## 12.1.1. Kafka概念

Kafka是一种高吞吐量、分布式、基于发布/订阅的消息系统，最初由LinkedIn公司开发，使用Scala语言编写，目前是Apache的开源项目。

|  |  |
| --- | --- |
| broker | Kafka服务器，负责消息存储和转发 |
| topic | 消息类别，Kafka按照topic来分类消息 |
| partition | topic的分区，一个topic可以包含多个partition，topic消息保存在各个partition上 |
| offset | 消息在日志中的位置，可以理解是消息在partition上的偏移量，也是代表该消息的唯一序号 |
| Producer | 消息生产者 |
| Consumer | 消息消费者 |
| Consumer Group | 消费者分组，每个Consumer必须属于一个group |
| Zookeeper | 保存着集群broker、topic、partition等meta数据；另外，还负责broker故障发现，partition leader选举，负载均衡等功能 |
|  | |

## 12.1.2. Kafka数据存储设计

### 12.1.2.1. partition的数据文件（offset，MessageSize，data）

partition中的每条Message包含了以下三个属性：offset，MessageSize，data，其中offset表示Message在这个partition中的偏移量，offset不是该Message在partition数据文件中的实际存储位置，而是逻辑上一个值，它唯一确定了partition中的一条Message，可以认为offset是partition中Message的id；MessageSize表示消息内容data的大小；data为Message的具体内容。

### 12.1.2.2. 数据文件分段segment（顺序读写、分段命令、二分查找）

partition物理上由多个segment文件组成，每个segment大小相等，顺序读写。每个segment数据文件以该段中最小的offset命名，文件扩展名为.log。这样在查找指定offset的Message的时候，用二分查找就可以定位到该Message在哪个segment数据文件中。

### 12.1.2.3. 数据文件索引（分段索引、稀疏存储）

Kafka为每个分段后的数据文件建立了索引文件，文件名与数据文件的名字是一样的，只是文件扩展名为.index。index文件中并没有为数据文件中的每条Message建立索引，而是采用了稀疏存储的方式，每隔一定字节的数据建立一条索引。这样避免了索引文件占用过多的空间，从而可以将索引文件保留在内存中。



## 12.1.3. 生产者设计

### 12.1.3.1. 负载均衡（partition会均衡分布到不同broker上）

由于消息topic由多个partition组成，且partition会均衡分布到不同broker上，因此，为了有效利用broker集群的性能，提高消息的吞吐量，producer可以通过随机或者hash等方式，将消息平均发送到多个partition上，以实现负载均衡。 

### 12.1.3.2. 批量发送

批量发送是提高消息吞吐量重要的方式，Producer端可以在内存中合并多条消息后，以一次请求的方式发送了批量的消息给broker，从而大大减少broker存储消息的IO操作次数。但也一定程度上影响了消息的实时性，相当于以时延代价，换取更好的吞吐量。

### 12.1.3.3. 压缩（GZIP或Snappy）

Producer端可以通过GZIP或Snappy格式对消息集合进行压缩。Producer端进行压缩之后，在Consumer端需进行解压。压缩的好处就是减少传输的数据量，减轻对网络传输的压力，在对大数据处理上，瓶颈往往体现在网络上而不是CPU（压缩和解压会耗掉部分CPU资源）。

## 12.1.4. 消费者设计



### 12.1.4.1. Consumer Group

同一Consumer Group中的多个Consumer实例，不同时消费同一个partition，等效于队列模式。partition内消息是有序的，Consumer通过pull方式消费消息。Kafka不删除已消费的消息，对于partition，顺序读写磁盘数据，以时间复杂度O(1)方式提供消息持久化能力。