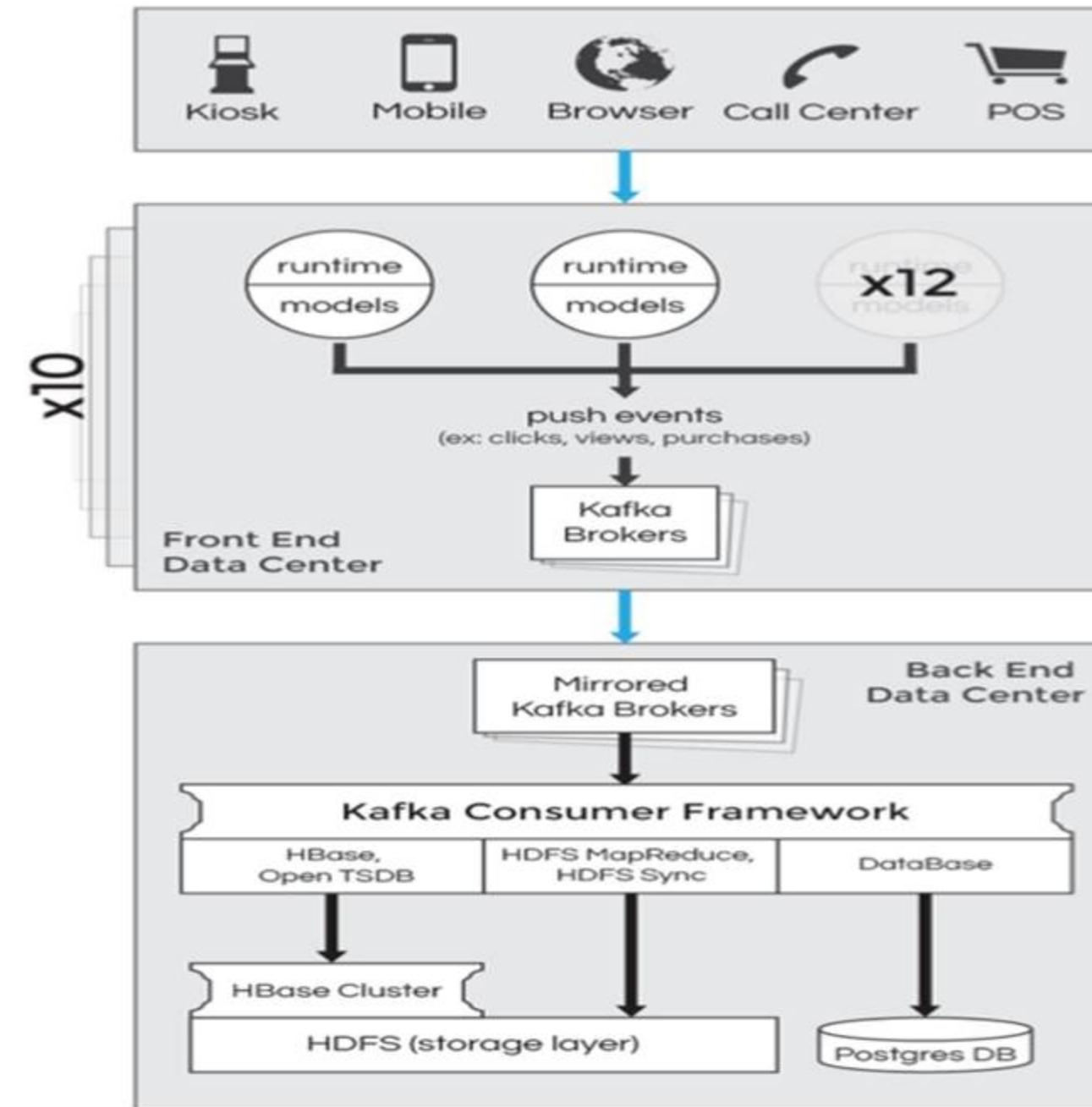


- How consumer read the data?



- **RichRelevance** is a company, based in San Francisco, California, that offers personalized shopping experiences for large retail brands, including Wal-Mart, Sears and others
- They evaluated several technologies including Flume, Active MQ and Kafka.
- Selected Kafka
- Why?
 - They want to decouple the rate at which messages are produced and transferred to the Kafka server from the rate at which the messages are consumed.
 - Another important criterion was fault-tolerance.
 - After evaluating many messaging system, they found that, Kafka has better throughput, built-in partitioning, replication, and fault-tolerance—which makes it an operationally scalable solution for large scale message processing applications.

RichRelevance Real Time use-case



- Loggly is the world's most popular cloud-based log management.
- Requirement:
 - A massive stream of incoming events with bursts reaching 100,000+ events per second and lasting several hours
 - The need for a “no log left behind” policy: Every log has the potential to be the critical one, and our customers can't afford for us to drop a single one

- Selected Kafka
 - **No single point of failure:** Every day, we move terabytes of data through our Kafka cluster without losing a single event. We use age-based retention to purge old data on disks.
 - **Low latency:** 99.99999% of the time our data is coming from disk cache and RAM; only very rarely do we hit the disk.
 - **Performance:** It's crazy good!. One of our consumer groups (eight threads) can process about 200,000 events per second draining from 192 partitions spread across three brokers.
 - **Scalability:** Its ability to increase the partition count per topic and can add new nodes on run time.

Who all are using Kafka?

- LinkedIn:
 - Apache Kafka is used at LinkedIn for activity stream data and operational metrics. This powers various products like LinkedIn Newsfeed, LinkedIn Today in addition to our offline analytics systems like Hadoop
- Twitter:
 - As part of their Storm stream processing infrastructure
- Netflix:
 - Real-time monitoring and event-processing pipeline.

Who all are using Kafka?



sematext

foursquare™

The Cisco logo features a series of seven vertical bars of increasing height from left to right, followed by the word "CISCO" in red capital letters and a small "TM" symbol.

The LinkSmart logo features a blue stylized 'L' icon followed by the text "LinkSmart" in a blue sans-serif font.

The Ooyala logo features a cluster of colored dots (red, green, blue, yellow) followed by the word "OOYALA" in a black sans-serif font with a registered trademark symbol.

- Now a days Kafka has become a prime requirement to solve any big data real time problem.
- Every one require a distributed messaging system to move effectively and efficiently data from source to data store.

Thank You

DataFlair Web Services Pvt Ltd

+91-8451097879

info@data-flair.training

www.data-flair.training