

Kafka新建的分区会在哪个目录下创建

我们在《[Kafka创建Topic时如何将分区放置到不同的Broker中](#)》文章中已经学习到创建 Topic 的时候分区是如何分配到各个 Broker 中的。今天我们来介绍分区分配到 Broker 中之后，会在哪个目录下创建文件夹。

我们知道，在启动 Kafka 集群之前，我们需要配置好 `log.dirs` 参数，其值是 Kafka 数据的存放目录，这个参数可以配置多个目录，目录之间使用逗号分隔，通常这些目录是分布在不同的磁盘上用于提高读写性能。当然我们也可以配置 `log.dir` 参数，含义一样。只需要设置其中一个即可。

如果 `log.dirs` 参数只配置了一个目录，那么分配到各个 Broker 上的分区肯定只能在这个目录下创建文件夹用于存放数据。

但是如果 `log.dirs` 参数配置了多个目录，那么 Kafka 会在哪个文件夹中创建分区目录呢？答案是：Kafka 会在含有分区目录最少的文件夹中创建新的分区目录，分区目录名为 Topic名+分区ID。注意，是分区文件夹总数最少的目录，而不是磁盘使用量最少的目录！也就是说，如果你给 `log.dirs` 参数新增了一个新的磁盘，新的分区目录肯定是先在这个新的磁盘上创建直到这个新的磁盘目录拥有的分区目录不是最少为止。

代码实现逻辑如下；

```
private val logs = new Pool[TopicAndPartition, Log]()

/**
 * Create a log for the given topic and the given partition
 * If the log already exists, just return a copy of the existing log
 */
def createLog(topicAndPartition: TopicAndPartition, config: LogConfig): Log = {
  logCreationOrDeletionLock synchronized {
    var log = logs.get(topicAndPartition)

    // check if the log has already been created in another thread
    if(log != null)
      return log

    // if not, create it
    val dataDir = nextLogDir()
    val dir = new File(dataDir, topicAndPartition.topic + "-" + topicAndPartition.partition)
    dir.mkdirs()
    log = new Log(dir,
      config,
      recoveryPoint = 0L,
```

```
        scheduler,
        time)
logs.put(topicAndPartition, log)
info("Created log for partition [%s,%d] in %s with properties {%s}."
    .format(topicAndPartition.topic,
            topicAndPartition.partition,
            dataDir.getAbsolutePath,
            {import JavaConversions._; config.toProps.mkString(", ")}))
log
}
}

/**
 * Choose the next directory in which to create a log. Currently this is done
 * by calculating the number of partitions in each directory and then choosing the
 * data directory with the fewest partitions.
 */
private def nextLogDir(): File = {
  if(logDirs.size == 1) {
    logDirs(0)
  } else {
    // count the number of logs in each parent directory (including 0 for empty directories
    val logCounts = allLogs.groupBy(_.dir.getParent).mapValues(_.size)
    val zeros = logDirs.map(dir => (dir.getPath, 0)).toMap
    var dirCounts = (zeros ++ logCounts).toBuffer

    // choose the directory with the least logs in it
    val leastLoaded = dirCounts.sortBy(_._2).head
    new File(leastLoaded._1)
  }
}
```

从上面代码可以清楚看出，需要创建新的分区时，Kafka先从 logs 存储池中获取当前分区对应的 Log 对象。如果获取到了，说明不是新的分区，这时候直接返回 Log 实例；如果这个分区是新建的，肯定是获取不到，这时候需要调用 nextLogDir 函数获取再哪个目录上创建分区目录。其核心思想就是找到分区数最少的目录来创建新的分区。

当然，这种实现上会有几个问题：

- 分区数最少的目录未必是数据量最少的目录，如果分区数最少的目录恰恰是数据量最多的目录这样会导致磁盘使用不均衡；
- 这种实现也没有考虑到磁盘的读写负载。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: **【】**（**）**