

RangeAssignor分配策略(默认使用该策略)

RoundRobin strategy

Sticky strategy

在kafka原理中介绍, kafka在消费组分配分区时, 有两种算法: range 和 round-robin和Sticky (0. 11. x版本), 前两种都存在弊端

消费者客户端参数`partition.assignment.strategy`可以配置多个分配策略, 彼此之间以逗号分隔。

## RangeAssignor分配策略(默认使用该策略)

RangeAssignor策略的原理是按照消费者总数和分区总数进行整除运算来获得一个跨度, 然后将分区按照跨度进行平均分配, 以保证分区尽可能均匀地分配给所有的消费者。对于每一个topic, RangeAssignor策略会将消费组内所有订阅这个topic的消费者按照名称的字典序排序, 然后为每个消费者划分固定的分区范围, 如果不够平均分配, 那么字典序靠前的消费者会被多分配一个分区。(就是挨个给, 轮着来)

原文链接: <https://blog.csdn.net/u013256816/article/details/81123600>

## RoundRobin strategy

使用RoundRobin策略有两个前提条件必须满足:

- 同一个Consumer Group里面的所有消费者的num.streams必须相等;
- 每个消费者订阅的主题必须相同。

所以这里假设前面提到的2个消费者的num.streams = 2。RoundRobin策略的工作原理: 将所有主题的分区分成 TopicAndPartition 列表, 然后对 TopicAndPartition 列表按照 hashCode 进行排序, 这里文字可能说不清, 看下面的代码应该会明白:

```
val allTopicPartitions = ctx.partitionsForTopic.flatMap { case(topic, partitions) =>
  info("Consumer %s rebalancing the following partitions for topic %s: %s"
    .format(ctx.consumerId, topic, partitions))
  partitions.map(partition => {
    TopicAndPartition(topic, partition)
  })
}.toSeq.sortWith((topicPartition1, topicPartition2) => {
```

```
/*  
 * Randomize the order by taking the hashCode to reduce the likelihood of all  
 partitions of a given topic ending  
 * up on one consumer (if it has a high enough stream count).  
 */  
topicPartition1.toString.hashCode < topicPartition2.toString.hashCode  
})
```

最后按照round-robin风格将分区分别分配给不同的消费者线程。

在我们的例子里面，加入按照 hashCode 排序完的topic-partitions组依次为

T1-5, T1-3, T1-0, T1-8, T1-2, T1-1, T1-4, T1-7, T1-6, T1-9,

我们的消费者线程排序为

C1-0, C1-1, C2-0, C2-1,

最后分区分配的结果为：

C1-0 将消费 T1-5, T1-2, T1-6 分区；

C1-1 将消费 T1-3, T1-1, T1-9 分区；

C2-0 将消费 T1-0, T1-4 分区；

C2-1 将消费 T1-8, T1-7 分区；

链接：<https://www.jianshu.com/p/b7647a748329>

## Sticky strategy

Kafka从0.11.x版本开始引入这种分配策略，它主要有两个目的：

- 分区的分配要尽可能的均匀；
- 分区的分配尽可能的与上次分配的保持相同。

当两者发生冲突时，第一个目标优先于第二个目标。鉴于这两个目标，StickyAssignor策略的具体实现要比RangeAssignor和RoundRobinAssignor这两种分配策略要复杂很多。

有点像轮询的分配，但是加了一条，“尽可能的与上次分配保持相同”

来自：<https://blog.csdn.net/u013256816/article/details/81123625>

