kafka

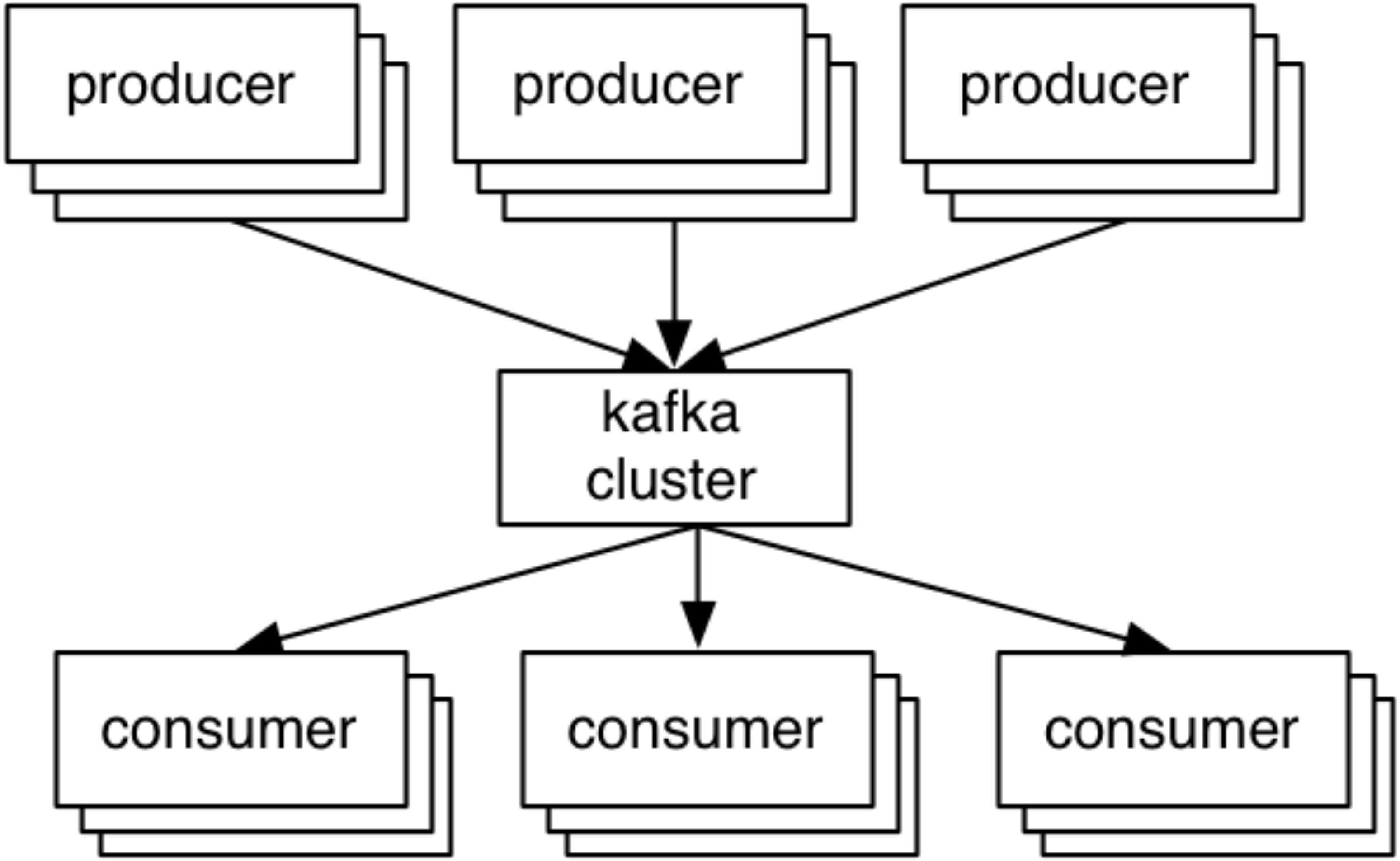
History

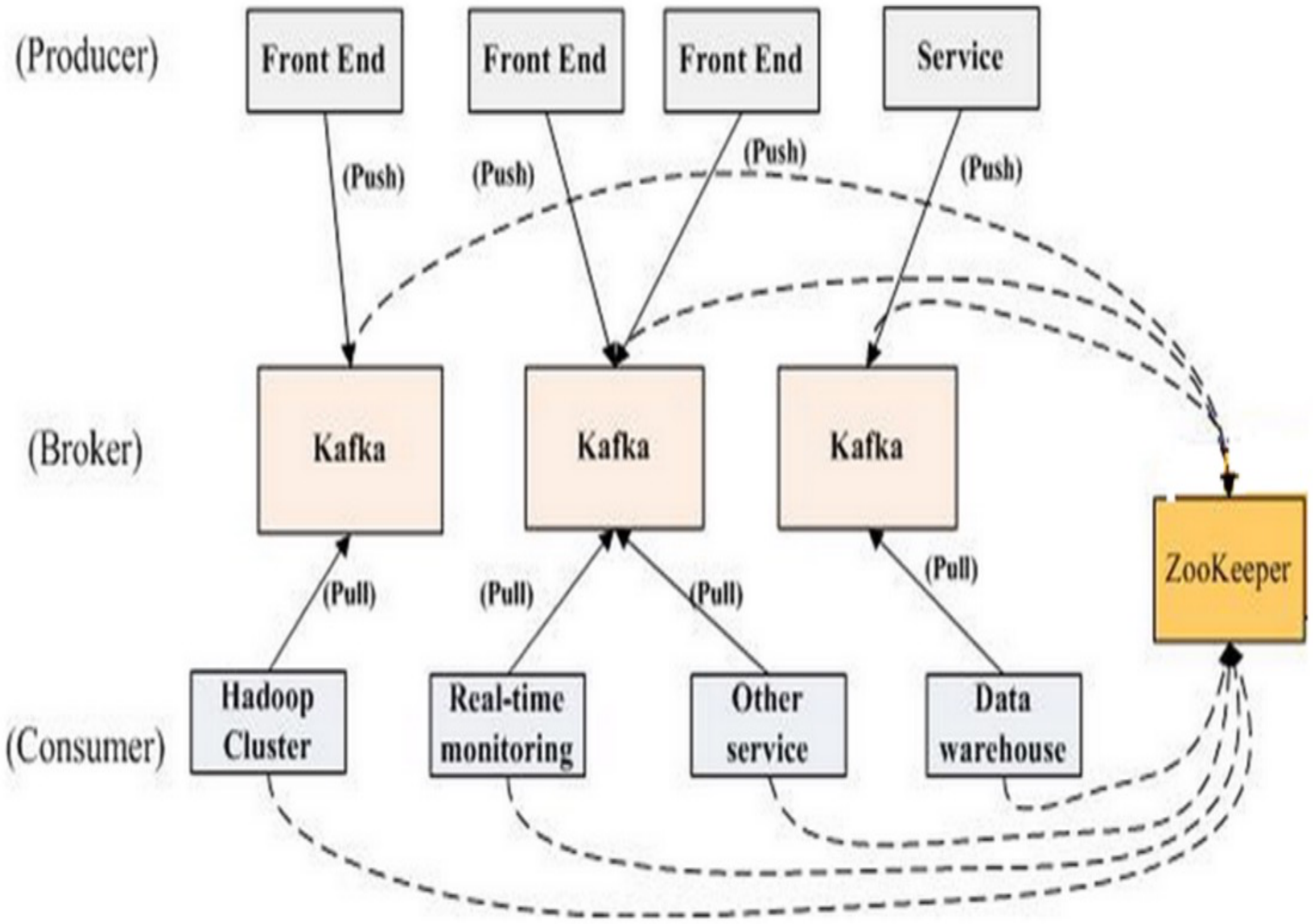
- Originally developed by LinkedIn
- Opensourced in early 2011
- “Confluent” with a focus on Kafka
- Written in Scala
- 9 core committers, + 20 contributors

Features

- Distributed messaging system
- Provide a unified, high-throughput, low-latency platform for handling real-time data feeds
- Up to 2M writes/reads on 3 commodity machine cluster

Kafka Abstract




(Producer)
Front End
Front End
Front End
Service
(Push)
(Push)
(Push)
(Push)
(Broker)
Kafka
Kafka
Kafka
ZooKeeper
(Pull)
(Pull)
(Pull)
(Pull)
(Consumer)
Hadoop Cluster
Real-time monitoring
Other service
Data warehouse

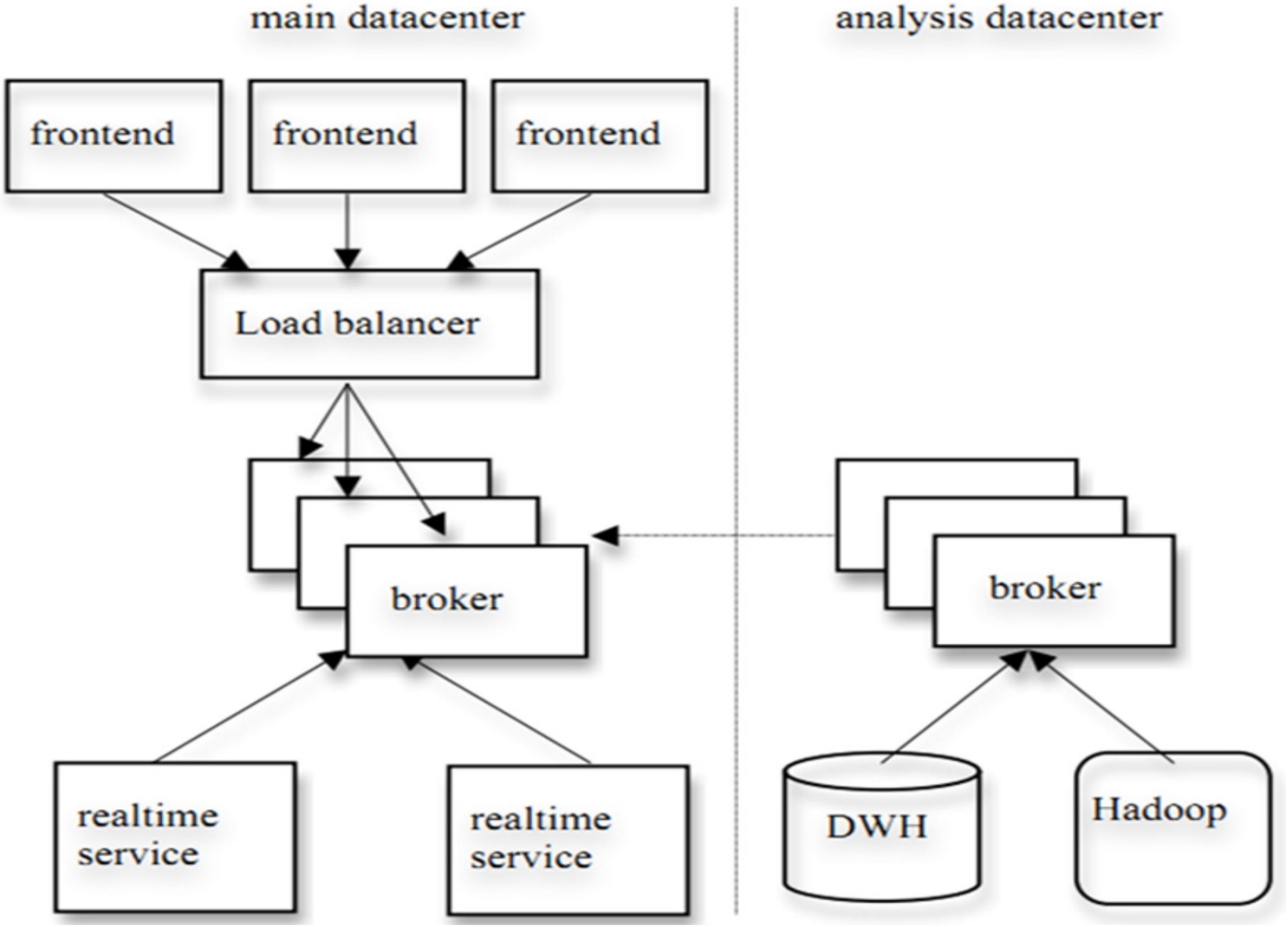
Kafka at LinkedIn

- 350+ commodity machines
- 8,000+ topics
- 140,000+ partitions

- 278 Billion messages/day
- 49 TB/day in
- 176 TB/day out

- Peak Load
  - 4.4 Million messages per second
  - 6 Gigabits/sec Inbound
  - 21 Gigabits/sec Outbound

Real-time and Batch Processing



Messaging Component

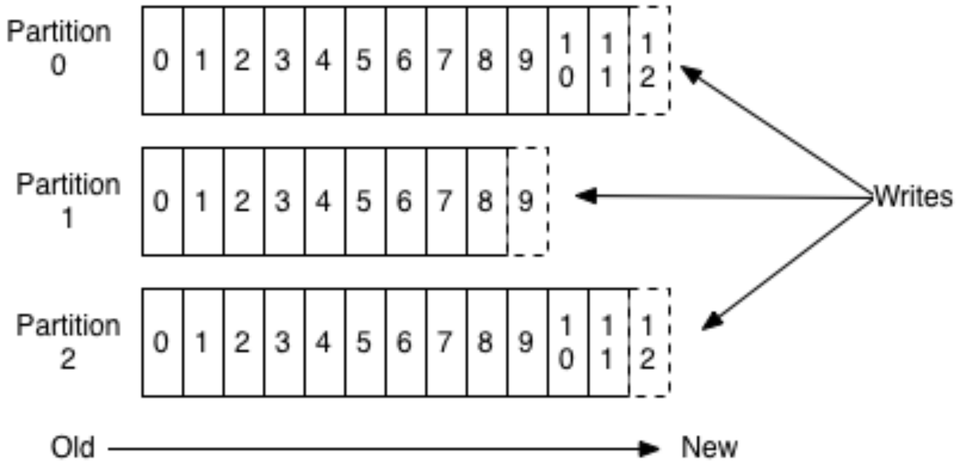
- Maintains feeds of messages in categories called ***topics***
- Call processes that publish messages to a Kafka topic ***producers***
- Call processes that subscribe to topics and process the feed of published messages ***consumers***
- Run as a cluster comprised of one or more servers each of which is called a ***broker***

Topic

- Category or feed name to which messages

- partitioned log

Anatomy of a Topic



Topic Partition Distribution

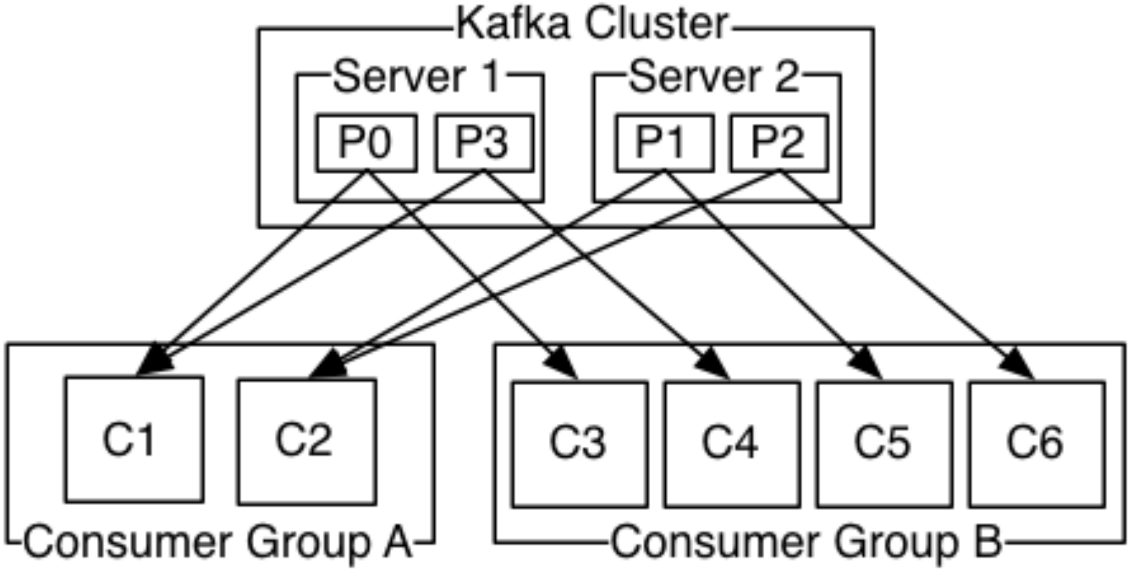
- Each partition is replicated
- Each partition has one server which acts as the "leader“
- The leader handles all read and write requests
- One or more servers which act as "followers“
- If the leader fails, one of the followers will automatically become the new leader

Producer

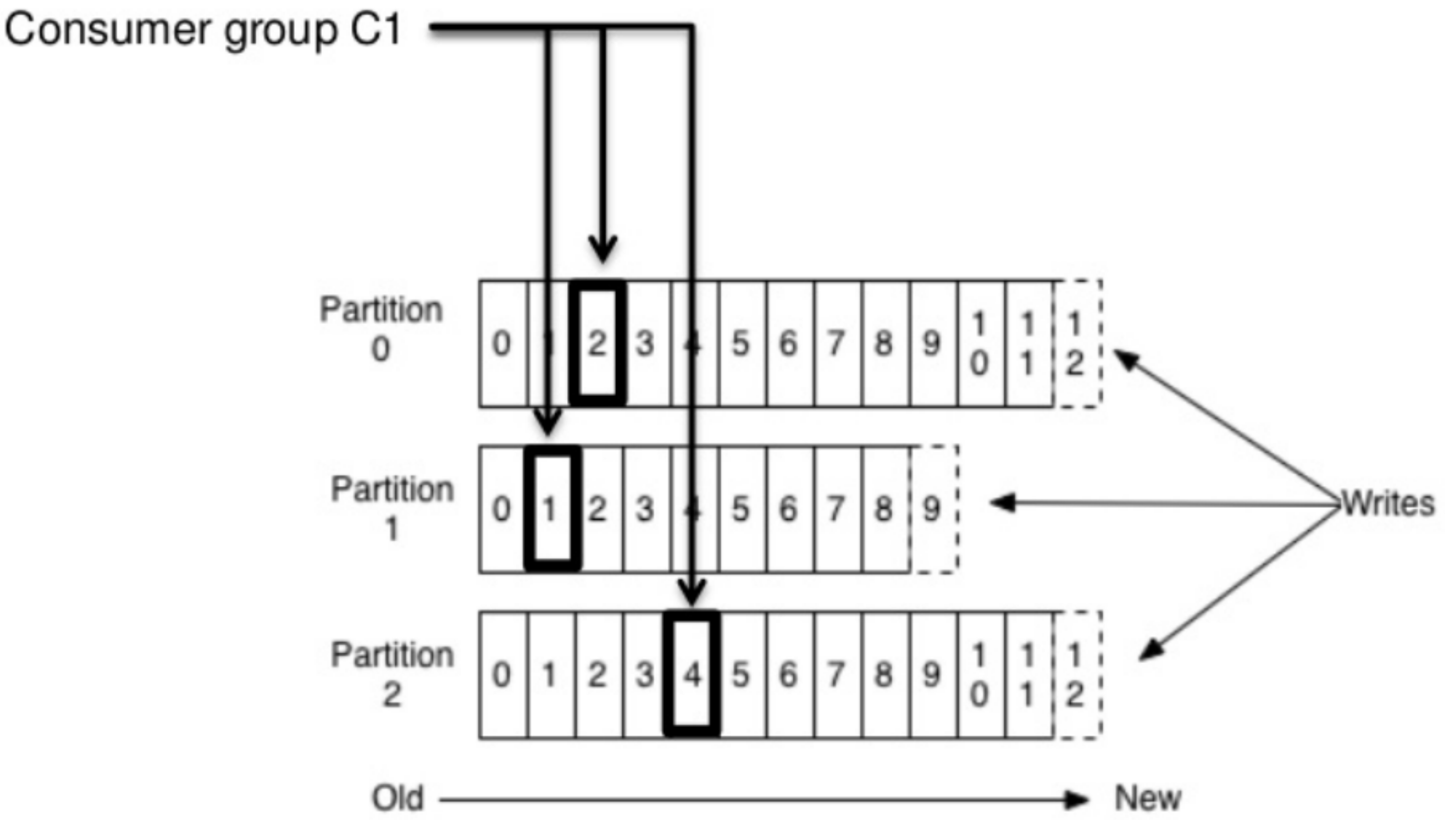
- Publish data to the topics
- Decide message to assign to which partition

Consumer

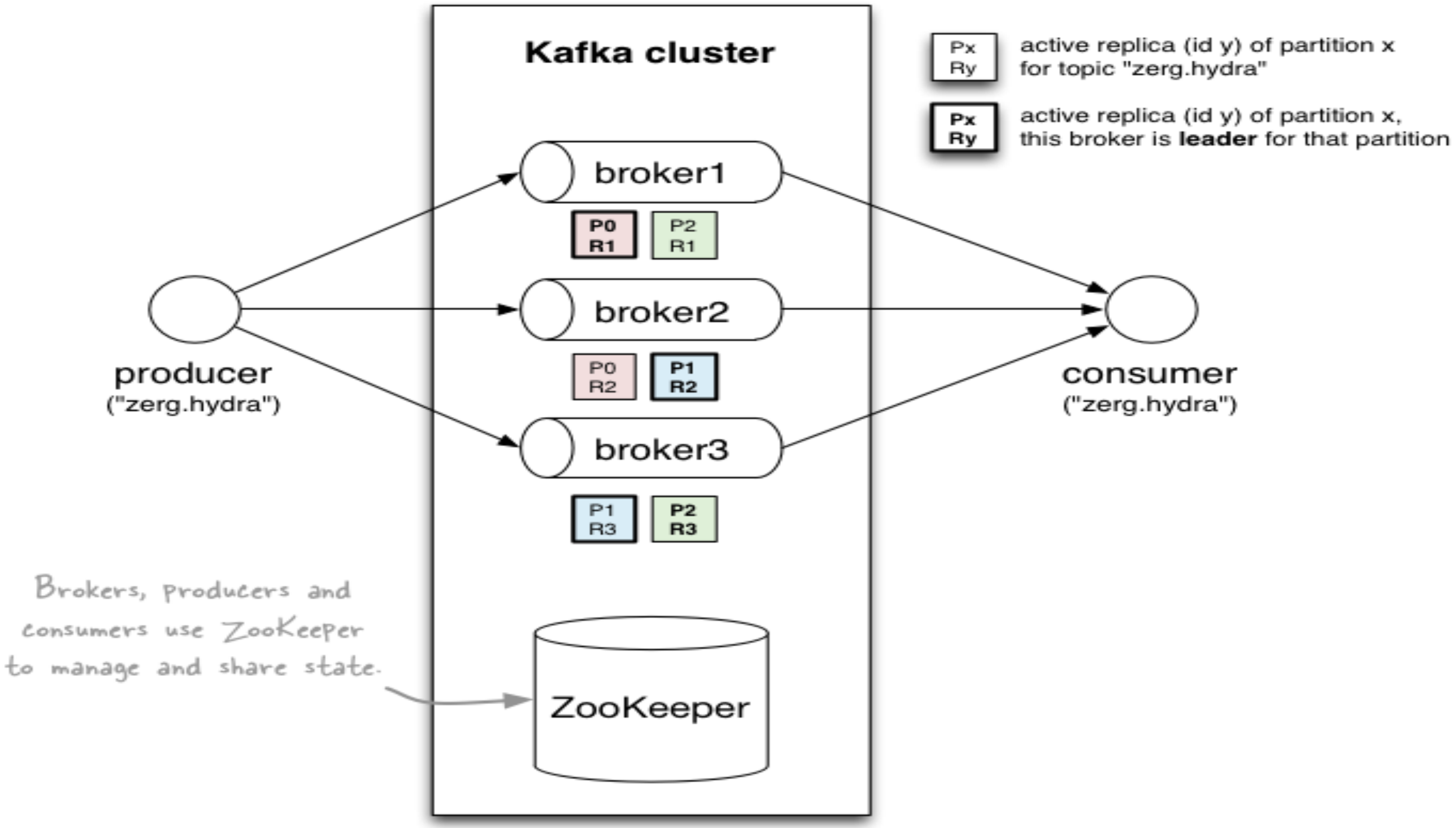
- Consumers assign to consumer group name
- If only one group, work like message queue
- If not, work like publish subscribe



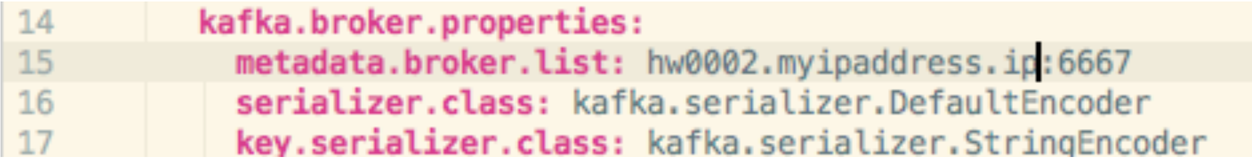
Partition Offset



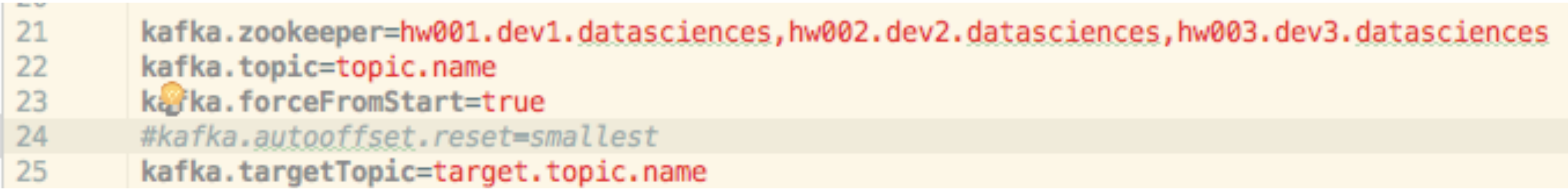
Data Flow


Kafka cluster
Px
Ry
active replica (id y) of partition x
for topic "zerg.hydra"
Px
Ry
active replica (id y) of partition x,
this broker is **leader** for that partition
broker1
P0
R1
P2
R1
broker2
P0
R2
P1
R2
broker3
P1
R3
P2
R3
producer
("zerg.hydra")
consumer
("zerg.hydra")
Brokers, producers and
consumers use ZooKeeper
to manage and share state.
ZooKeeper

Properties/Yaml File



```
kafka.broker.properties:
  metadata.broker.list: hw0002.myipaddress.ip:6667
  serializer.class: kafka.serializer.DefaultEncoder
  key.serializer.class: kafka.serializer.StringEncoder
```



```
kafka.zookeeper=hw001.dev1.datasciences,hw002.dev2.datasciences,hw003.dev3.datasciences
kafka.topic=topic.name
kafka.forceFromStart=true
#kafka.autooffset.reset=smallest
kafka.targetTopic=target.topic.name
```

Major Use Case

- Messaging
- Website Activity Tracking
- Metrics – operational monitoring data
- Log Aggregation
- Stream Processing
- Event Sourcing
- Commit Log

Quick Start Demo

- Create a topic
- Describe a topic
- List topics
- Send message
- Consume message

Reference

- http://kafka.apache.org/documentation.html
- http://www.confluent.io/blog/how-to-choose-the-number-of-topicspartitions-in-a-kafka-cluster/