<http://www.cnblogs.com/cjsblog/p/9664536.html> 这篇文章很有参考价值，基本讲了kafka核心的部分

Q:broker,producer,consumer,partition，offset

A:broker一个kafka服务，很多时候也是夜歌节点，producer是生产者，consumer是消费者，典型消息队列概念，partition分区，是发送和消费的基本单元，是kafka高性能负载均衡的核心，offset，kafka并没有采用发送-消费-删除的范式，而是采用一个消息队列，通过偏移量去消费，消费后以consumer group的方式修改偏移量而不需要删除消费过的消息，便于容错和针对需要重复消费的业务。

Q：zookeeper与kafka启动顺序

A:zookeeper要先于kafka启动，因为卡夫ka服务依赖zk

Q：lsof - i：port

A:通过端口查看进程的常用方式

Q:kafka是如何实现一根分组内只有一个人可以消费，不同组之前是广播的

A:通过分区负载均衡，保证一个消费组里只有一个消费者绑定一个分区，并且偏移量的记录也是以group形式记录，每个group各自维护自己的偏移量，不冲突，自然就实现了不同group广播，因为偏移量不共享

Q：kafka有哪些特性（以后也可以从这些角度去思考一个框架，中间件）

A:分布式，支持分区 便于横向扩展，冗余备份

Q：kafka是如何实现这些特性的

A:分布式式其实是分区和副本的分开，分区和副本也是尽量分布在不同的机器上

Q：分区与分区之间的关系，或者说partition是副本么，还是每个分区都是一部分，比如这个topic有100条数据，两个分区，每个分区50条，还是每个分区100条

A：这么说吧，可以看producer的api，你会发现可以指定分区，这就说明分区之间并不是副本关系，那么很有可能是个50，那么kafka的冗余备份是怎么实现的呢？在这里是有一个分区策略的：

1. 看你有没有指定分区，如果指定了就发到指定分区，在这里肯定也会有api提供给开发人员所有的分区
2. key，producer是可以传递key参数的，他会根据key的hash值落在某一分区上
3. 如果key也没有，分区也没有指定就会采用轮询的方式的选择一个分区。

Q：那么，消费者是如何准确的消费到消息的呢？会不会出现，producer发到了p1.但consumer去p2或者p3取数据，这明显取不到呀？

A：首先消费者是可以收到一个topic所有分区的消息的，而且是必须，否则消息肯定会漏掉，在消费者和消息队列之间有一个消息协调器和消息流，消息流会把topic的消息全部放在里面，消费者可以迭代式访问消息流以获取信息。对于low level api还会先去查询消息在哪个分区和以及偏移量再去取消息，所以消息是不会漏掉的

Q：就当你说的分布不是冗余备份的，而是分段的，那么一个消费组中有两个消费者，一个topic也有两个分区，如果分区是一样的，那么那个被分配到后50条怎么办，按照业务肯定是全部呀，第二个消费岂不是只能读取？这不对呀？

A:这里有明显的问题，这里存在消息重复消费的场景。我可以提供两种都可以从头开始的读的解决方案，第一使用广播方案，第二指定分区

A：首先需要讲下分区与消费者是如何分配的，同一时刻，一条消息只能被组中的一个消费者实例消费，消费者组订阅这个主题，意味着主题下所有分区都会被组中的消费者消费到

**Q：****一个分区被多个消费者消费会出现什么问题？**

K：只有一个可以读到，另一个读不到

**Q：kakfa的冗余备份是如何实现的**

**A:副本机制，消息持久化**

Q：按照你说的，一个消费者怎么可以订阅多个主题呢？还是说一个消费者本来就可以订阅多个主题？

A：不错，一个消费者其实可以订阅多个topic

**Q：不管是采用哪种分配策略，都会存在分区不完全分配给一个消费者情况，那么如果我需要的数据刚好在那个没分配给我的的分区怎么办？**

**A:在上面我已经说了，会访问所有的分区，保证拿到数据**

**Q:一个分区只有一个消费者，是针对组么？如果在广播情况下，明显有多个消费者使用一个分区。难道广播的时候？**

**K:可以确定的是在正式消费的流程中都是要先进行配**

**Q：kafka分区深度，他是无限扩展的？还是一个分区走完，就到另一个分区去了？**

**Q：kafka reblance是什么**

**A:消费者创建删除修改，或者新的消费者加入离开，相应消费者节点发生改变都会发生rebalance**

**Q：topic保存在哪里文件；**

**A:消息在log.dirs目录下，元数据存在zk中**

**A：server.properties文件里的log.dirs**

**Q:producer发送消息是同步还是异步的?**

**A:既可以同步也可以异步**

**Q:Consumer消费过程及源码剖析**



