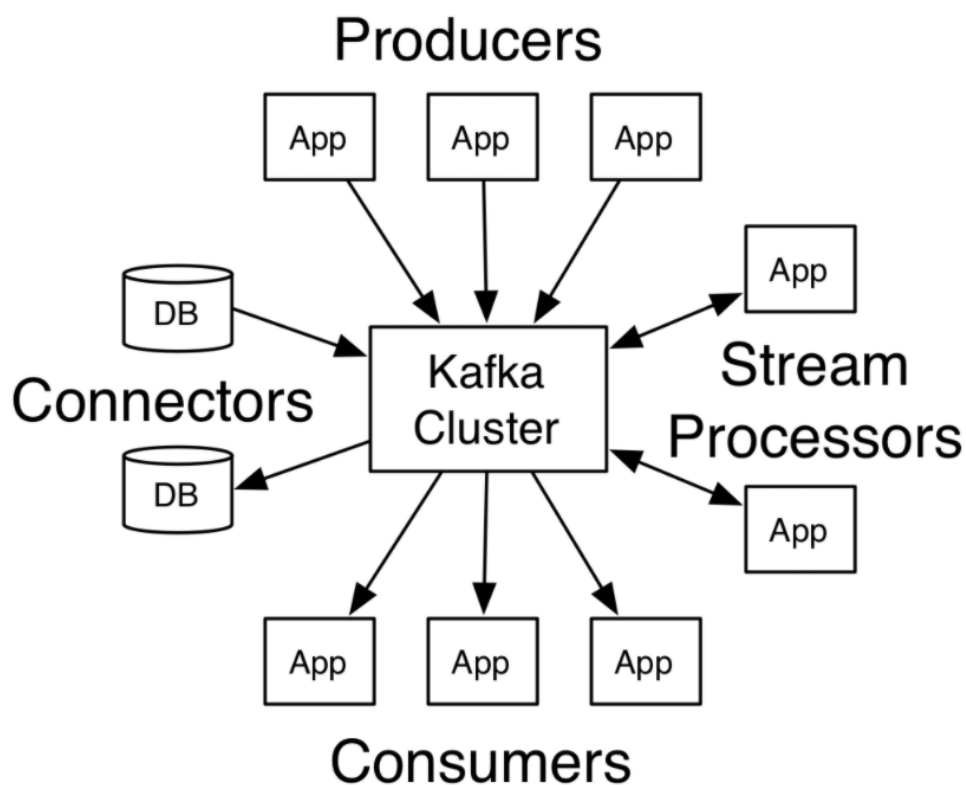


<https://kafka.apachecn.org/quickstart.html> 官方文档

<https://kafka.apachecn.org/intro.html> (知乎) 再过半小时，你就能明白kafka的工作原理了

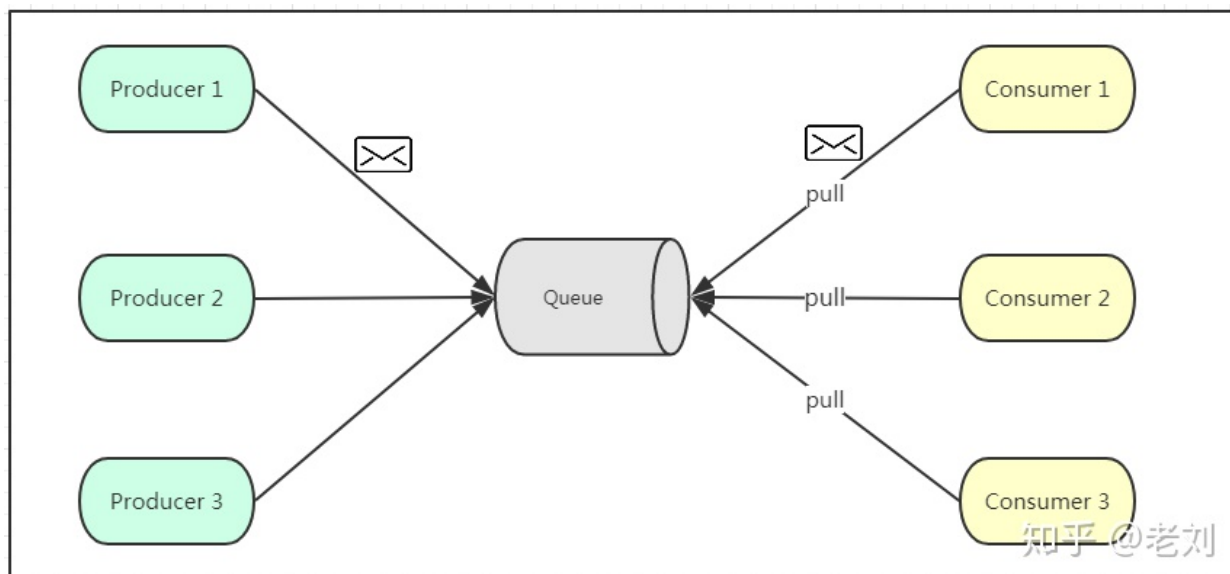
Apache Kafka® 是一个分布式流处理平台



- Kafka作为消息系统
- Kafka 作为存储系统
- Kafka用做流处理

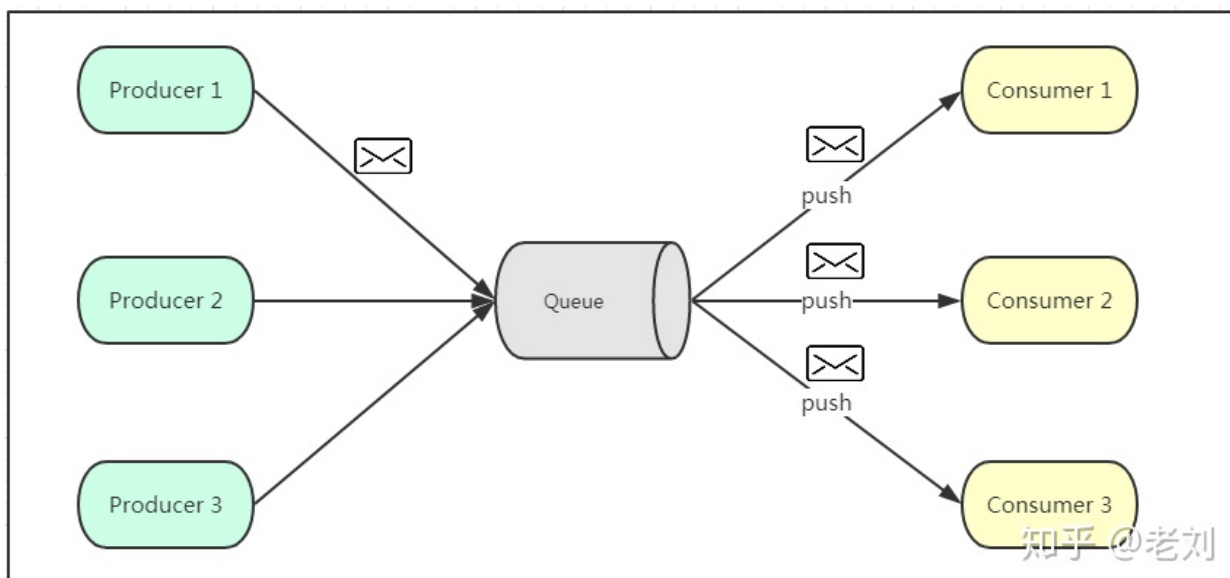
消息系统

点对点模式



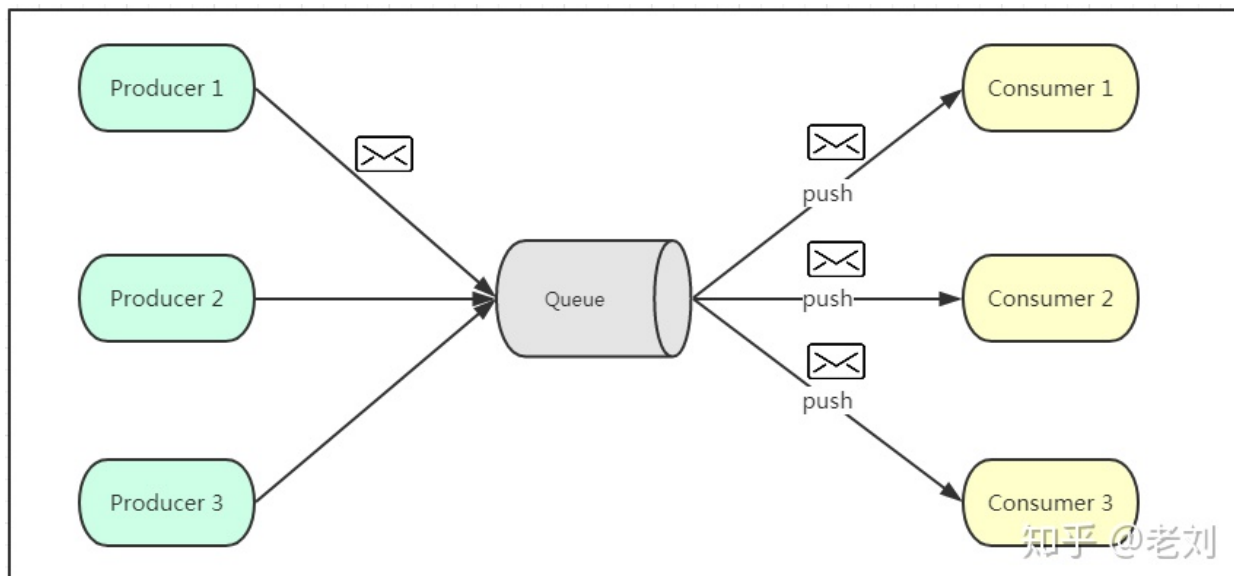
点对点模式通常是基于拉取或者轮询的消息传送模型，这个模型的特点是发送到队列的消息被一个且只有一个消费者进行处理。生产者将消息放入消息队列后，由消费者主动的去拉取消息进行消费。点对点模型的的优点是消费者拉取消息的频率可以由自己控制。但是消息队列是否有消息需要消费，在消费者端无法感知，所以在消费者端需要额外的线程去监控。

发布订阅模式



发布订阅模式是一个基于消息送的消息传送模型，改模型可以有多种不同的订阅者。生产者将消息放入消息队列后，队列会将消息推送给订阅过该类消息的消费者（类似微信公众号）。由于是消费者被动接收推送，所以无需感知消息队列是否有待消费的消息！但是consumer1、consumer2、consumer3由于机器性能不一样，所以处理消息的能力也会不一样，但消息队列却无法感知消费者消费的速度！所以推送的速度成了发布订阅模模式的一个问题！假设三个消费者处理速度分别是8M/s、5M/s、2M/s，如果队列推送的速度为5M/s，则consumer3无法承受！如果队列推送的速度为2M/s，则consumer1、consumer2会出现资源的极大浪费！

kafka术语



Producer: Producer即生产者，消息的产生者，是消息的入口。

Broker: Broker是kafka实例，每个服务器上有一个或多个kafka的实例，我们姑且认为每个broker对应一台服务器。每个kafka集群内的broker都有一个不重复的编号，如图中的broker-0、broker-1等.....

Topic: 消息的主题，可以理解为消息的分类，kafka的数据就保存在topic。在每个broker上都可以创建多个topic。

Partition: Topic的分区，每个topic可以有多个分区，分区的作用是做负载，提高kafka的吞吐量。同一个topic在不同的分区的数据是不重复的，partition的表现形式就是一个一个的文件夹！

Replication:每一个分区都有多个副本，副本的作用是做备胎。当主分区（Leader）故障的时候会选择一个备胎（Follower）上位，成为Leader。在kafka中默认副本的最大数量是10个，且副本的数量不能大于Broker的数量，follower和leader绝对是在不同的机器，同一机器对同一个分区也只能可能存放一个副本（包括自己）。

Message: 每一条发送的消息主体。

Consumer: 消费者，即消息的消费方，是消息的出口。

Consumer Group: 我们可以将多个消费组组成一个消费者组，在kafka的设计中同一个分区的数据只能被消费者组中的某一个消费者消费。同一个消费者组的消费者可以消费同一个topic的不同分区的数据，这也是为了提高kafka的吞吐量！

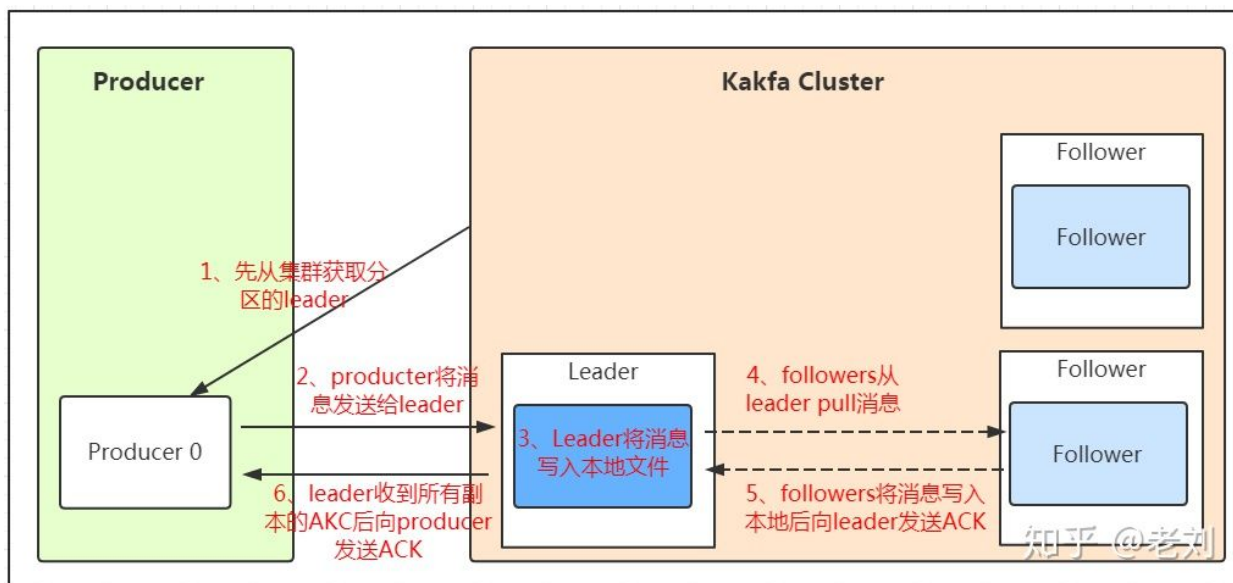
Zookeeper: kafka集群依赖zookeeper来保存集群的的元信息，来保证系统的可用性

kafka工作流程

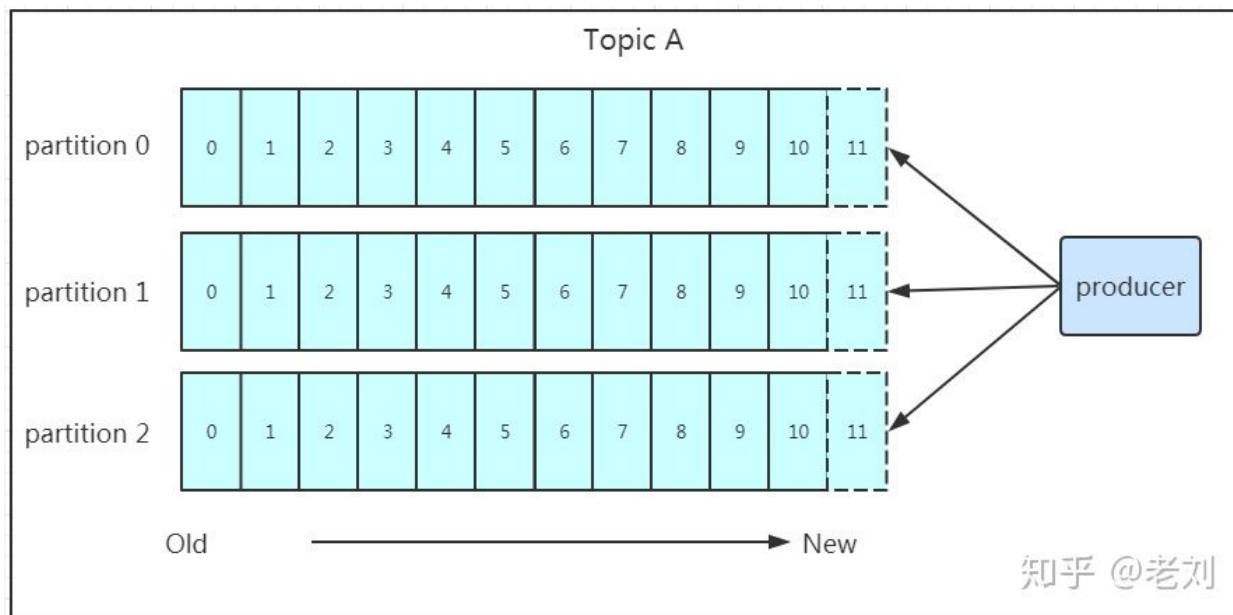
生产数据

producer就是生产者，是数据的入口。注意看图中的红色箭头，Producer在写入数据的时候永远的找leader，不会直接将数据写入follower！那leader怎么找呢？写入的流程又是什么样的呢？我们看

下图：



需要注意的一点是，消息写入leader后，follower是主动的去leader进行同步的！producer采用push模式将数据发布到broker，每条消息追加到分区中，顺序写入磁盘，所以保证同一分区内的数据是有序的！写入示意图如下



kafka为什么要做分区呢？

1、方便扩展。因为一个topic可以有多个partition，所以我们可以扩展机器去轻松的应对日益增长的数据量。

2、提高并发。以partition为读写单位，可以多个消费者同时消费数据，提高了消息的处理效率。

当我们向某个服务器发送请求的时候，服务端可能会对请求做一个负载，将流量分发到不同的服务器，那在kafka中，如果某个topic有多个partition，producer又怎么知道该将数据发往哪个partition呢？kafka中有几个原则：

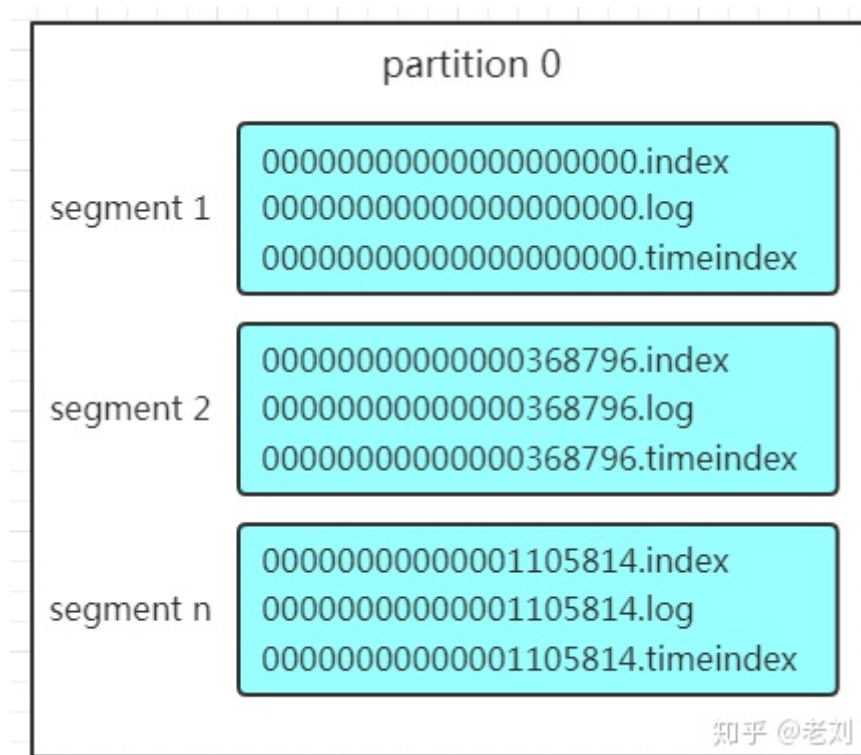
保存数据

kafka将数据保存在磁盘，每个topic都可以分为一个或多个partition，如果你觉得topic比较抽象，那partition就是比较具体的东西了！Partition在服务器上的表现形式就是一个一个的文件夹，每个partition的文件夹下面会有多组segment文件，每组segment文件又包含.index文件、.log文件、.timeindex文件（早期版本中没有）三个文件，log文件就实际是存储message的地方，而index和timeindex文件为索引文件，用于检索消息。

- 存储策略

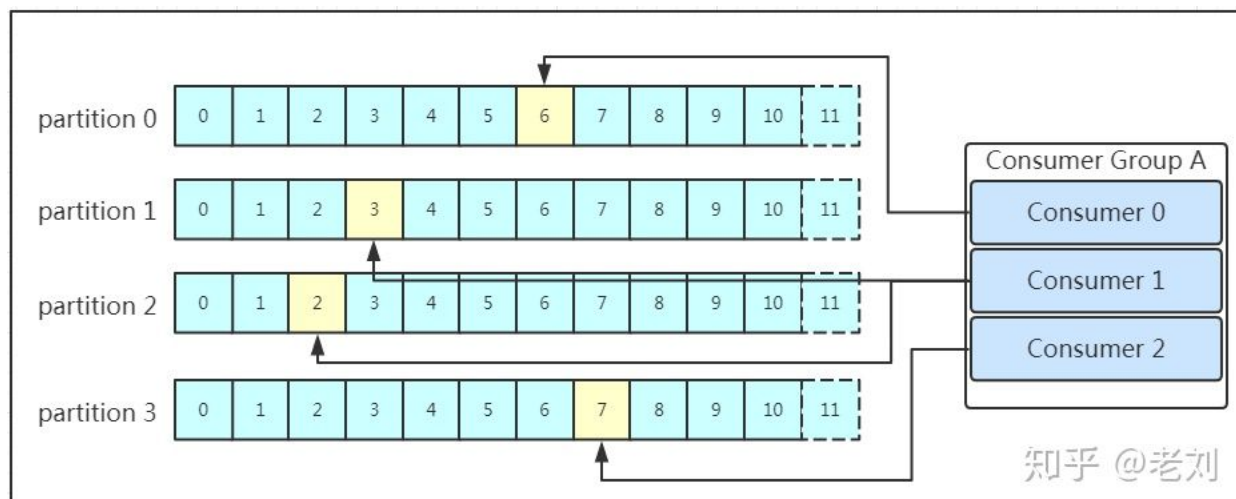
无论消息是否被消费，kafka都会保存所有的消息。那对于旧数据有什么删除策略呢？

- 1、基于时间，默认配置是168小时（7天）。
- 2、基于大小，默认配置是1073741824。



消费数据

消息存储在log文件后，消费者就可以进行消费了。在讲消息队列通信的两种模式的时候讲到过点对点模式和发布订阅模式。Kafka采用的是点对点的模式，消费者主动的去kafka集群拉取消息，与producer相同的是，消费者在拉取消息的时候也是找leader去拉取。



图示是消费者组内的消费者小于partition数量的情况，所以会出现某个消费者消费多个partition数据的情况，消费的速度也就不及只处理一个partition的消费者的处理速度！如果是消费者组的消费者多于partition的数量，那会不会出现多个消费者消费同一个partition的数据呢？上面已经提到过不会出现这种情况！多出来的消费者不消费任何partition的数据。所以在实际的应用中，建议消费者组的consumer的数量与partition的数量一致！

这套机制是建立在offset为有序的基础上，利用segment+有序offset+稀疏索引+二分查找+顺序查找等多种手段来高效的查找数据！至此，消费者就能拿到需要处理的数据进行处理了。那每个消费者又是怎么记录自己消费的位置呢？在早期的版本中，消费者将消费到的offset维护zookeeper中，consumer每间隔一段时间上报一次，这里容易导致重复消费，且性能不好！在新的版本中消费者消费到的offset已经直接维护在kafka集群的_consumer_offsets这个topic中！

后台启动zookeeper

```
./zookeeper-server-start.sh -daemon ../config/zookeeper.properties
```

后台启动kafka

```
./kafka-server-start.sh -daemon ../config/server.properties
```