ThoughtWorks®

The Best of Apache Kafka Architecture

Ranganathan Balashanmugam

@ran_than

Helló Budapest

About Me

- ❏ Graduated as Civil Engineer.
- ❏ <dev> 10+ years </dev>
- ❏ <Thoughtworker from="India"/>
- ❏ Organizer of Hyderabad Scalability Meetup with 2000+ members.
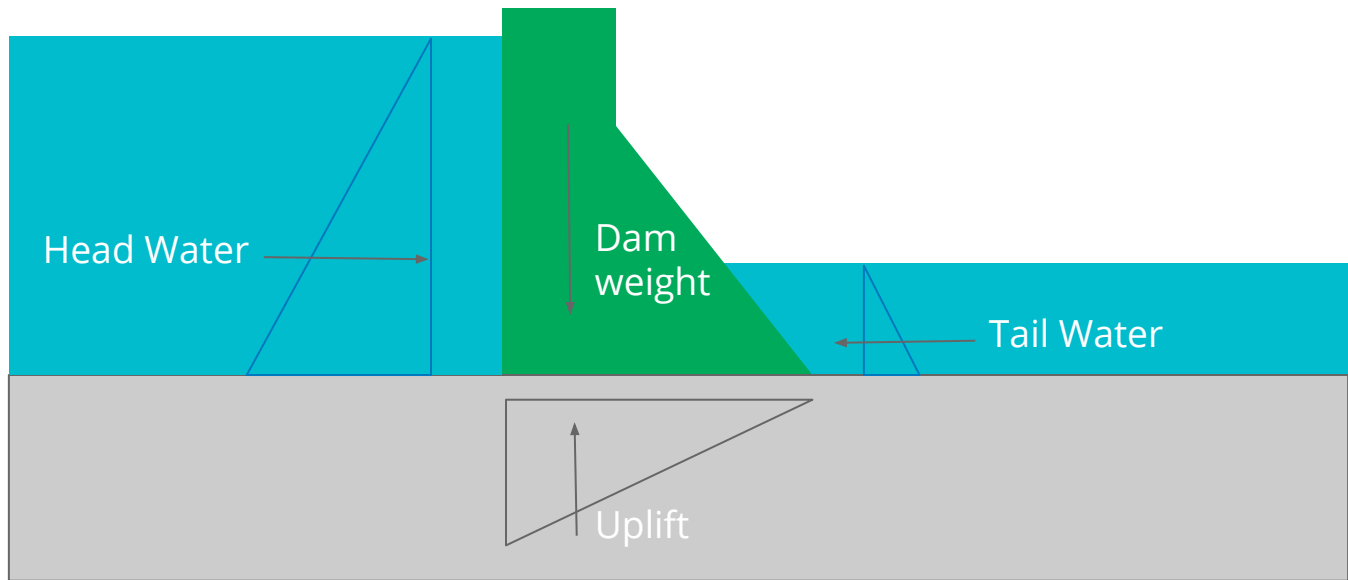
"Form follows function."

- Louis Sullivan

Gravity Dam

Indirasagar Dam, India

Forces on a gravity dam

Head Water

Dam weight

Tail Water

Uplift

kafka

- ❏ publish-subscribe messaging service
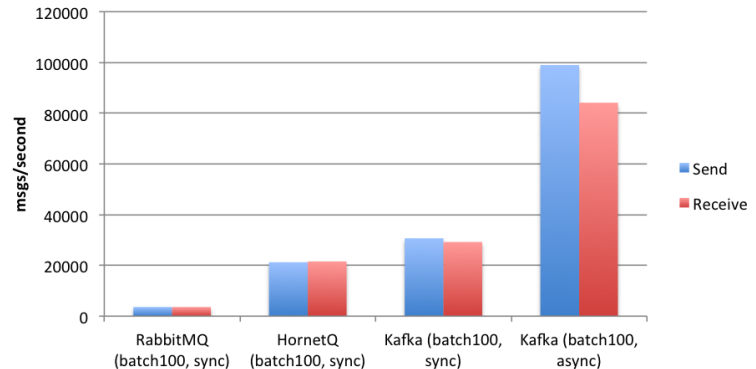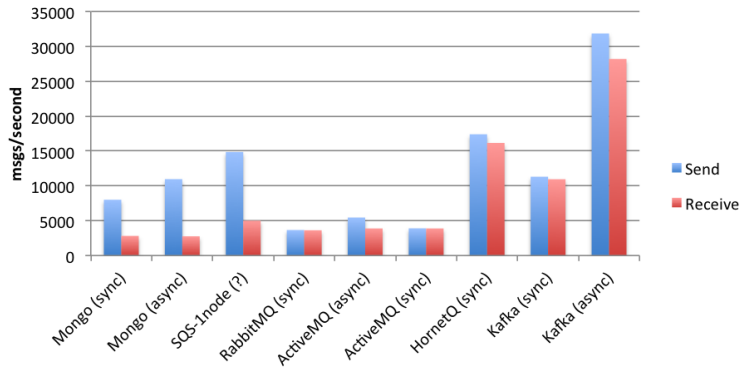- ❏ distributed commit/write-ahead log

"producers produce, consumers consume, in large distributed reliable way -- real time"

Why Kafka?

- ❏ DBs
- ❏ Logs
- ❏ Brokers
- ❏ HDFS

"For highly distributed messages, Kafka stands out."

Kafka Vs _____

Timeline

Open sourced by LinkedIn, as version 0.6

Graduated from Apache

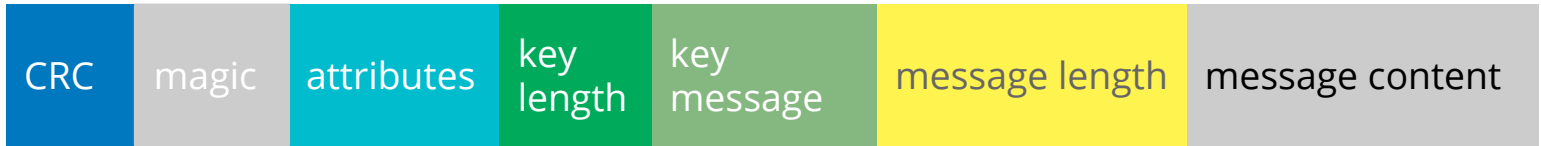Several Engineers who built Kakfa create Confluent
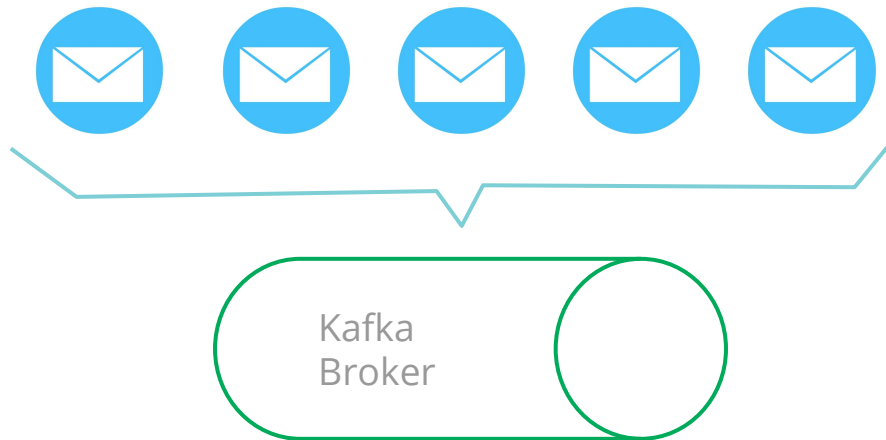
Latest stable - 0.8.2.1

| 2011 | | 2012 | | 2013 | | 2014 | | 2015 |

A Kafka Message

CRC	magic	attributes	key length	key message	message length	message content

kafka.message.Message

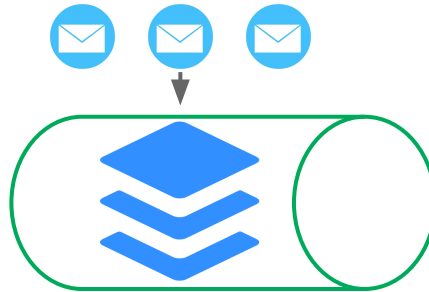Change requested:KAFKA-2511

Producers - push

Request => RequiredAcks Timeout [TopicName [Partition MessageSetSize MessageSet]]

Kafka
Broker

Response => [TopicName [Partition ErrorCode Offset]]
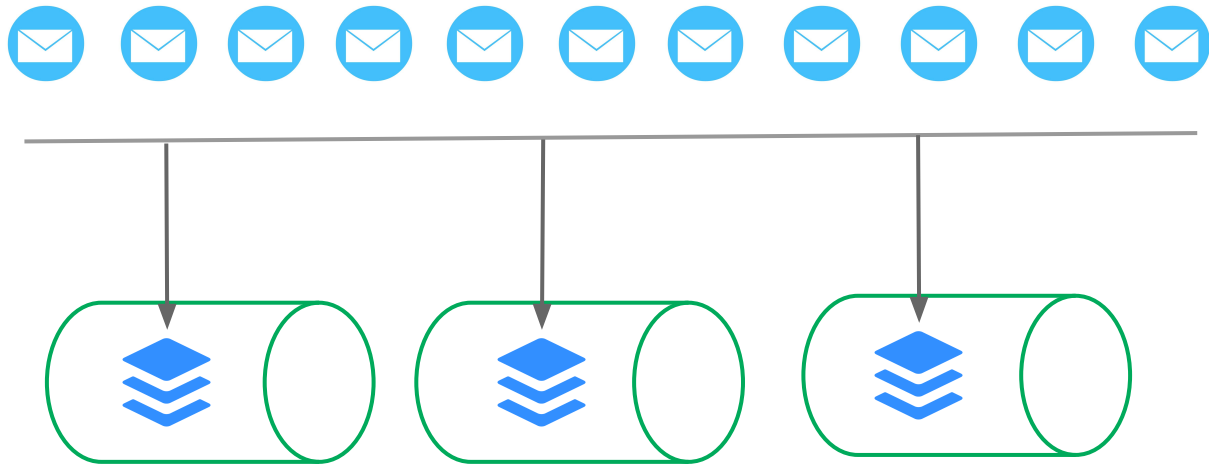
org.apache.kafka.clients.producer.KafkaProducer

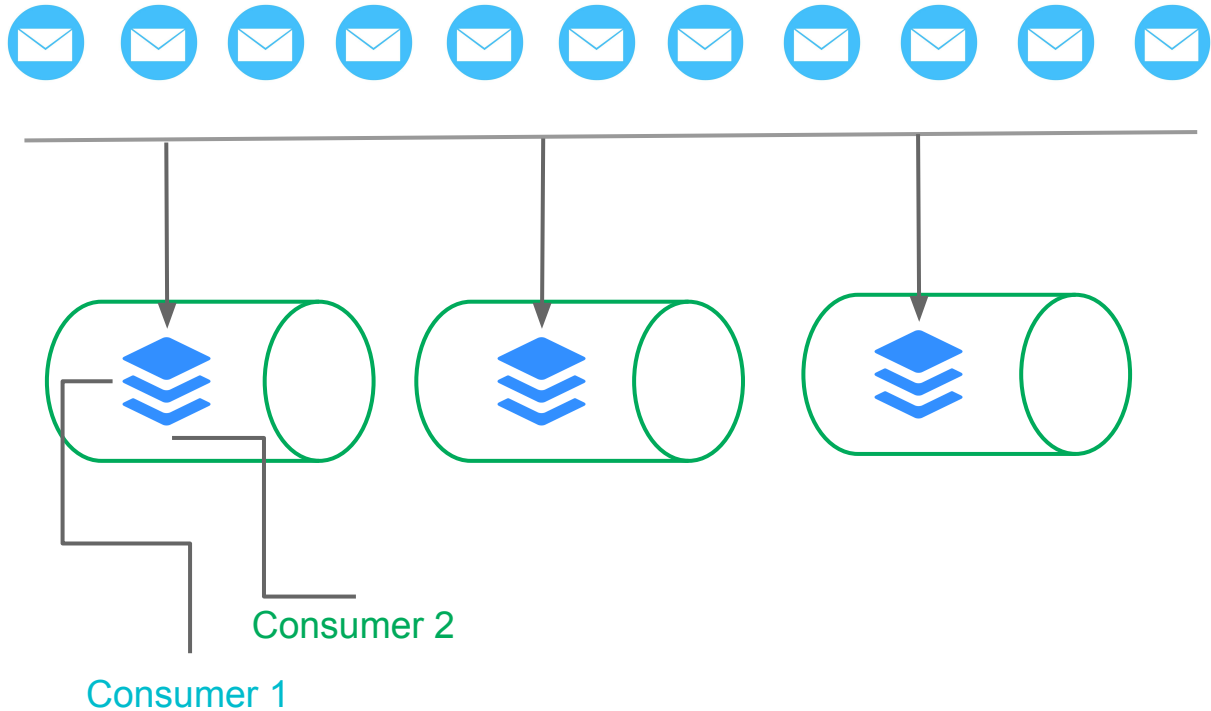# Topic

Remove messages based on

| number of messages | time | size |

kafka.common.Topic

Partitions

Serves: Horizontal scaling, Parallel consumer reads

kafka.cluster.Partition

Consumers - pull

Consumer 2

Consumer 1

kafka.consumer.ConsumerConnector,
kafka.consumer.SimpleConsumer

Consumer offsets

committing and fetching consumer offsets

kafka:// - protocol

"Binary protocol over TCP"

- Metadata
- Send
- Fetch
- Offsets
- Offset commit
- Offset fetch

Mechanical Sympathy

"The most amazing achievement of the computer software industry is its continuing cancellation of the steady and staggering gains made by the computer hardware industry." - Henry Peteroski
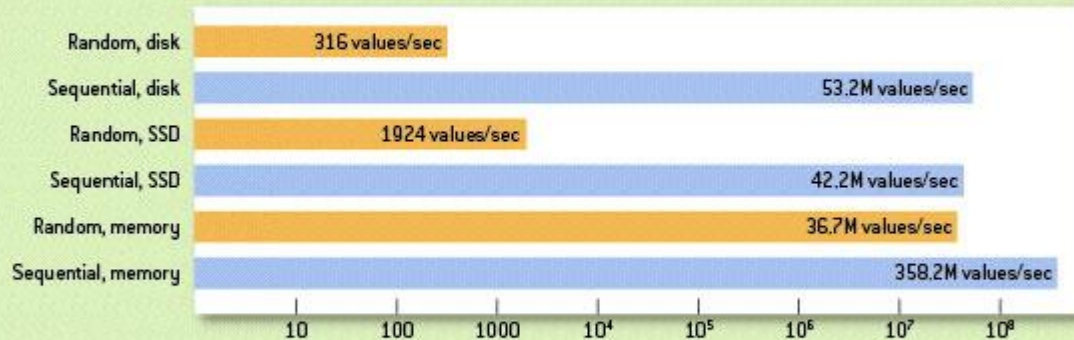
Persistence

"Everything is faster till the disk IO."

Disk faster than RAM

FIGURE 3

Comparing Random and Sequential Access in Disk and Memory

Random, disk	316 values/sec
Sequential, disk	53.2M values/sec
Random, SSD	1924 values/sec
Sequential, SSD	42.2M values/sec
Random, memory	36.7M values/sec
Sequential, memory	358.2M values/sec

10 100 1000 10^4 10^5 10^6 10^7 10^8

Note: Disk tests were carried out on a freshly booted machine (a Windows 2003 server with 64-GB RAM and eight 15,000-RPM SAS disks in RAID5 configuration) to eliminate the effect of operating-system disk caching. SSD test used a latest-generation Intel high-performance SATA SSD.

Linear Read & Writes

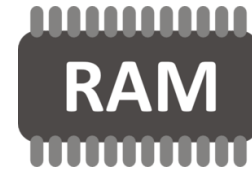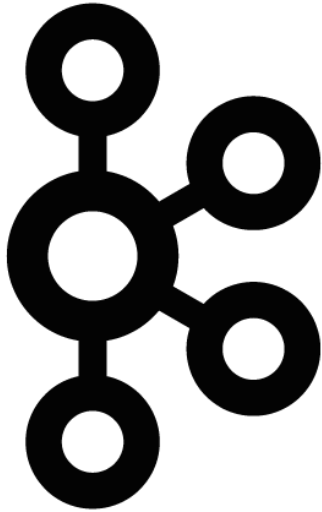*On high level there are only two operations:*

Append to end of log

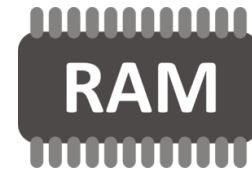fetch messages from a partition beginning from a particular message id
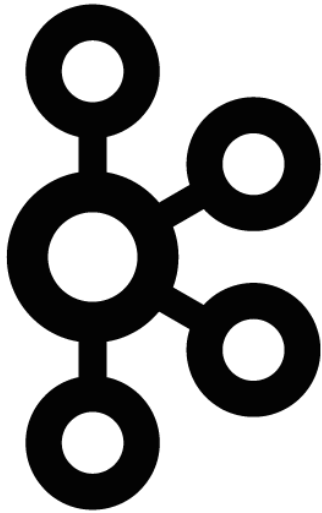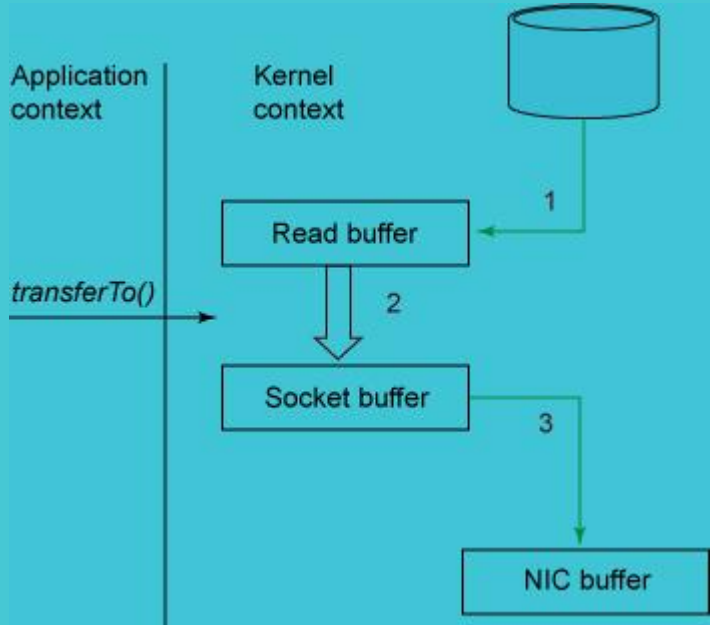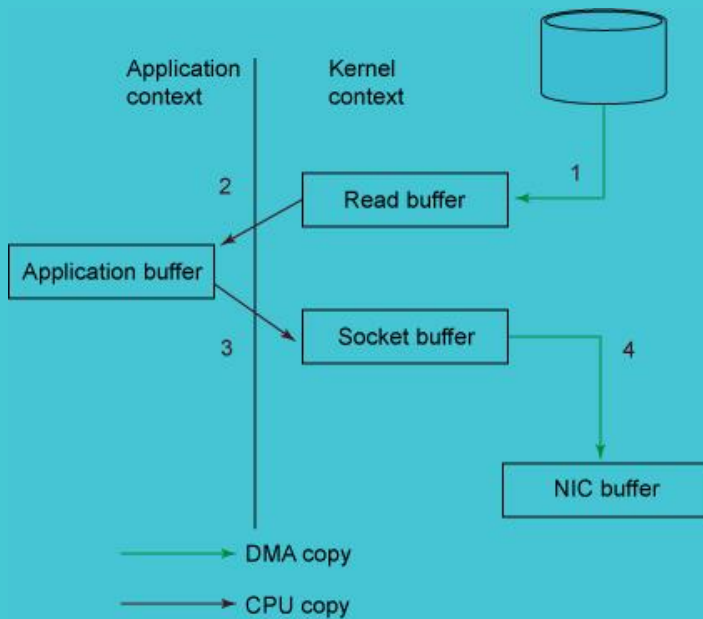
sequential file I/O

"Let us play pictionary"

Linux Page Cache

"Kafka ate my RAM"

ZeroCopy

src: http://www.ibm.com/developerworks/library/j-zerocopy/

Batching

small latency to improve throughput

img src: https://prashanthpanduranga.files.wordpress.com/2015/05/tirupati.jpg
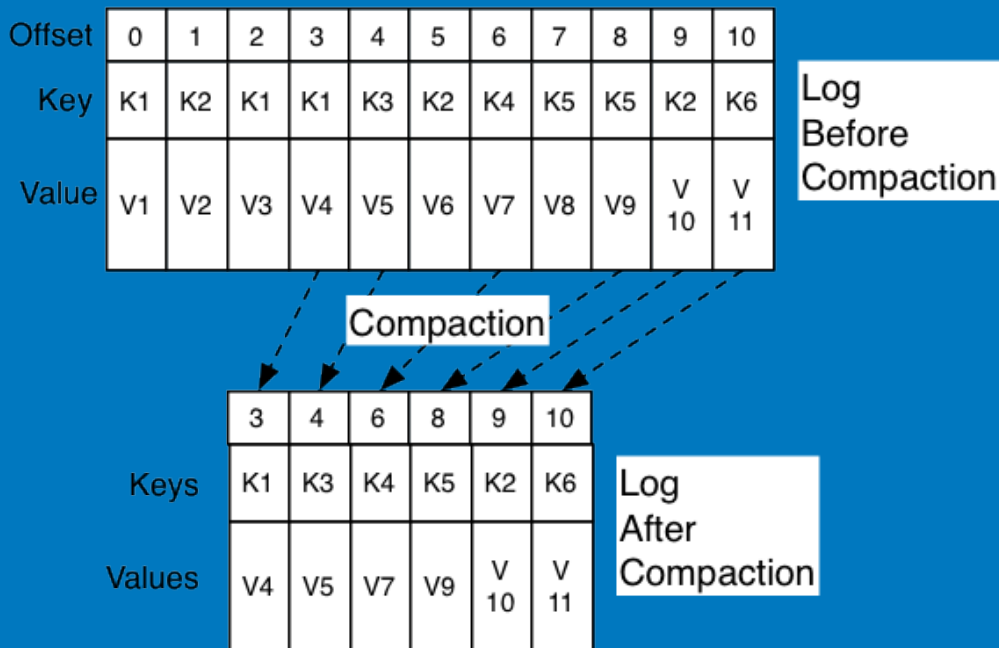
Compression

bandwidth is more expensive per-byte to scale than disk I/O, CPU, or network bandwidth capacity within a facility

kafka.message.CompressionCodec

Log compaction

img src: http://kafka.apache.org/083/documentation.html

Message Delivery

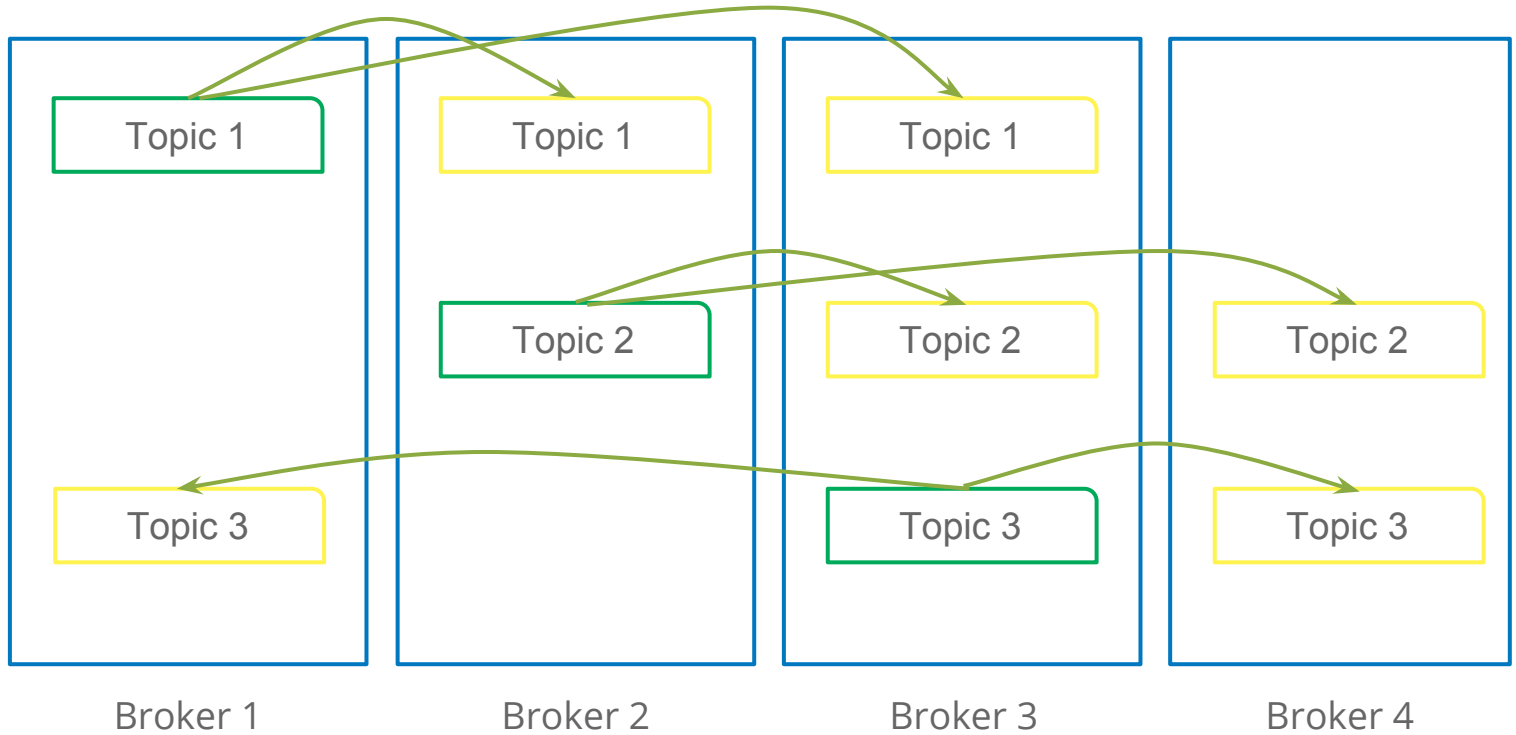Atleast once

Atmost once

Exactly once

Replication

un-replicated = replication factor of one

Quorum based

- Better latency
- To tolerate **"f"** failures, need **"2f+1"** replicas

Primary-backup replication

Broker 1   Broker 2   Broker 3   Broker 4

ZooKeeper

cluster coordinator

THANK YOU

For questions or suggestions:

Ran.ga.na.than B

ranganab@thoughtworks.com

@ran_than

ThoughtWorks®