修改副本因子

创建主题之后我们还可以修改分区的个数，同样可以修改副本因子（副本数）。修改副本因子的使用场景也很多，比如在创建主题时填写了错误的副本因子数而需要修改，再比如运行一段时间之后想要通过增加副本因子数来提高容错性和可靠性。

前面主要讲述了分区重分配的相关细节，本节中修改副本因子的功能也是通过重分配所使用的 kafka-reassign-partition.sh 脚本实现的。我们仔细观察一下上一节中的示例使用的 project.json 文件：

```
{
    "version": 1,
    "partitions": [
        {
            "topic": "topic-throttle",
            "partition": 1,
            "replicas": [
                2,
                0
            ],
            "log_dirs": [
                "any",
                "any"
            ]
        },
        {
            "topic": "topic-throttle",
            "partition": 0,
            "replicas": [
                0,
```

```
                2
            ],
            "log_dirs": [
                "any",
                "any"
            ]
        },
        {
            "topic": "topic-throttle",
            "partition": 2,
            "replicas": [
                0,
                2
            ],
            "log_dirs": [
                "any",
                "any"
            ]
        }
    ]
}
```

可以观察到 JSON 内容里的 replicas 都是2个副本，我们可以自行添加一个副本，比如对分区1而言，可以改成下面的内容：

```
{
    "topic": "topic-throttle",
    "partition": 1,
    "replicas": [
        2,
        1,
        0
    ],
    "log_dirs": [
        "any",
        "any",
        "any"
    ]
}
```

我们可以将其他分区的 replicas 内容也改成[0,1,2]，这样每个分区的副本因子就都从2增加到了3。注意增加副本因子时也要在log_dirs中添加一个“any”，这个log_dirs 代表 Kafka 中的日志目录，对应于 broker 端的 log.dir 或 log.dirs 参数的配置值，如果不需要关注此方面的细节，那么可以简单地设置为“any”。我们将修改后的 JSON 内容保存为新的 add.json 文件。在执行 kafka-reassign-partition.sh 脚本前，主题 topic-throttle 的详细信息（副本因子为2）如下：

```
[root@node1 kafka_2.11-2.0.0]# bin/kafka-
topics.sh --zookeeper localhost:2181/ kafka --
describe --topic topic-throttle
Topic:topic-throttle    PartitionCount:3
ReplicationFactor:2 Configs:
    Topic: topic-throttle   Partition: 0
Leader: 0   Replicas: 0,1   Isr: 0,1
    Topic: topic-throttle   Partition: 1
Leader: 1   Replicas: 1,2   Isr: 2,1
    Topic: topic-throttle   Partition: 2
Leader: 2   Replicas: 2,0   Isr: 2,0
```

执行 kafka-reassign-partition.sh 脚本（execute），详细信息如下：

```
[root@node1 kafka_2.11-2.0.0]# bin/kafka-
reassign-partitions.sh --zookeeper
localhost:2181/kafka --execute --reassignment-
json-file add.json
Current partition replica assignment

{"version":1,"partitions":[{"topic":"topic-
throttle","partition":2,"replicas":
[2,0],"log_dirs":["any","any"]},{"topic":"topic-
throttle","partition":1,"replicas":
[1,2],"log_dirs":["any","any"]},{"topic":"topic-
throttle","partition":0,"replicas":
[0,1],"log_dirs":["any","any"]}]}

Save this to use as the --reassignment-json-file
option during rollback
Successfully started reassignment of partitions.
```

执行之后再次查看主题 topic-throttle 的详细信息，详细信息如下：

```
[root@node1 kafka_2.11-2.0.0]# bin/kafka-
topics.sh --zookeeper localhost:2181/ kafka --
describe --topic topic-throttle
Topic:topic-throttle    PartitionCount:3
ReplicationFactor:3 Configs:
    Topic: topic-throttle   Partition: 0
Leader: 0   Replicas: 0,1,2 Isr: 0,1,2
    Topic: topic-throttle   Partition: 1
Leader: 1   Replicas: 0,1,2 Isr: 2,1,0
    Topic: topic-throttle   Partition: 2
Leader: 2   Replicas: 0,1,2 Isr: 2,0,1
```

可以看到相应的副本因子数已经增加到3了。

与修改分区数不同的是，副本数还可以减少，这个其实很好理解，最直接的方式是关闭一些 broker，不过这种手法不太正规。这里我们同样可以通过 kafka-reassign-partition.sh 脚本来减少分区的副本因子。再次修改 project.json 文件中的内容，内容参考如下：

```
{"version":1,"partitions":[{"topic":"topic-
throttle","partition":2,"replicas":
[0],"log_dirs":["any"]},{"topic":"topic-
throttle","partition":1,"replicas":
[1],"log_dirs":["any"]},{"topic":"topic-
throttle","partition":0,"replicas":
[2],"log_dirs":["any"]}]}
```

再次执行 kafka-reassign-partition.sh 脚本（execute）之后，主题 topic-throttle 的详细信息如下：

```
[root@node1 kafka_2.11-2.0.0]# bin/kafka-
topics.sh --zookeeper localhost:2181/ kafka --
describe --topic topic-throttle
Topic:topic-throttle    PartitionCount:3
ReplicationFactor:1 Configs:
     Topic: topic-throttle  Partition: 0
Leader: 2   Replicas: 2 Isr: 2
     Topic: topic-throttle  Partition: 1
Leader: 1   Replicas: 1 Isr: 1
     Topic: topic-throttle  Partition: 2
Leader: 0   Replicas: 0 Isr: 0
```

可以看到主题 topic-throttle 的副本因子又被修改为1了。

细心的读者可能注意到我们执行 kafka-reassign-partition.sh 脚本（execute）所使用的候选方案都是手动修改的，在增加副本因子的时候由于整个示例集群中只有3个 broker 节点，从2增加到3只需填满副本即可。再者，示例中减少副本因子的时候改成了1，这样可以简单地把各个 broker 节点轮询一遍，如此也就不太会有负载不均衡的影响。不过在真实应用中，可能面对的是一个包含了几十个 broker 节点的集群，将副本数从2修改为5，或者从4修改为3的时候，如何进行合理的分配是一个关键的问题。

我们可以参考17节中的分区副本的分配来进行相应的计算，不过如果不是通过程序来得出结果而是通过人工去计算的，也确实比较烦琐。下面演示了如何通过程序来计算出分配方案（实质上是17节中对应的方法），如代码清单24-1所示。

代码清单24-1 分配方案计算（Scala）

```
object ComputeReplicaDistribution {
  val partitions = 3
  val replicaFactor = 2

  def main(args: Array[String]): Unit = {
    val brokerMetadatas = List(new
BrokerMetadata(0, Option("rack1")),
      new BrokerMetadata(1, Option("rack1")),
      new BrokerMetadata(2, Option("rack1")))
    val replicaAssignment =
AdminUtils.assignReplicasToBrokers(brokerMetadata
s,
      partitions, replicaFactor)
    println(replicaAssignment)
  }
}
```

代码中计算的是集群节点为[0,1,2]、分区数为3、副本因子为2、无机架信息的分配方案，程序输出如下：

```
Map(2 -> ArrayBuffer(0, 2), 1 -> ArrayBuffer(2,
1), 0 -> ArrayBuffer(1, 0))
```

分区2对应于[0,2]，分区1对应于[2,1]，分区0对应于[1,0]，所以在一个3节点的集群中将副本因子修改为2的对应候选方案为：

```
{"version":1,"partitions":[{"topic":"topic-
throttle","partition":2,"replicas":
[0,2],"log_dirs":["any","any"]},{"topic":"topic-
throttle","partition":1,"replicas":
[2,1],"log_dirs":["any","any"]},{"topic":"topic-
throttle","partition":0,"replicas":
[1,0],"log_dirs":["any","any"]}]}
```