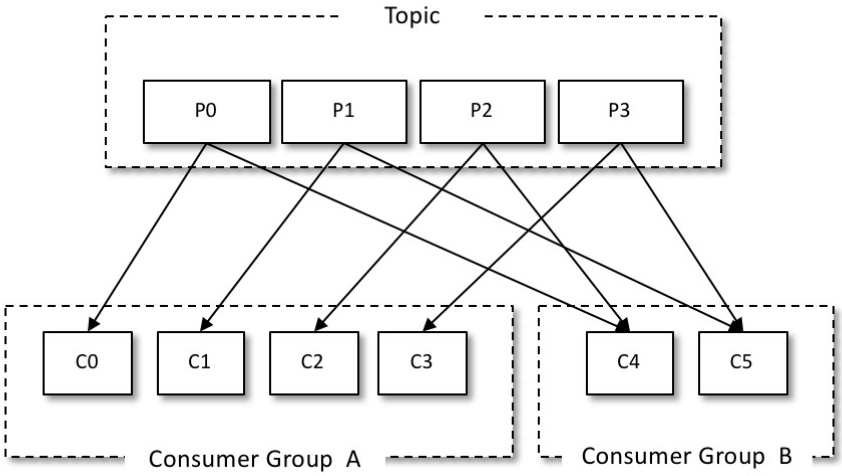
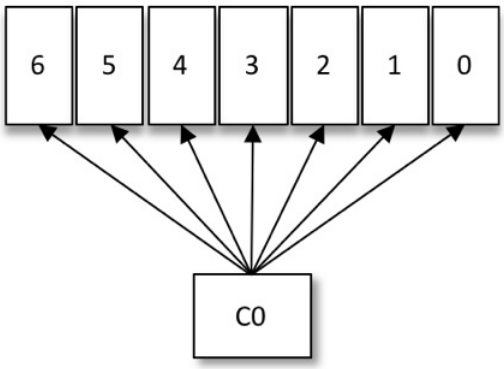
消费者与消费组

与生产者对应的是消费者，应用程序可以通过 KafkaConsumer 来订阅主题，并从订阅的主题中拉取消息。不过在使用 KafkaConsumer 消费消息之前需要先了解消费者和消费组的概念，否则无法理解如何使用 KafkaConsumer。本章首先讲解消费者与消费组之间的关系，进而再细致地讲解如何使用 KafkaConsumer。

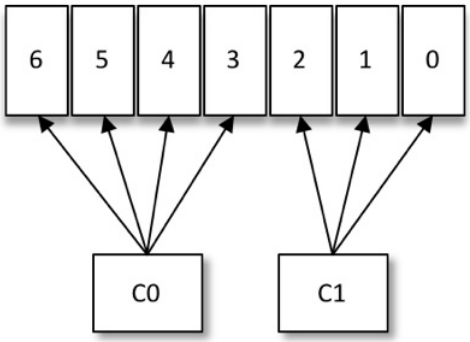
消费者（Consumer）负责订阅 Kafka 中的主题（Topic），并且从订阅的主题上拉取消息。与其他一些消息中间件不同的是：在 Kafka 的消费理念中还有一层消费组（Consumer Group）的概念，每个消费者都有一个对应的消费组。当消息发布到主题后，只会被投递给订阅它的每个消费组中的一个消费者。



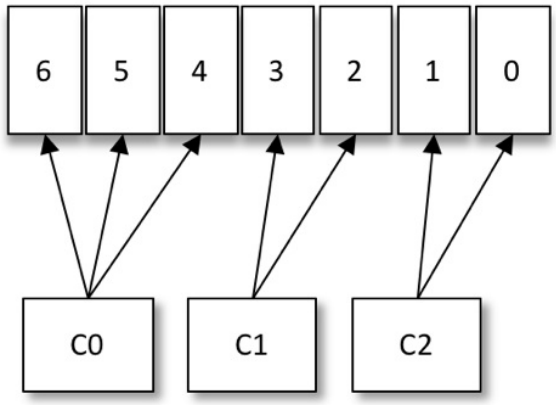
如上图所示，某个主题中共有4个分区（Partition）：P0、P1、P2、P3。有两个消费组A和B都订阅了这个主题，消费组A中有4个消费者（C0、C1、C2和C3），消费组B中有2个消费者（C4和C5）。按照 Kafka 默认的规则，最后的分配结果是消费组A中的每一个消费者分配到1个分区，消费组B中的每一个消费者分配到2个分区，两个消费组之间互不影响。每个消费者只能消费所分配到的分区中的消息。换言之，每一个分区只能被一个消费组中的一个消费者所消费。



我们再来看一下消费组内的消费者个数变化时所对应的分区分配的演变。假设目前某消费组内只有一个消费者C0，订阅了一个主题，这个主题包含7个分区：P0、P1、P2、P3、P4、P5、P6。也就是说，这个消费者C0订阅了7个分区，具体分配情形参考上图。

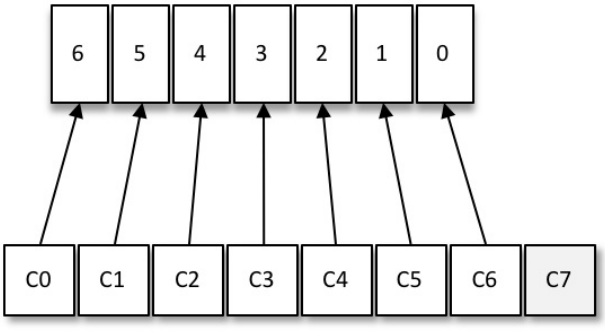


此时消费组内又加入了一个新的消费者C1，按照既定的逻辑，需要将原来消费者C0的部分分区分配给消费者C1消费，如上图所示。消费者C0和C1各自负责消费所分配到的分区，彼此之间并无逻辑上的干扰。



紧接着消费组内又加入了一个新的消费者C2，消费者C0、C1和C2按照上图中的方式各自负责消费所分配到的分区。

消费者与消费组这种模型可以让整体的消费能力具备横向伸缩性，我们可以增加（或减少）消费者的个数来提高（或降低）整体的消费能力。对于分区数固定的情况，一味地增加消费者并不会让消费能力一直得到提升，如果消费者过多，出现了消费者的个数大于分区个数的情况，就会有消费者分配不到任何分区。参考下图，一共有8个消费者，7个分区，那么最后的消费者C7由于分配不到任何分区而无法消费任何消息。



以上分配逻辑都是基于默认的分区分配策略进行分析的，可以通过消费者客户端参数 partition.assignment.strategy 来设置消费者与订阅主题之间的分区分配策略。

对于消息中间件而言，一般有两种消息投递模式：点对点（P2P，Point-to-Point）模式和发布/订阅（Pub/Sub）模式。点对点模式是基于队列的，消息生产者发送消息到队列，消息消费者从队列中接收消息。发布订阅模式定义了如何向一个内容节点发布和订阅消息，这个内容节点称为主题（Topic），主题可以认为是消息传递的中介，消息发布者将消息发布到某个主题，而消息订阅者从主题中订阅消息。主题使得消息的订阅者和发布者互相保持独立，不需要进行接触即可保证消息的传递，发布/订阅模式在消息的一对多广播时采用。Kafka 同时支持两种消息投递模式，而这正是得益于消费者与消费组模型的契合：

- 如果所有的消费者都隶属于同一个消费组，那么所有的消息都会被均衡地投递给每一个消费者，即每条消息只会被一个消费者处理，这就相当于点对点模式的应用。
- 如果所有的消费者都隶属于不同的消费组，那么所有的消息都会被广播给所有的消费者，即每条消息会被所有的消费者处理，这就相当于发布/订阅模式的应用。

消费组是一个逻辑上的概念，它将旗下的消费者归为一类，每一个消费者只隶属于一个消费组。每一个消费组都会有一个固定的名称，消费者在进行消费前需要指定其所属消费组的名称，这个可以通过消费者客户端参数 group.id 来配置，默认值为空字符串。

消费者并非逻辑上的概念，它是实际的应用实例，它可以是一个线程，也可以是一个进程。同一个消费组内的消费者既可以部署在同一台机器上，也可以部署在不同的机器上。