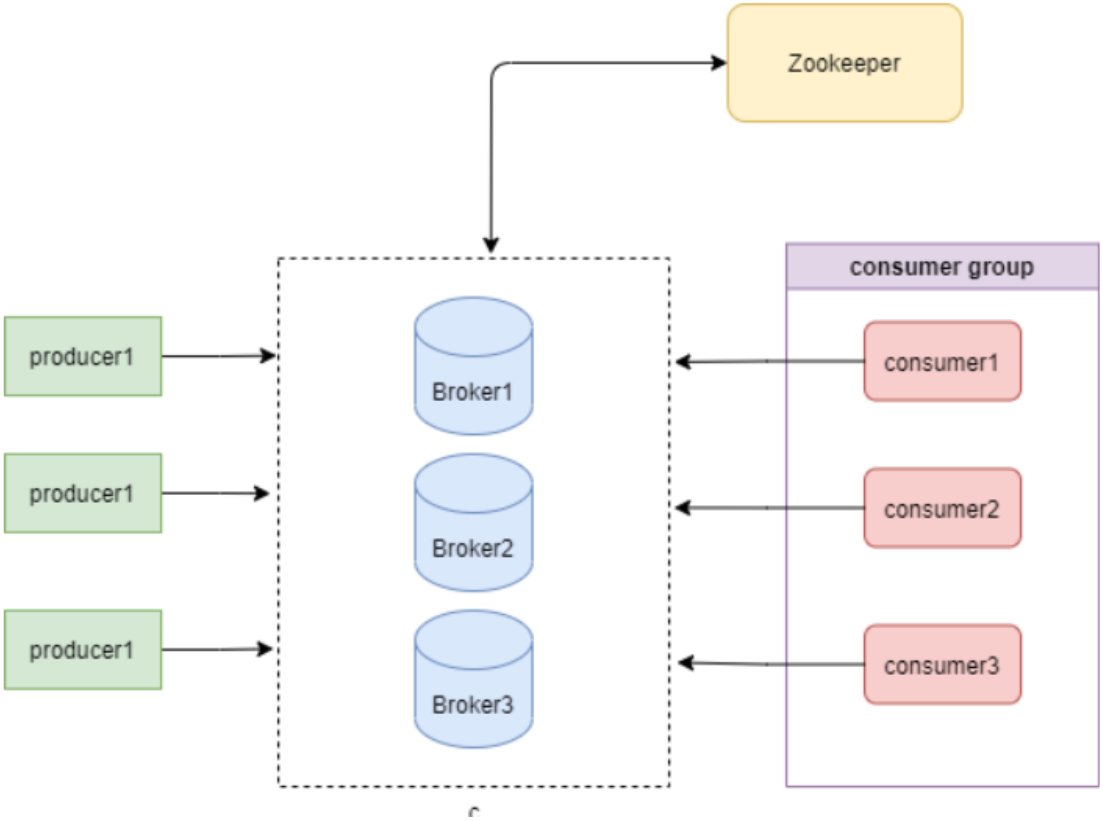
Apache Pulsar

kafka 和 Pulsar 架构

Kafka

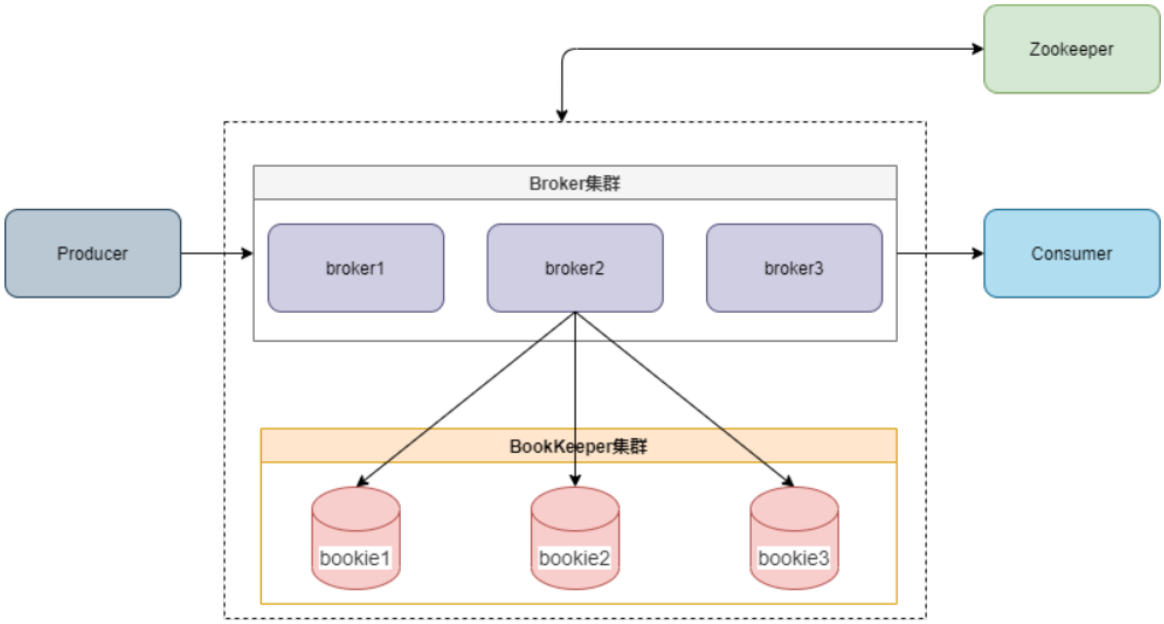
Kafka架构由broker和zookeeper组成，如下图：



> 注意：Kafka2.8版本可以不依赖Zookeeper独立运行了

Pulsar

Pulsar的架构如下：



> Pulsar Broker会在本地缓存消息，并且支持TTL，

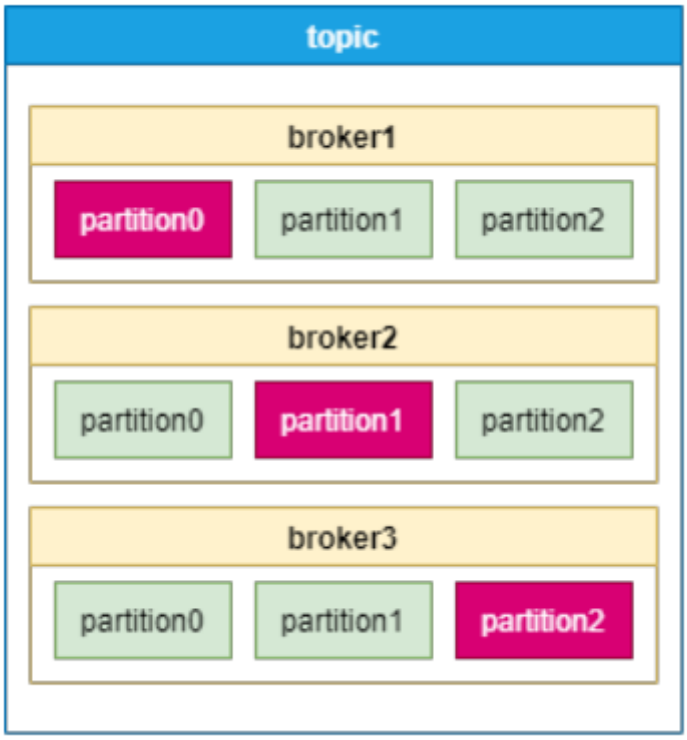
从上面的2个架构我们看到，Kafka和Pulsar有3点不同：

- Pulsar采用分层架构，将计算和存储相分离，存储使用BookKeeper集群，计算使用Broker集群，Broker需要内置BookKeeper客户端。
- Pulsar的部署和架构更加复杂，但是也更具有伸缩性。
- Pulsar在最新版本中依然不能脱离Zookeeper独立运行。

消息存储模型

Kafka

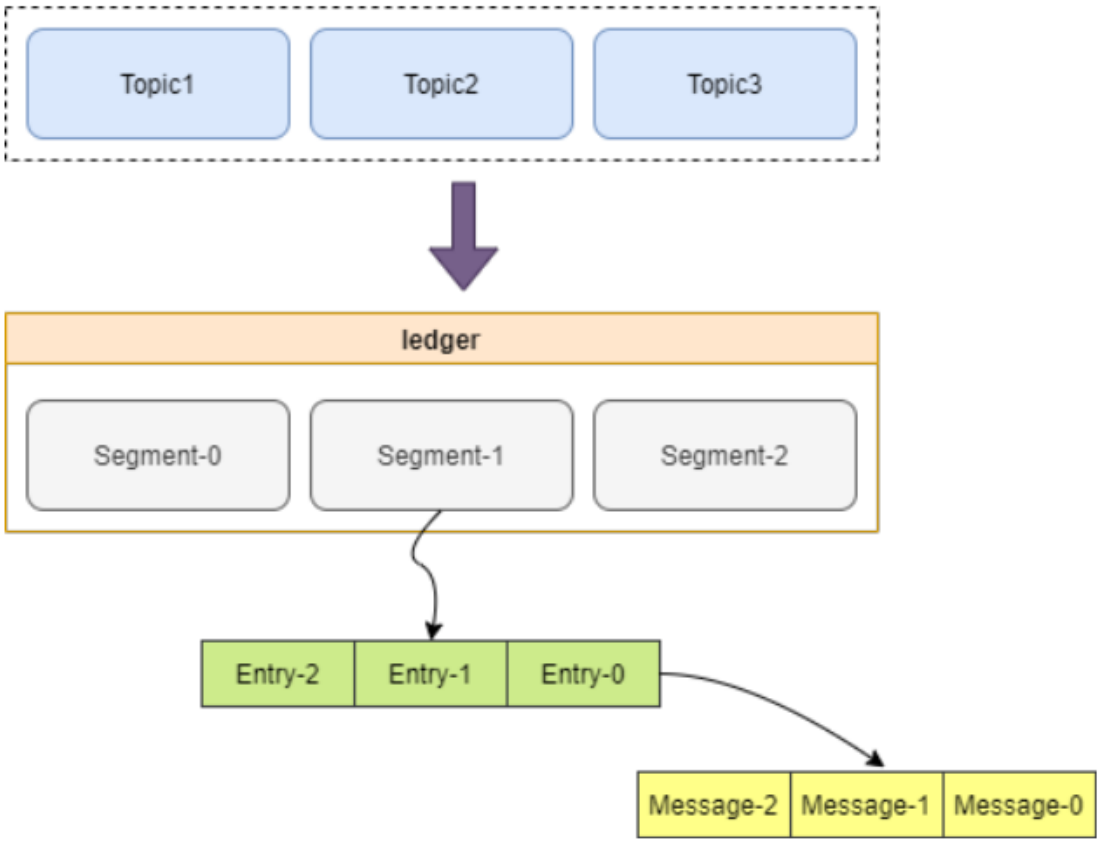
Kafka采用分区(Partition)的方式来保存topic，模型图如下：



每个topic都会在不同的broker保存多个分区副本，其中只有一个副本的分区是leader分区，供消费者使用。如果某个broker宕机了，这个broker上的leader分区失效，需要在其他broker上重新进行选举。

Pulsar

跟Kafka不同的是，Pulsar的消息存储模型采用了分层的方式，如下图：



[2]

第一层是Topic，用来存储Producer追加的messages，Topic下面是ledger层，保存了分片(Segment)，分片里面保存更小粒度的ertries，entries存储一条条的Message。

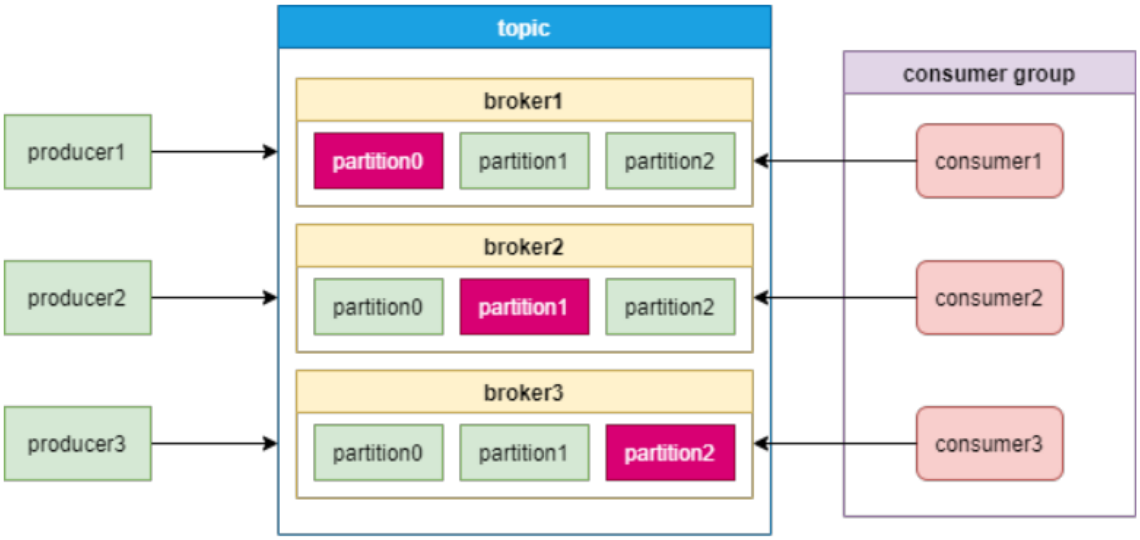
> Bookkeeper中，数据的最小操作单位是Segment。

Ledger中的最后一个分片是最新写入的分片，如上图Segment-2。Segment-2之前的所有分片已完成封装，这些分片的数据是不会再发生变化的。这样增加或删除一个BookKeeper节点，或者迁移长期存储节点，都不会发生一致性问题。

消息消费模型

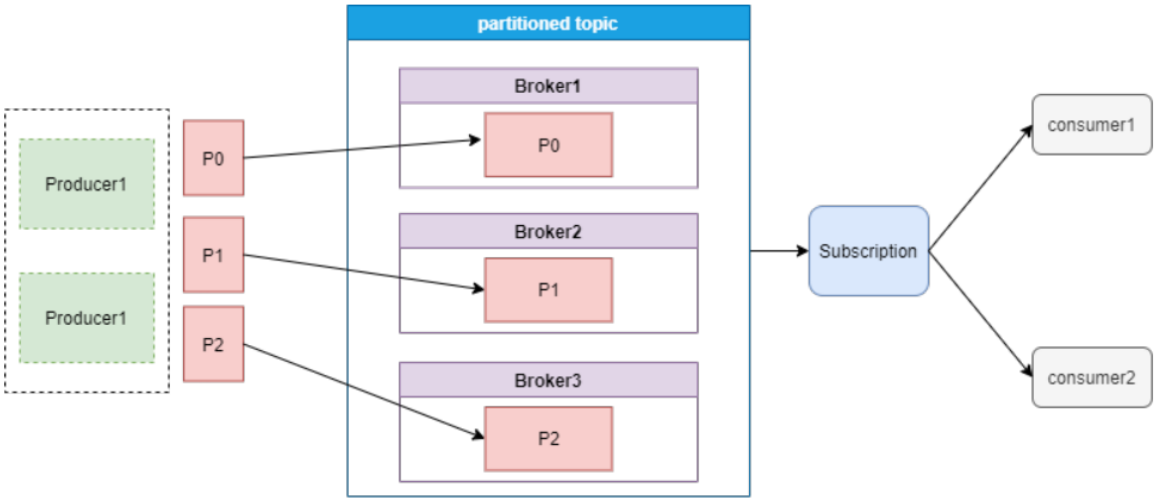
Kafka

Kafka的消费模型是采用消费者组的模式，每一个分区只能给消费者组中的一个消费者消费。如下图：



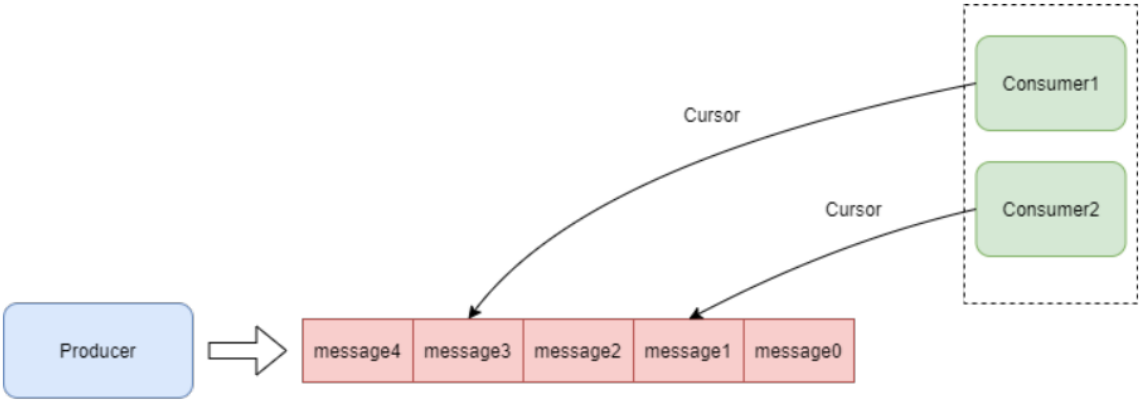
Pulsar

Pulsar的消费模型如下图：



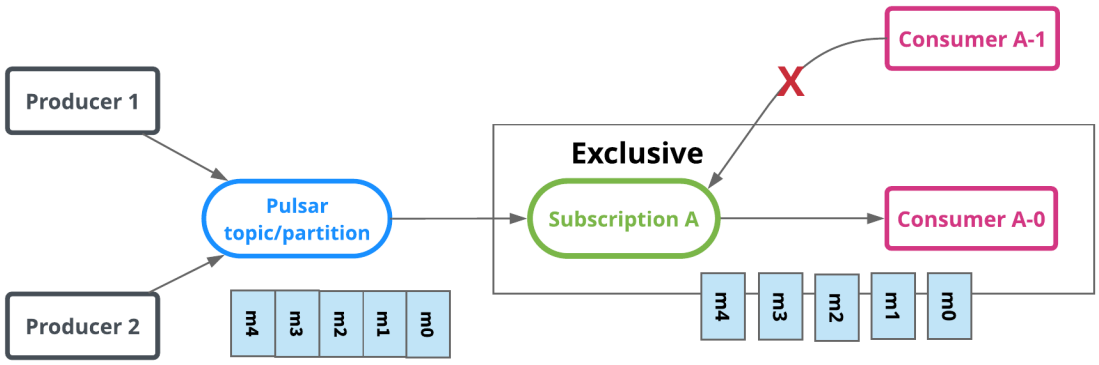
[3]Pulsar的topic是一种partitioned topic，可以被保存到多个broker，提高了topic的吞吐量。

Consumer通过Subscription获取消息，同一Topic的Subscription可以获取到Topic数据的完整拷贝，这样Subscription为每一个Consumer分配一个Cursor，Consumer之间互不影响。如下图：

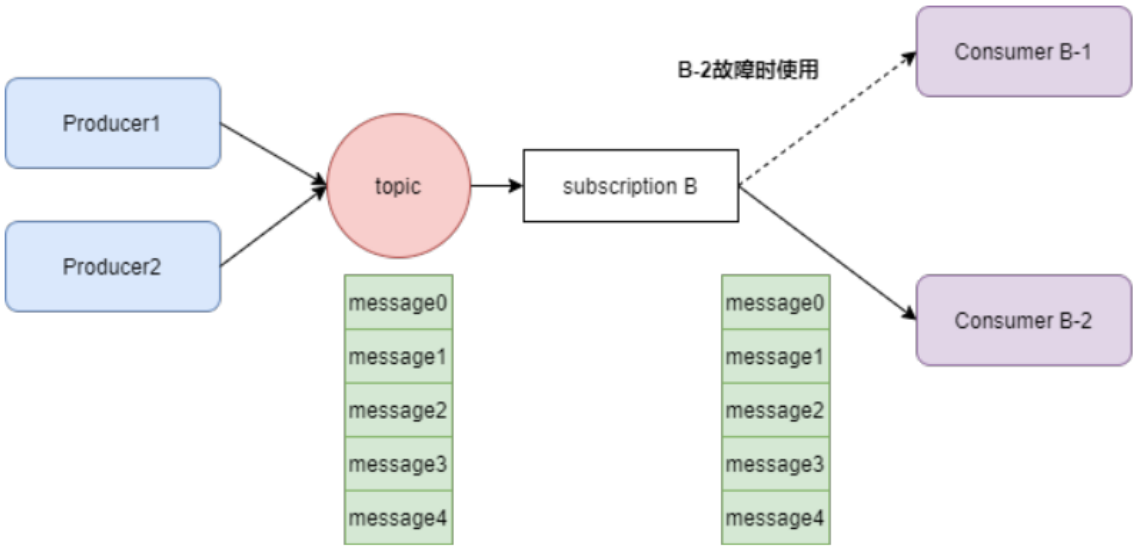


Pulsar的消费模型有4种：

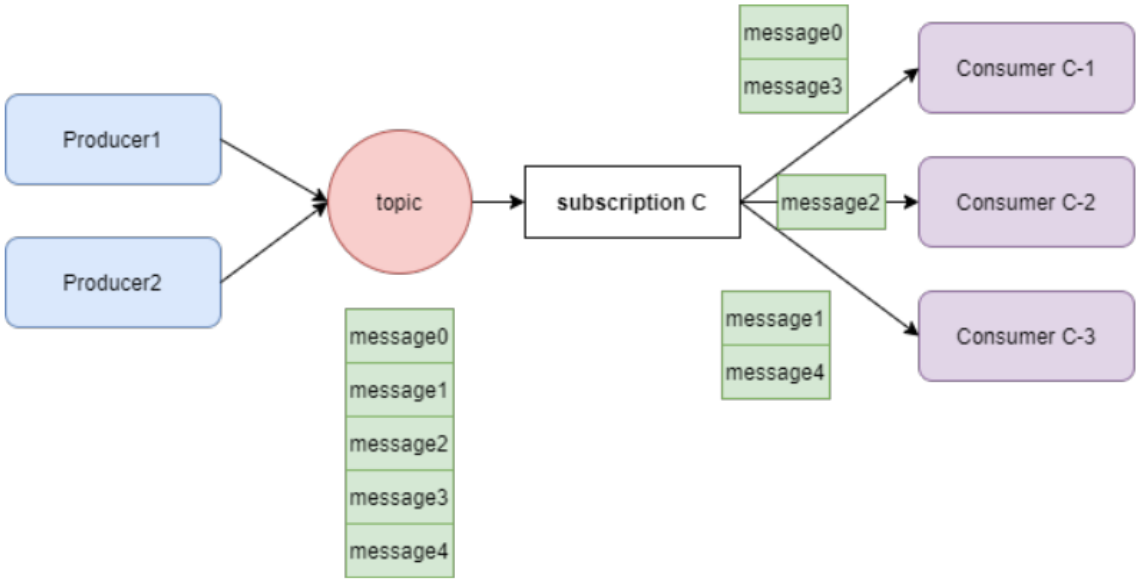
- 独占模式(Exclusive):同一个topic只能有一个消费者订阅，如果多个消费者订阅，就会出错。Exclusive模式为默认订阅模式。



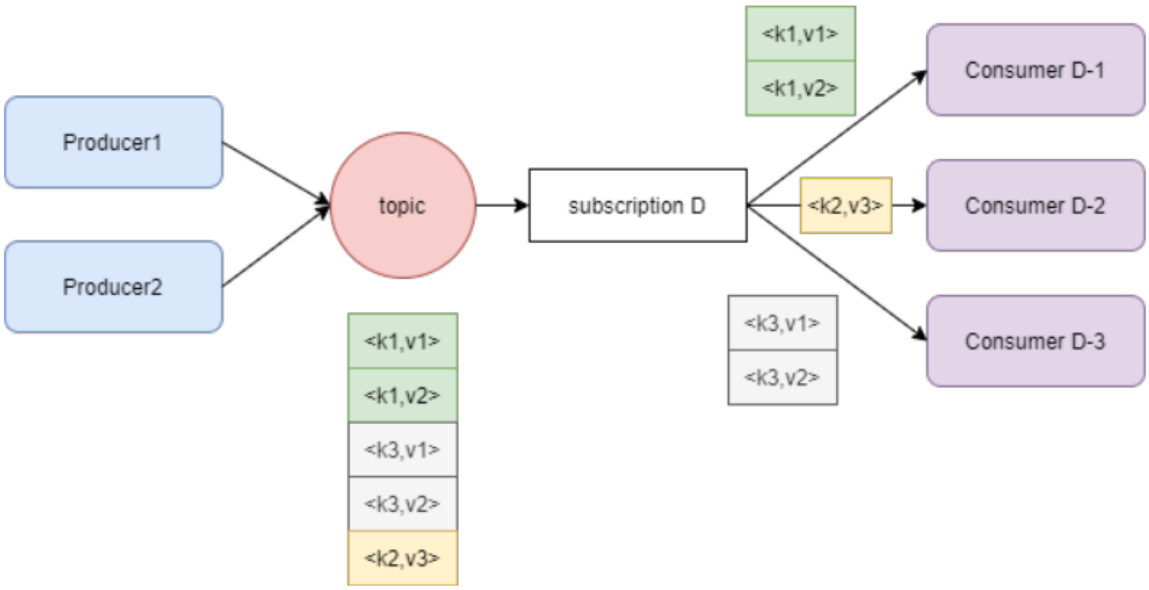
- 灾备模式(Failover)：同一个topic可以有多个消费者订阅，但是只能有一个消费者消费，其他订阅的消费者作为故障转移的消费者，只有当前消费者出了故障才可以进行消费当前的topic。如下图：



- 共享订阅(Shared):同一个topic可以由多个消费者订阅和消费。消息通过round robin轮询机制分发给不同的消费者，并且每个消息仅会被分发给一个消费者。当消费者断开，发送给它的没有被消费的消息还会被重新分发给其它存活的消费者。如下图：



- Key_Shared：消息和消费者都会绑定一个key，消息只会发送给绑定同一个key的消费者。如果有新消费者建立连接或者有消费者断开连接，就需要更新一些消息的key。如下图：



多租户

Pulsar

Pulsar是一个多租户系统，租户可以跨集群分布，每个租户都可以有单独的认证和授权机制。租户也是存储配额、消息 TTL 和隔离策略的管理单元。

Pulsar中topic的URL如下，可以看到租户是最基本的管理单位：

```
persistent://tenant/namespace/topic
```

上面的URL可以看到，Pulsar通过tenant和namespace来支持多租户。

> namespace是一个术语，指租户的管理单元。同一个namespace上设置的配置策略适用于在namespace中创建的所有 topic。

Pulsar为实例中的每个租户分配：

- 授权机制
- 适用于租户配置的集群配置

Kafka

Kafka为了控制客户端对broker资源的限制，从0.9版本引入了配额(quotas)管理，强制客户端请求使用配额。目前Kafka支持两种类型的配额：

- 网络带宽配额，用来定义byte-rate阈值(从0.9版本开始)
- 请求速率配额，将CPU利用率阈值定义为网络和I/O线程的百分比(从0.11开始)

> 生产者和消费者有可能以很高的速率生产和消费大量的请求，从而垄断broker资源，导致网络饱和，最终影响到其他客户端和broker本身。使用配额可以防止这些问题，让集群体验更好。

运维

集群部署

Kafka去除Zookeeper以后，部署是非常简单的。而Pulsar目前还没有去除Zookeeper的详细计划，而且需要使用到BookKeeper集群，部署复杂不少。

扩容

Pulsar支持自动负载均衡，这对于增加broker节点和增加存储节点都非常方便。

云原生支持

Pulsar 计算和存储节点分离，对云原生支持很好。

Kafka 多数组件也支持云原生。

替换broker

Pulsar的broker节点是无状态的，替换时不用考虑数据丢失。

社区

Pulsar社区发展非常迅速，StreamNative 还推出了StreamNative Hub来支持Pulsar社区建设。[4]

但Pulsar毕竟是一个新型的消息中间件，文档和社区都不太完善。在过去的一年多时间里，Pulsar在这方面做了很多的努力，包括举办全球峰会，创作视频和培训教程，邀请专业讲师进行培训。

使用Pulsar时，遇到的一些问题可能在网上找不到答案，需要查找源代码来解决。这对于中小公司来说，无疑增加了使用成本。

而Kafka作为非常成熟中间件，用户遇到的问题也非常多，新用户可以很方便地从网上找到答案。

Kafka与Pulsar异同：

	Kafka	**Pulsar**
概念	生产者 - 主题 - 消费者群体 - 消费者	生产者 - 主题 - 订阅 - 消费者
消费	更专注于分区上的流式传输、独占消息传递，没有共享消费。	统一消息传递模型和API。 通过独占的故障转移订阅进行流式传输通过共享订阅队列
Acking	简单的偏移offset管理 在Kafka 0.8之前，偏移量存储在ZooKeeper中在Kafka 0.8之后，偏移量存储在偏移主题上	统一消息传递模型和API。 通过独占的故障转移订阅进行流式传输通过共享订阅队列
Retention	根据保留删除消息，如果消费者在保留期之前未读取消息，则会丢失数据。	消息仅在所有订阅消耗后删除，即使订阅的消费者长时间down，也没有数据丢失。 即使所有订阅都使用消息，也允许消息保留一段配置的保留期。
TTL	没有TTL支持	支持消息TTL

Apache Pulsar将高性能流式处理（Apache Kafka所追求的）和灵活的传统队列（RabbitMQ所追求的）结合到一个统一的消息传递模型和API中，Pulsar使用统一的API提供一个流式处理和队列系统，具有相同的高性能。

总结

Pulsar作为新型的云原生分布式消息流平台，确实有很多优秀的设计理念。

在Yahoo内部支持应用服务平台中 140 万个topic，日处理消息超过 1000 亿条。腾讯的分布式交易引擎TDXA也使用了Pulsar，应用于腾讯的计费平台。[5]

kafka目前的使用场景最多的还是日志大数据处理，对金融场景的应用比较少。

但这并不能说明Pulsar可以取代Kafka，Kafka用户群体庞大，社区和资源完善，而且在2.8版本中去除了Zookeeper，部署非常容易。毕竟不是每家公司都需要Yahoo和腾讯这样的集群体量。

参考资料

[1]quotas: http://kafka.apache.org/documentation/#design_quotas

[2]pulsar: http://pulsar.apache.org/docs/zh-CN/next/concepts-messaging/

[3]segmentfault: https://segmentfault.com/a/1190000023605433

[4]StreamNative: https://www.cnblogs.com/StreamNative/p/14624039.html

[5]腾讯计费平台: https://cloud.tencent.com/developer/article/1492260