

# 【Kafka零基础学习】图解Kafka Consumer整体流程

特别说明：本文基于Kafka 0.10.0.1。

## 一、基本概念

Kafka是一种分布式的、基于发布/订阅的消息系统。它具备快速、可扩展、可持久化的特点。它涉及到消息系统中的一些常见概念，生产者（图中Producer），消费者（图中Consumer），消息中间件集群（图中Kafka broker Cluster），主题（topic）。然而Kafka中还有一些独特的概念，比如消费组（图中Consumer Group），分区（图中Partition）。partition是topic中数据的物理分区，用不同文件夹表示。

从图中可以看出Consumer和ConsumerGroup之间的关系，即一个Consumer对应一个ConsumerGroup，一个ConsumerGroup包含多个Consumer。另外，一个ConsumerGroup中的不同Consumer只能消费不同的Partition，跨ConsumerGroup中的Consumer则没有这个限制，即ConsumerA1和ConsumerB1可以消费同一个Partition。

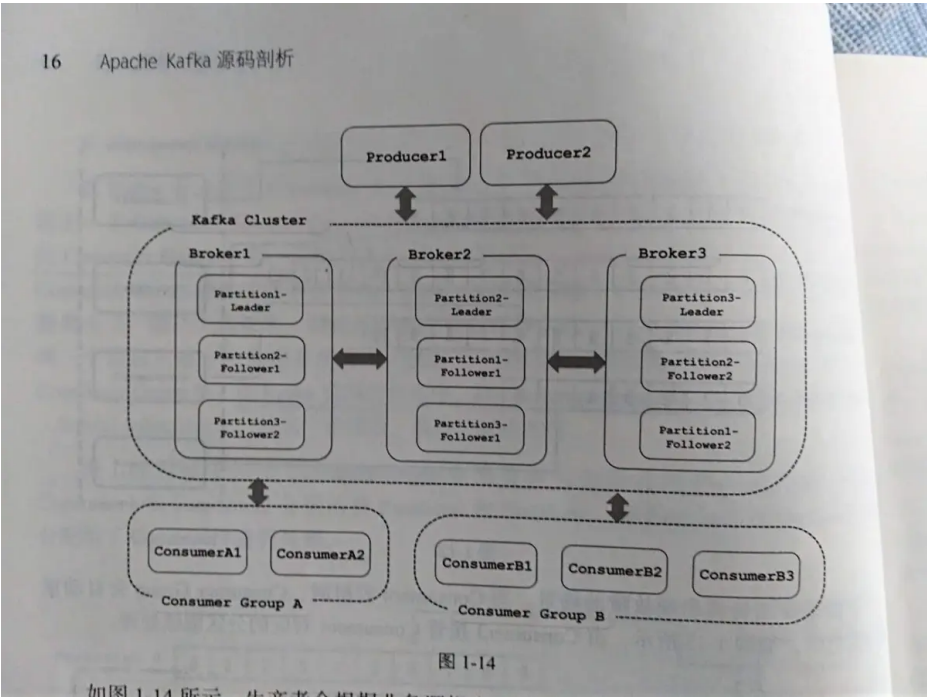


图 1-14

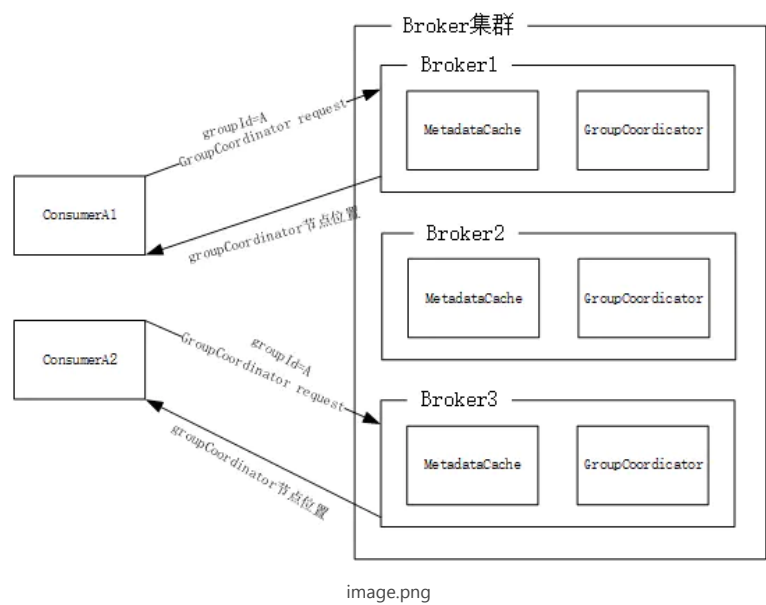
## 二、Consumer向Broker发送消息的完整流程（从启动到消费）

Consumer只能消费topic中某个partition的数据（其中topic个数可为多个）。为了获取分配的partition，Consumer需要经过下面的几个步骤。

### 第一步：Consumer发送GroupCoordinator请求获取GroupCoordinator所在节点。

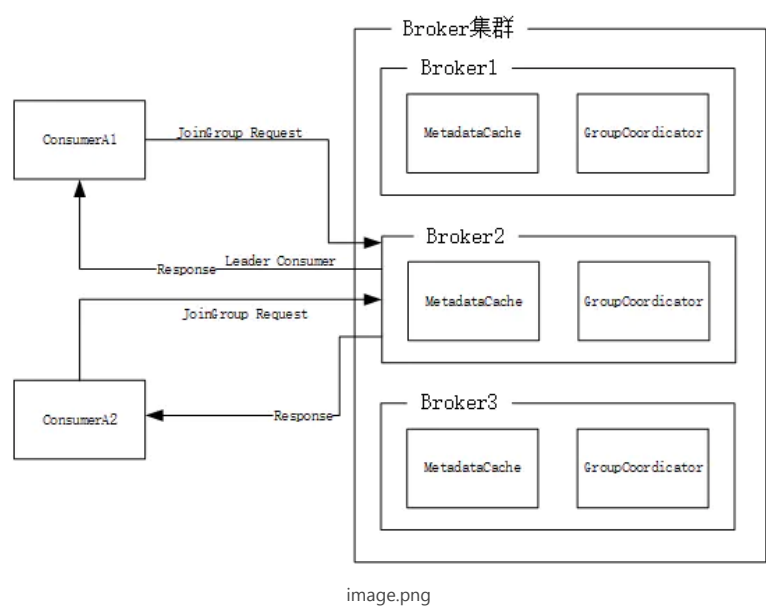
如下图，ConsumerA1向Broker1请求GroupCoordinator节点，ConsumerA2向Broker3请求GroupCoordinator节点，Broker集群需要保证Broker1和Broker3分配的是同一个

GroupCoordinator（注意一个Broker对应一个GroupCoordinator），这里是通过MedataCache来保证。MedataCache数据在集群中是一致的。



## 第二步：Consumer发送JoinGroup请求加入GroupCoordinator。

假设在第一步中分配结果为Broker2中的GroupCoordinator。此时ConsumerA1和ConsumerA2将分别提交JoinGroup请求加入GroupCoordinator。JoinGroup请求包含消费者的订阅信息、支持的分区分配器、和自定义数据。Broker2中GroupCoordinator收到请求后，会从中选取一个Consumer做为Leader，还会选取一个支持所有Consumer的分区分配策略，然后以JoinGroupResponse返回。图中ConsumerA1被选取作为了Leader。虽然每个消费者都会收到JoinGroupResponse，但是只有 Leader Consumer收到的JoinGroupResponse中封装了其他消费者的信息。当消费者确定自己是Leader后，会根据消费者的信息以及选定的分区分配策略进行分区分配。



## 第三步：Consumer发送SyncGroup请求获取分区分配。

如图所示，ConsumerA1和ConsumerA2各自发送SyncGroup请求去获取分区分配结果。它们的分配结果是在上一步由ConsumerA1分配的。

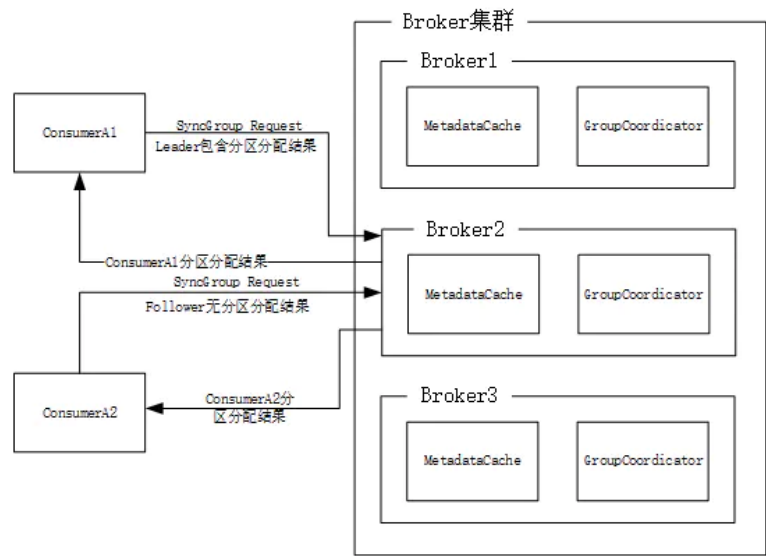


image.png

第四步：Consumer发送OffsetFetch请求获取分区消费位置。

假设第三步的分配结果为ConsumerA1对应Partition1-Leader，ConsumerA2对应Partition2-Leader。因为ConsumerA1和ConsumerA2从未消费过Partition数据，所以返回的offset为-1。

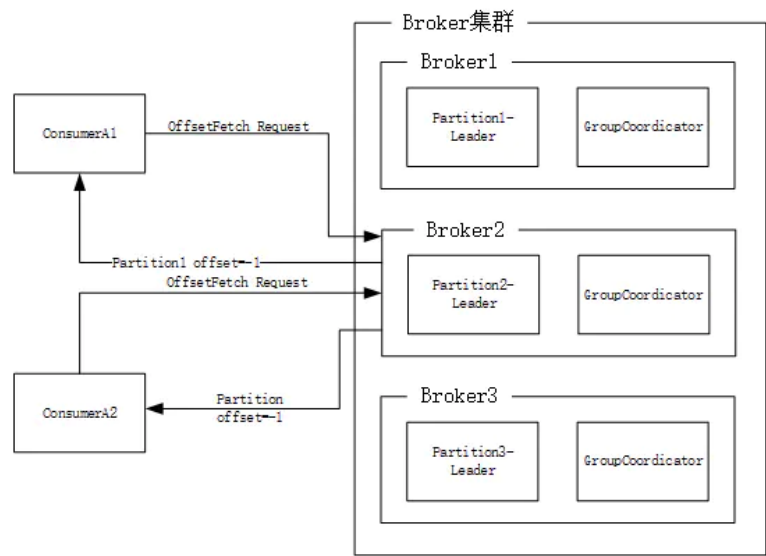
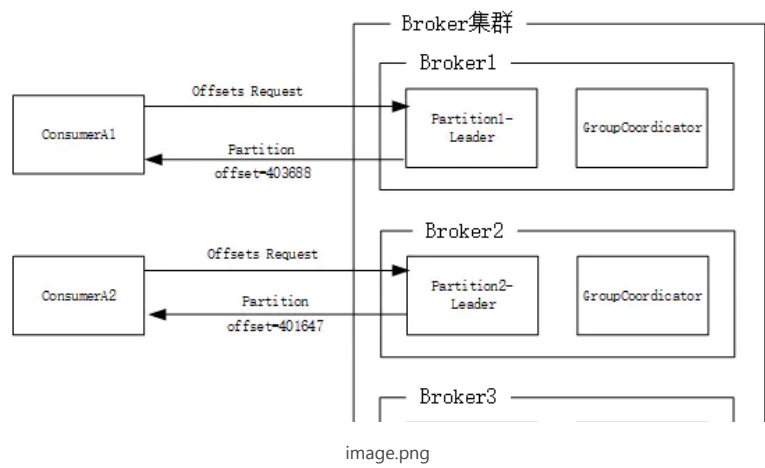


image.png

第五步：Consumer发送OffsetsRequest请求获取分区消费位置。

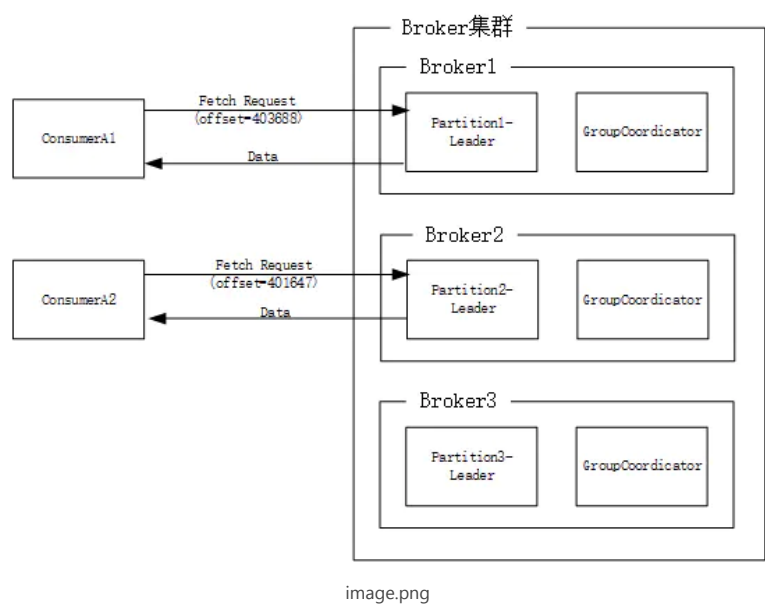
因为第四步未得到ConsumerA1和ConsumerA2的消费offset，所以ConsumerA1和ConsumerA2需要发送OffsetsRequest到各自的leader partition所在的broker以至获取起始offset。





第六步：Consumer发送Fetch请求获取数据。

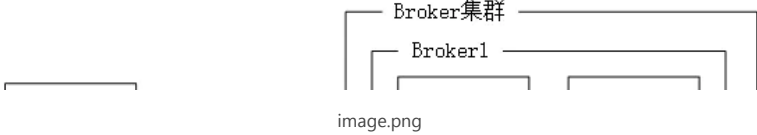
ConsumerA1和ConsumerA2根据上一步获取的offset从各自的Partition Leader中获取数据。



第七步：Consumer发送OffsetCommit请求到GroupCoordinator保存消费offset。

保存offset的目的是为了Consumer可以接着上次的消费offset继续消费（比比如Consumer重启）。





第八步：Consumer发送Heartbeat请求到GroupCoordinator。

Consumer发送心跳请求到GroupCoordinator。若GroupCoordinator在一个时间段内未检查到Consumer的心跳请求，则GroupCoordinator会要求ConsumerGroup中Consumer的Reblance，进而重新从第二步开始运行。

