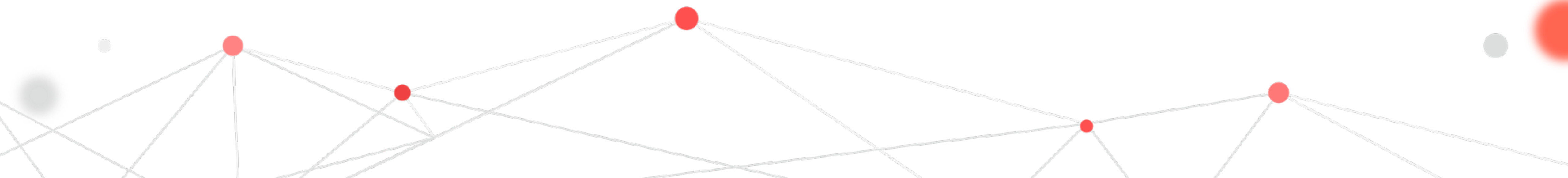




Kafka培训3-客户端

阿昌.何伟昌
2016.07.12

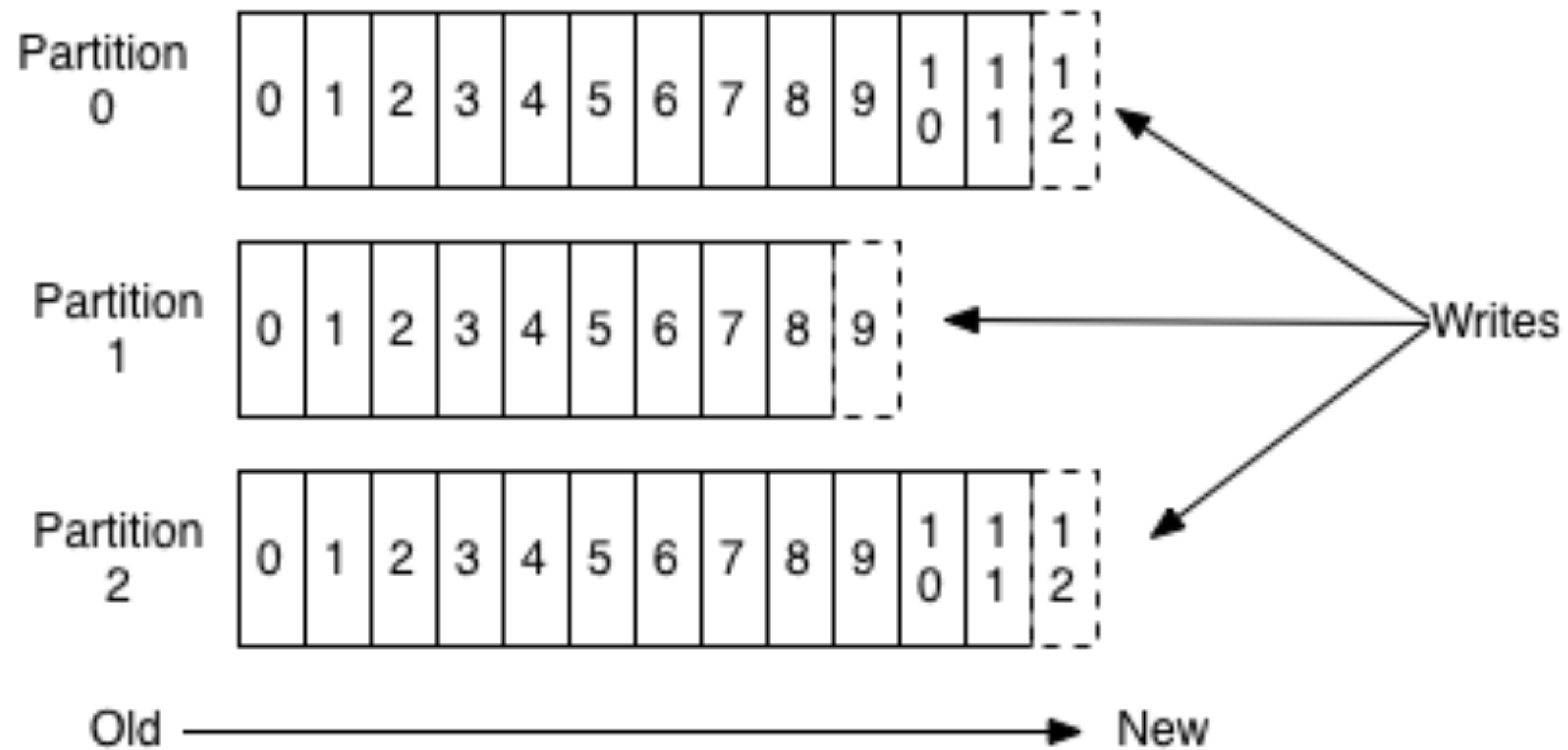


“Kafka是一个分布式的、基于分区的、具有副本的日志服务。
它提供了消息队列的功能，但基于一个独特的设计。”

—— Kafka官方介绍

基础：Topic & Partition

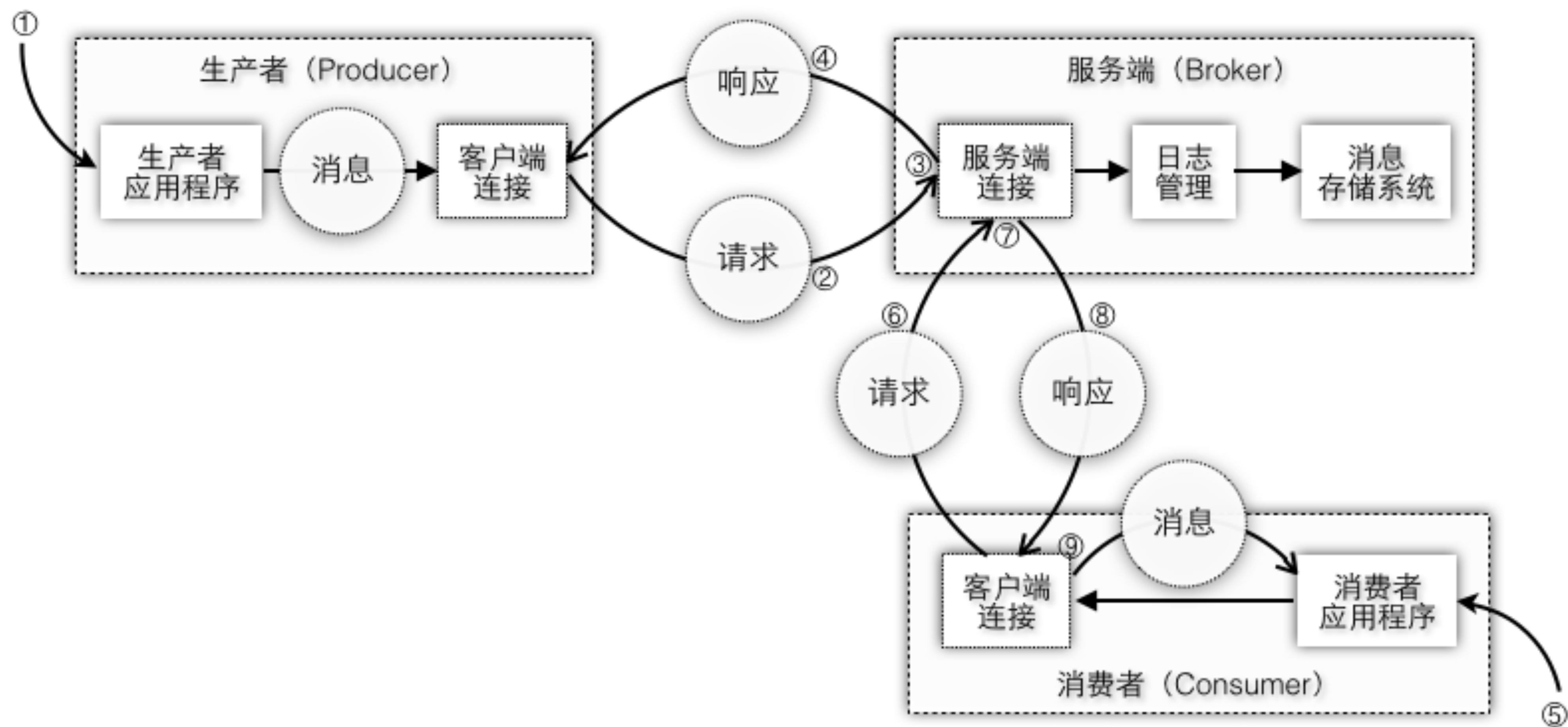
Anatomy of a Topic



@Juanpi, 对客户端的问题是？

- 消息能否有序？异步发送？
- Group.Id的含意，具体有什么用？
- 『至少一次』消费语义（Solr项目例子）？
- 客户端有哪些最佳实践？

基础：客户端与服务器交互

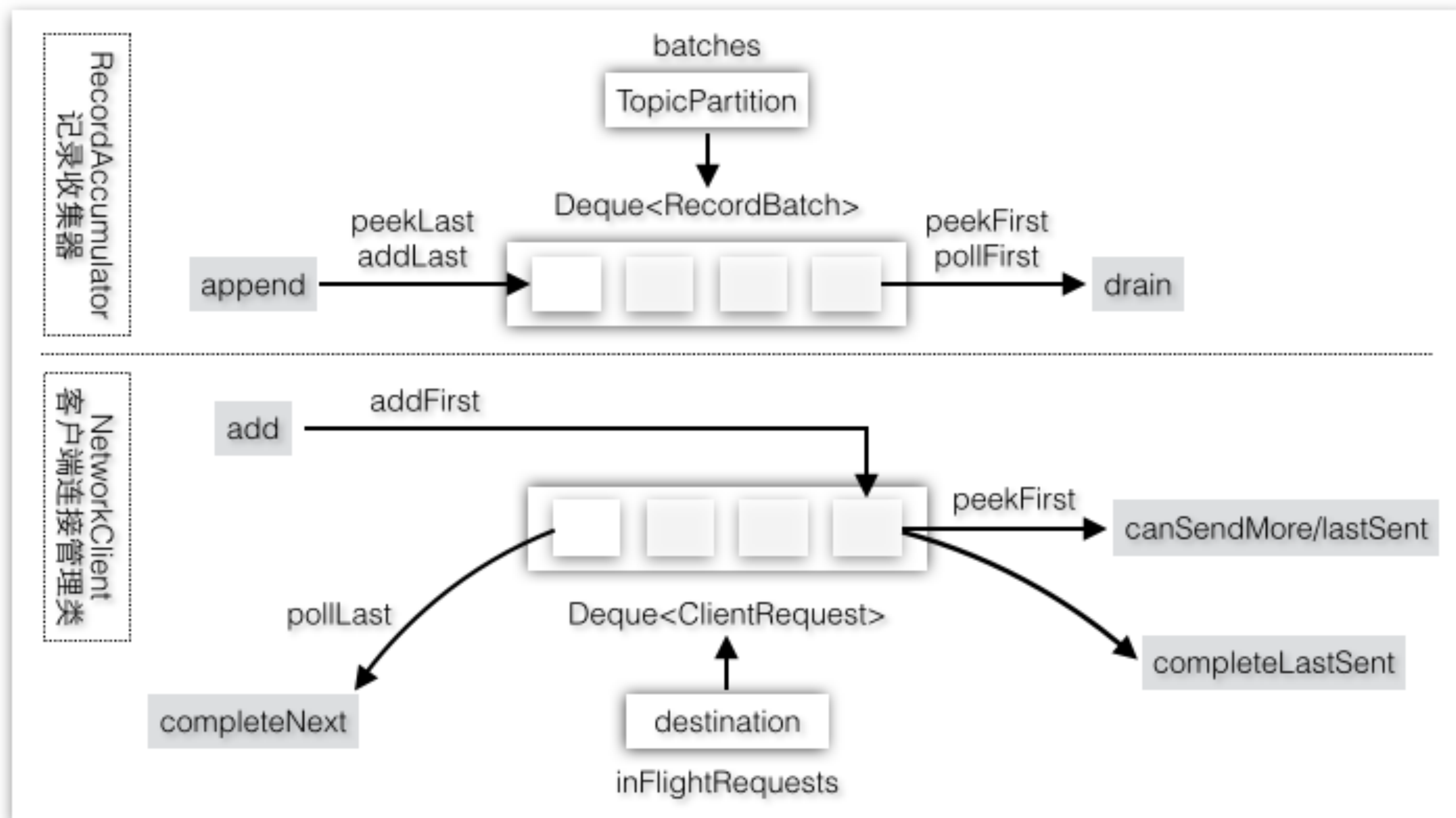


生产者

单分区有序

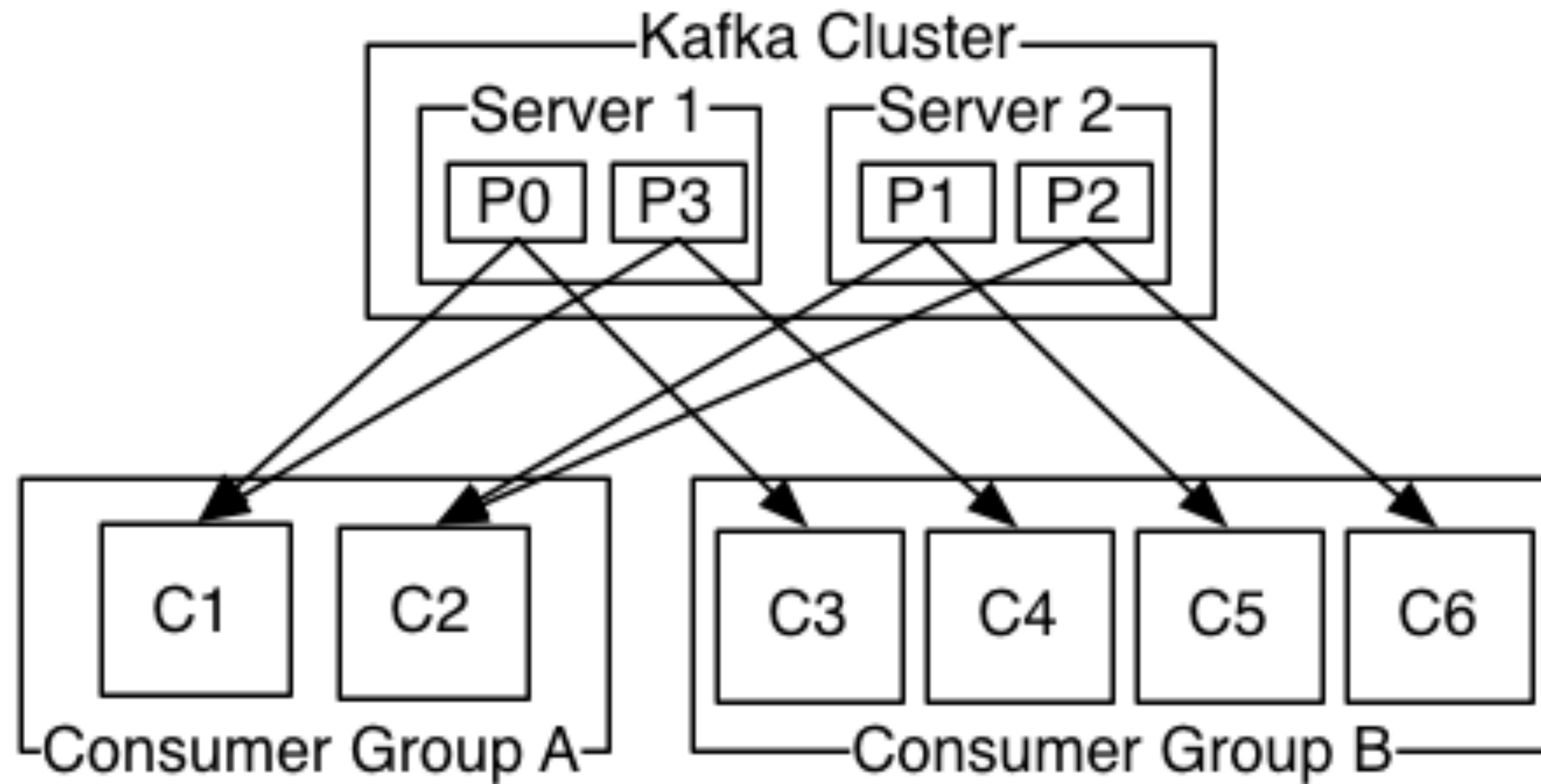
- 一个Topic分配N个分区。
- 生产者默认会以轮询的方式把消息发送到多个Leader分区
- Kafka提供了指定分区的方法，如订单Solr项目会根据userId作为分区标准。

异步批量发送，双端队列



消费者

分区数决定消费并行度



消费者组 (group.id)

- 不同的消费者组，同一条消息会被多个消费组消费，实现广播（一对多）。
- 相同一个消费组，同一条消息只会被这个消费组的一个消费者所消费，实现单播（一对一）。

拉模式 (pull)

- 消费者获取消息是使用拉模式
- 好处：服务器可以批量返回消费者，消费者可以控制消费速度。
- Offset控制消费进度，每个分区对应一个Offset，不同组不同Offset。
- 0.9.0.0会导致客户端的CPU负载变高，[KAFKA-315](#)

消费语义

- 至多一次 -- 消息会丢失，但不会重复消费。
- **至少一次 -- 消息不会丢失，但会重复消费。**
- 恰好一次 -- 消息不会丢失，也不会重复消费。

消费者处理过慢问题

- 现象：消息重复消费，甚至不停消费。
- 原因：如果消费者处理过慢（session timeout为30s），没有向kafka broker发送心跳和提交Offset，Broker就会发起客户端的Rebalancing
- 增大session time的时间或者减小最大拉取消息值（max.partition.fetch.bytes，默认值为1M）来暂缓问题。Kafka社区讨论这问题。增加最大拉取消息数的参数（max.poll.records）。

案例一： 订单Solr项目与Kafka

- 场景： 主要用Kafka来订阅订单消息。
- 问题： 订单信息要一条一条写入Solr，单条写入Solr大概是10ms，压测的时候发现消息不断地重复消费，然后追查发现订单1M的订单消息条数为3.5K，于是消费完这批数据需要35秒 ($3.5K * 10ms$)
- 临时解决办法： 减小最大拉取消息量
- 正确做法： 改为批量写入，利用『至少一次』语义来保证整批数据写入。

案例二： 一个Topic对应一个Handler?

- Topic的分区已在消费者初始化时确认。
- 两个系统之间通信本来只需要一个Topic
- 通过消息的标记来区分。



2016.07.12