# Kafka的基本原理

1. **Kafka是什么**

Kafka是一种高吞吐量的分布式发布订阅消息系统，有如下特性：

通过磁盘数据结构提供消息的持久化，这种结构对于即使数以TB的消息存储也能够保持长时间的稳定性能。

高吞吐量：即使是非常普通的硬件Kafka也可以支持每秒数百万的消息

支持通过Kafka服务器和消费机集群来分区消息。

支持Hadoop并行数据加载。

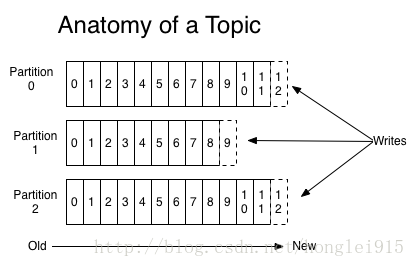
1. **Kafka的架构和组件**

Topic：话题，是特定类型的消息流。消息是字节的有效负载（Payload），话题是消息的分类名或种子（Feed）名。

Partition：分区，1个partition只能被同组的一个consumer消费，同组的consumer起到均衡效果，默认的分区数量可以通过server.property指定，允许数据能够扩展到更多的服务器上去。每个单独的分区的大小受到承载它的服务器的限制，但一个topic可能有很多分区，以便它能够支持海量的数据，更重要的意义是分区是进行并行处理的基础单元



每个分区都会配置一定数量的副本分区提供容错能力，每个分区都有一个充当“leader”，零个或者多个充当follower，follower被动的从leader中拷贝数据，只有当leader失败了，follower才会被启用（每个broker可能充当一些分区的leader和其他分区的follower，这样负载就会比较好的均衡）

针对每个topic，Kafka对它的日志进行了分区，如下图所示：，

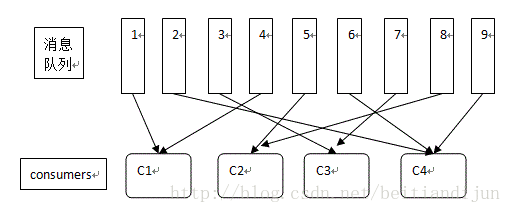
每个分区都由一系列有序的、不可变的消息组成，这些消息被连续的追加到分区中。分区的每个消息都由一个连续的序列号叫做offset（偏移量），这个offset在分区内是唯一的，这个偏移量由consumer控制，通常偏移量会随着consumer读取记录线性前进，但事实上，因为位置是由消费者进行控制的，消费者可以在任何位置读取记录

Producer：发布消息到话题的任何对象。Producer负责选择把记录分配到主题中的哪个分区，处于负载均衡的考虑，既可以通过循环遍历的方式分发消息，也可以通过执行基于某些分发函数来分发消息（即基于消息中的某些关键字）

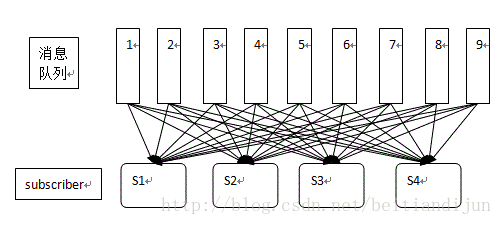
Consumer：以消费群（Consumer Group）来标识自己，每个发布到主题的消息都会发送给订阅了这个主题的消费群里面的一个消费者的一个实例，消费者的实例可以在单独的进程或单独的机器上



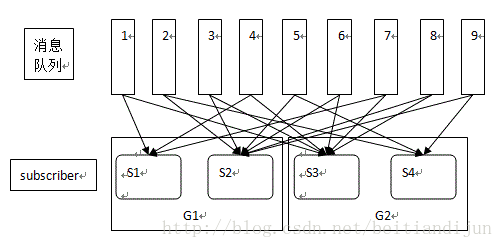
如果所有的消费者实例是同一个消费组，那么所有的消费记录实际上被所有的消费者实例均衡负载-------------------------消费模式



如果所有的消费者都在不同的消费组中，那么每条记录都会被广播到所有的消费进程中-------------------------------------------订阅模式

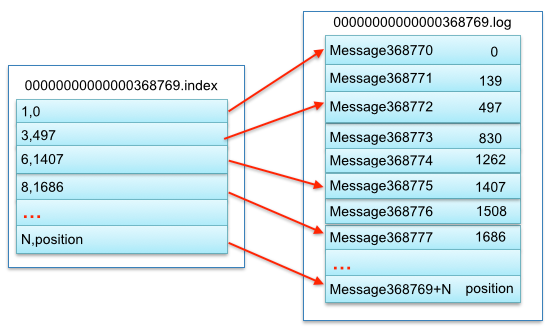


此外，kafka还提供一种consumer-group模式，具有以上两种的共同点



Broker：已发布的消息保存在一组服务器中，它们被称为代理（Broker）或Kafka集群。Broker的数量在Kafka集群启动的时候在server.property配置文件中，已经指定，一般有几台服务器就有几个broker，但是也可以在一个服务器上配置多个broker（配置多个server.property，然后启动）

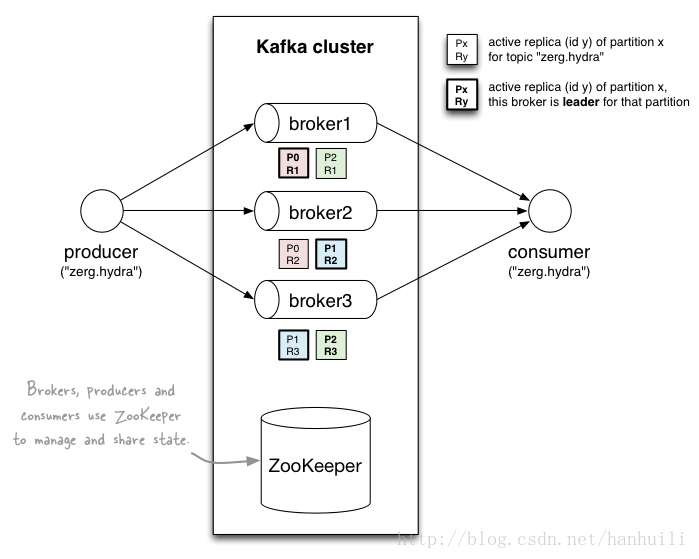
Kafka的消息存储机制：Kafka的消息是存储在磁盘中的，这样的话，数据不容易丢失，每个topic会被hash到不同的partition中，每个partition都会保存成一个文件，这个文件包含两部分：.index索引文件、.log消息内容文件



1. **Kafka的调用图解**



如上图所示：producer将消息以topic的形式向kafka集群发送消息（push方式），consumer从kafka集群中拉取指定的topic消息进行消费（pull）。



如上图所示：producer向broker中发送topic，topic会将消息hash到三个不同的partition上，并且将数据备份到对应的备份上，然后zookeeper监听broker的健康状况，如果有发生broker挂掉的情况，会重新选择其他broker上的follower作为leader来代替本来在此broker上的partition的leader，保证数据的完整性，然后consumer从partition中拉取指定topic的内容

1. **简单Kafka的使用（集群模式）**
2. **、启动Kafka集群**

在不同的节点上分别执行bin/kafka-server-start.sh config/server.properties &

从zookeeper的客户端上，查看broker的ids是否在zookeeper上注册

1. **、创建topic**

bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 3 --partitions 1 --topic my-topic

1. **、查看topic**

bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic my-topic

1. **、发布消息到指定的topic中**

bin/kafka-console-producer.sh --broker-list localhost:9092 --topic my-replicated-topic

运行之后你可以在控制台输入需要输入的内容：helloworld！！

1. **、consumer消费指定的topic**

bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --from-beginning --topic my-topic

运行之后会在控制台输出topic里面的内容：helloworld！！

Kafka的API使用





