

ThoughtWorks®

Apache: Big Data 2015

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."

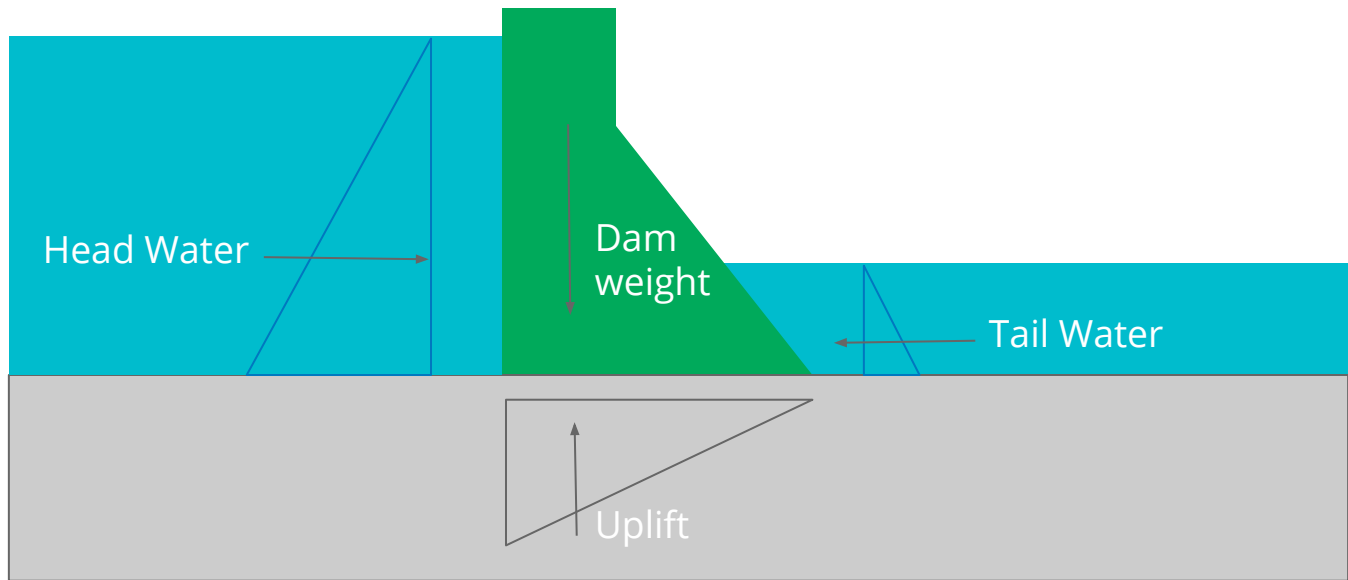
- Louis Sullivan



Gravity Dam

Indirasagar Dam, India

Forces on a gravity dam





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

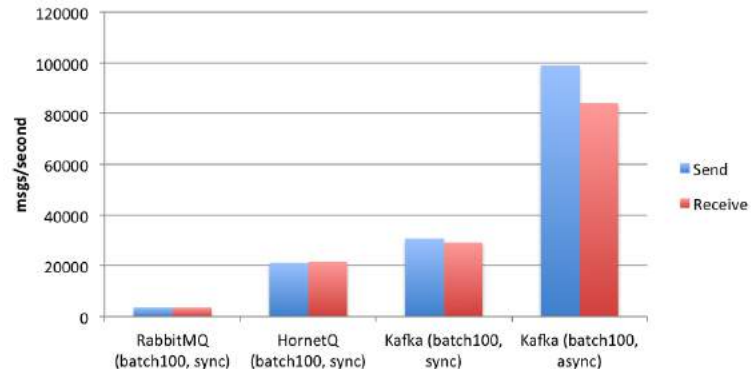
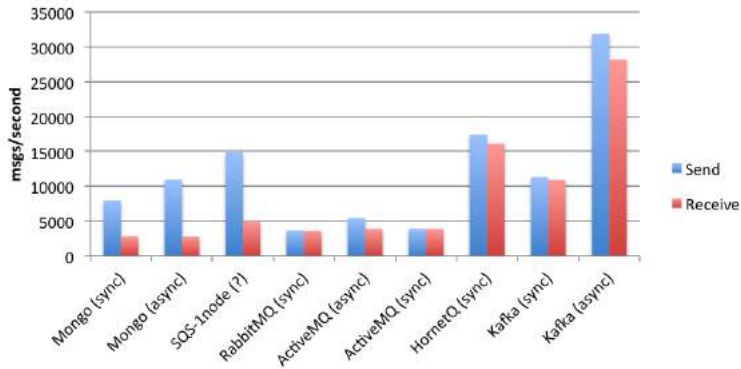
“producers produce, consumers consume, in large distributed reliable way -- real time”

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 Kafka create
Confluent

Latest stable - 0.8.2.1

2011

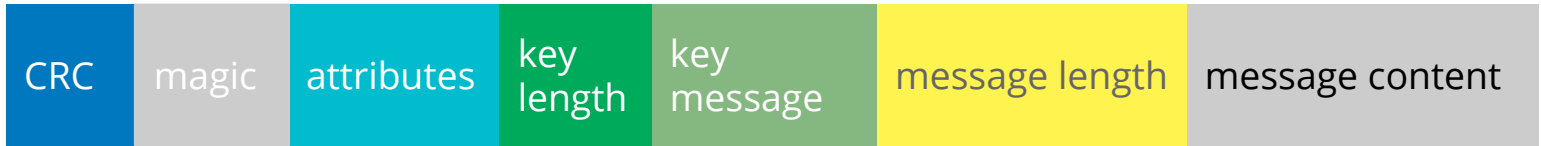
2012

2013

2014

2015

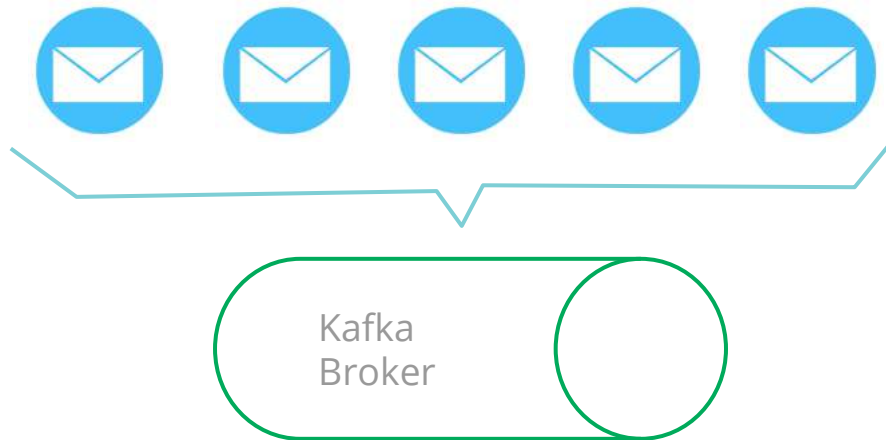
A Kafka Message



kafka.message.Message

Producers - push

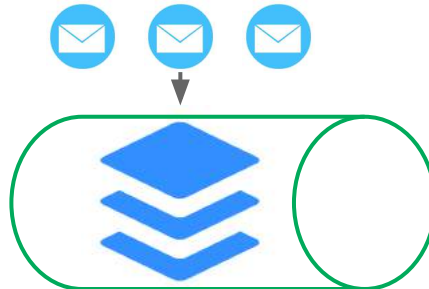
Request => RequiredAcks Timeout [TopicName [Partition MessageSetSize MessageSet]]



Response => [TopicName [Partition ErrorCode Offset]]

`org.apache.kafka.clients.producer.KafkaProducer`

Topic



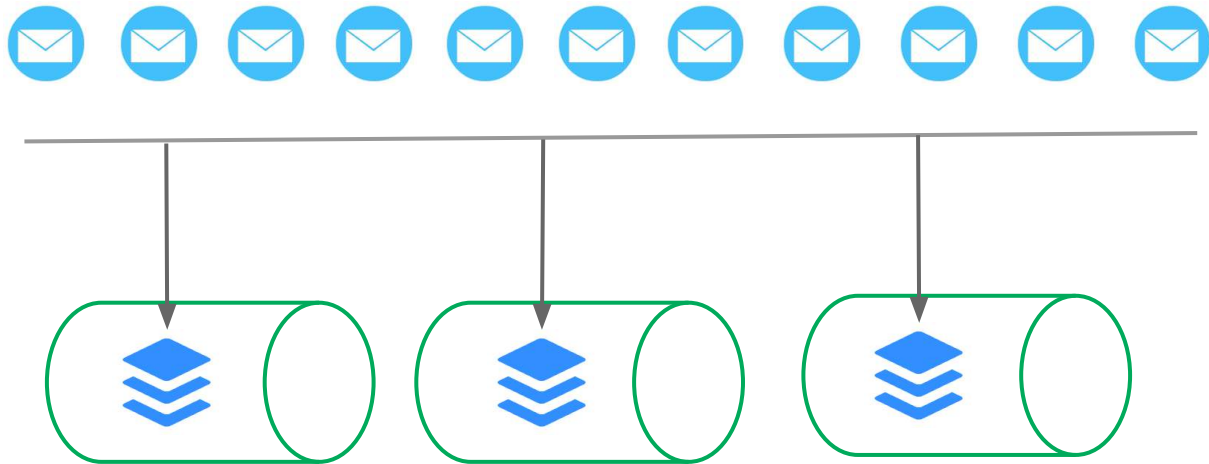
Remove messages based on

number of
messages

time

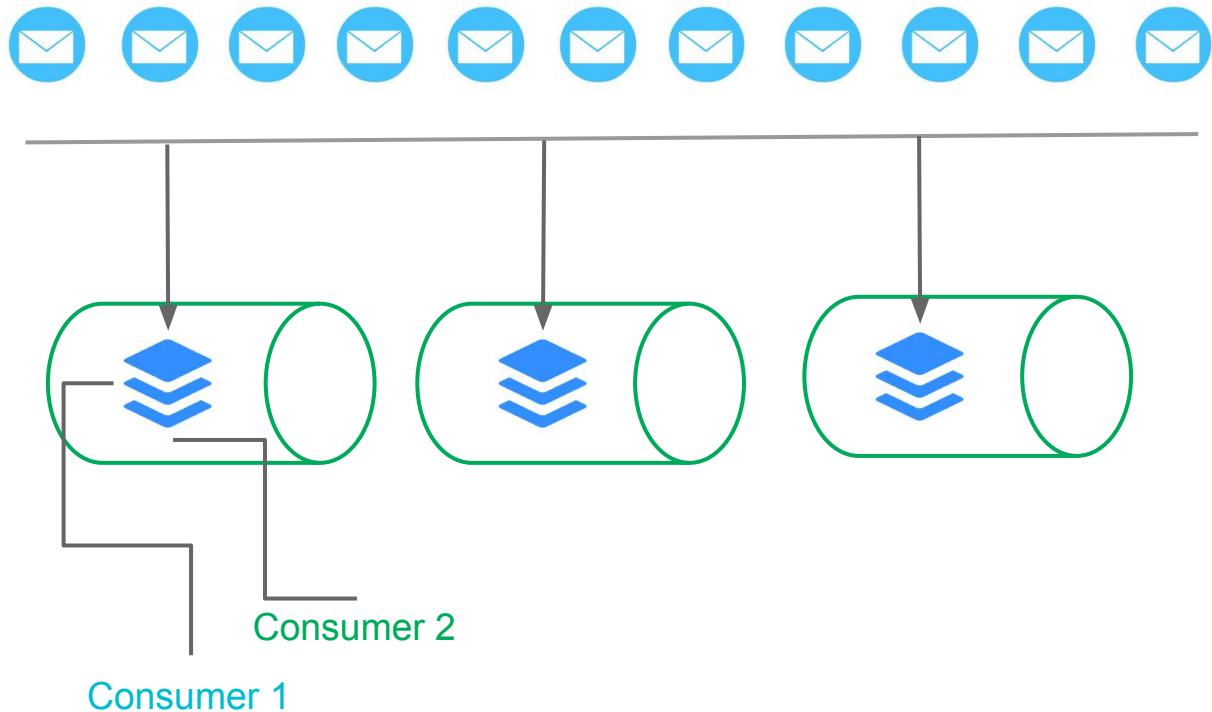
size

Partitions



Serves: Horizontal scaling, Parallel consumer reads

Consumers - pull



`kafka.consumer.ConsumerConnector,`
`kafka.consumer.SimpleConsumer`



Consumer offsets

committing and fetching consumer offsets

img src: <http://www.reynanprinting.com/photos/undefined/impression-offset1.jpg>

kafka:// - protocol

"Binary protocol over TCP"

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



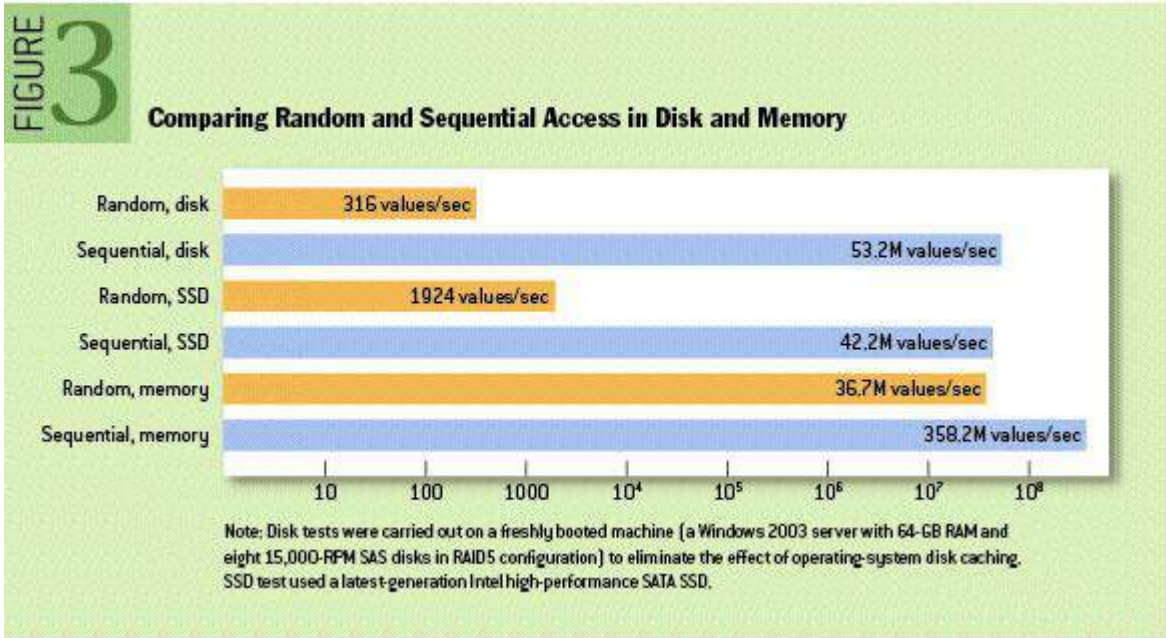
Mechanical Sympathy

"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



Linear Read & Writes

On high level there are only two operations:

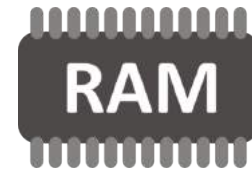
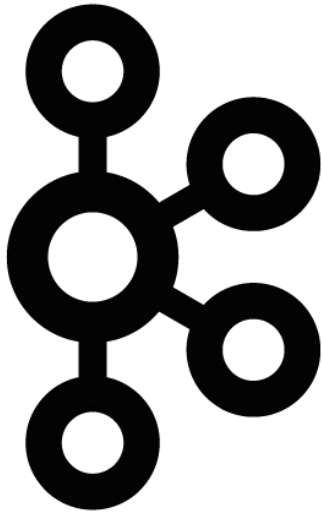


Append to end of log

fetch messages from a
partition beginning from a
particular message id

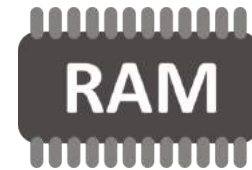
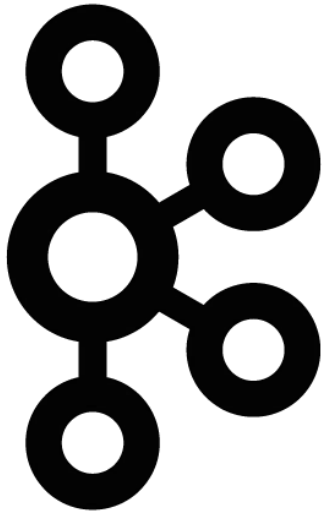
sequential file I/O

“Let us play pictionary”

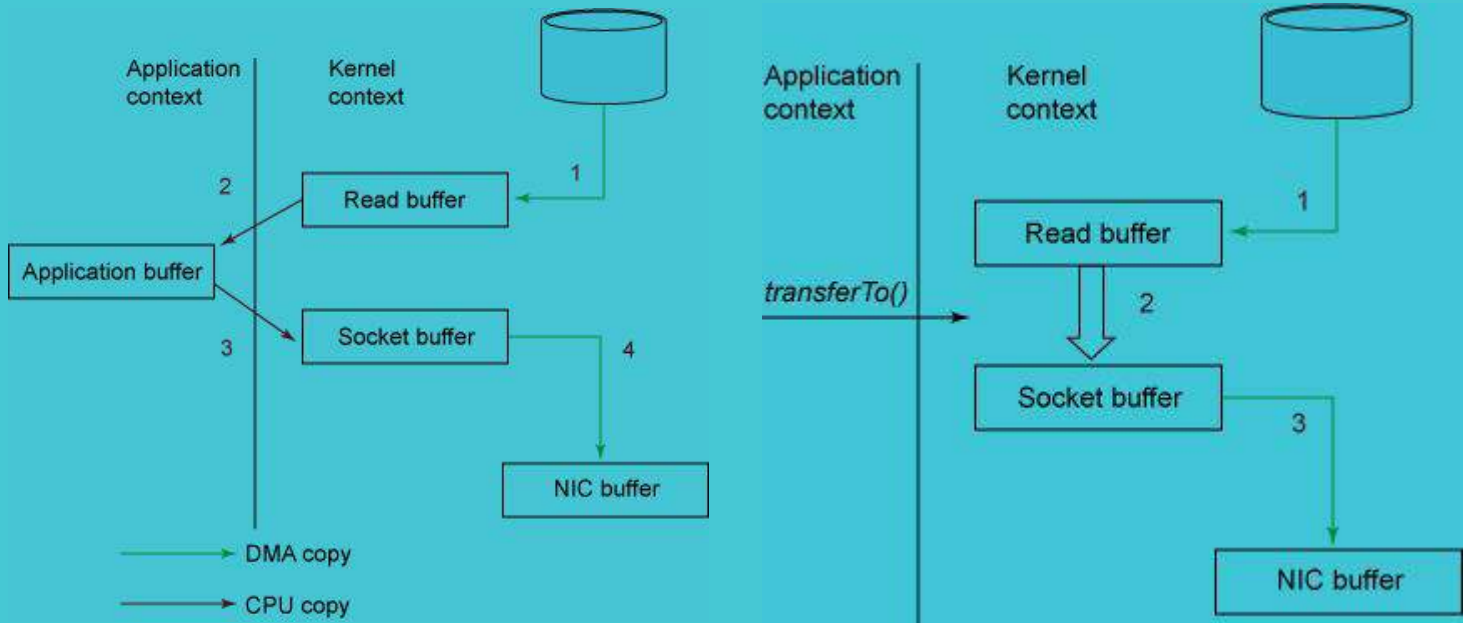


Linux Page Cache

"Kafka ate my RAM"



ZeroCopy





Batching

small latency to improve throughput

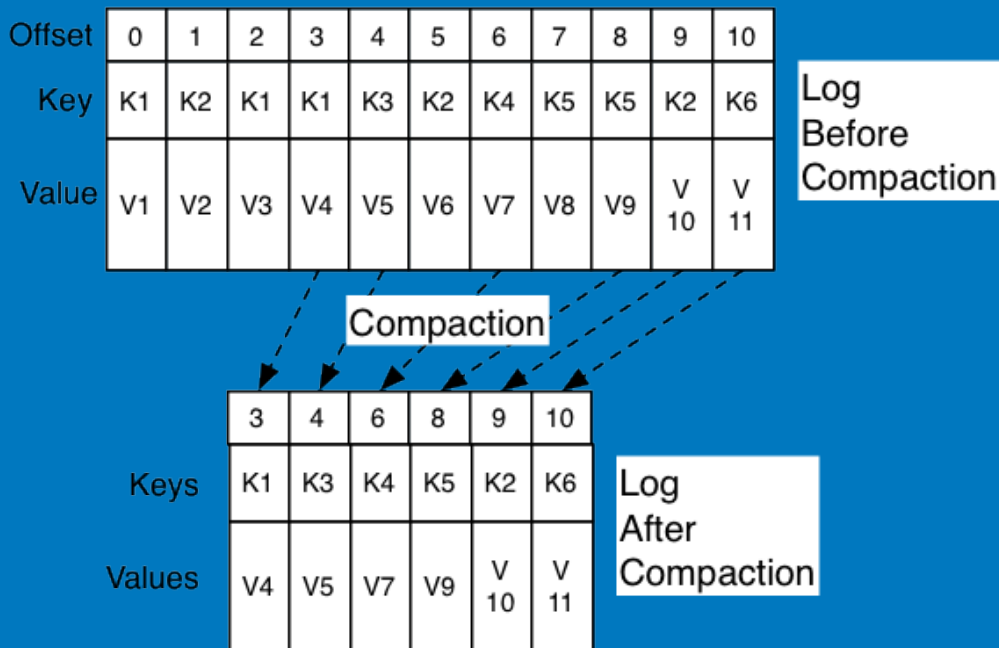
img src: <https://prashanthpanduranga.files.wordpress.com/2015/05/tirupati.jpg>



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

Message Delivery

A blue message card with a white border and a white diagonal line from the top-left corner to the bottom-right corner.

Atleast once

A green message card with a white border and a white diagonal line from the top-left corner to the bottom-right corner.

Atmost once

An orange message card with a white border and a white diagonal line from the top-left corner to the bottom-right corner.

Exactly once

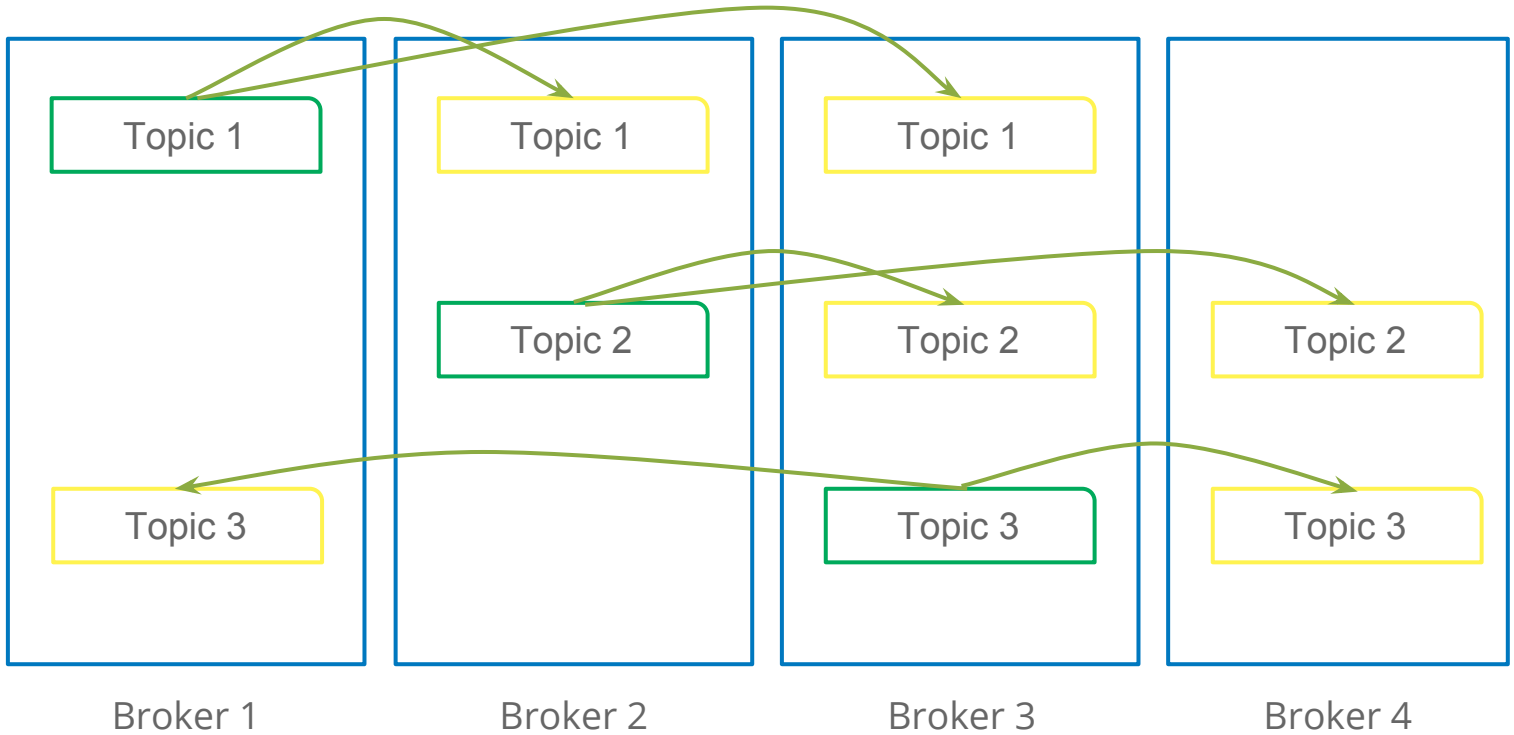
Replication

un-replicated = replication factor of one

Quorum based

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

Primary-backup replication



ZooKeeper

cluster coordinator



THANK YOU

For questions or suggestions:

Ran.ga.na.than B

ranganab@thoughtworks.com

@ran_than

ThoughtWorks®