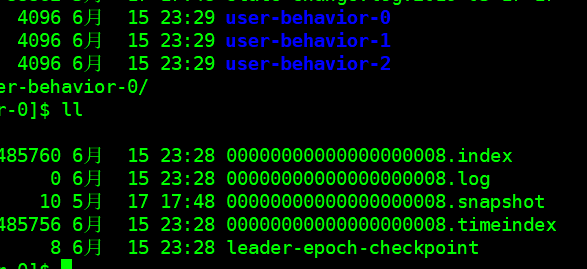
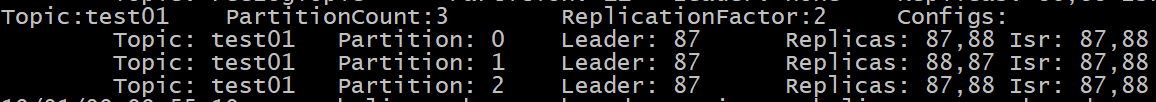
kafka集群扩容后，新的broker上面不会有数据进入这些节点，只有创建新的Topic后才会有数据，或者将已有的Partiton迁移到新的服务器上，将一些topic分区迁移到新的Broker上。另一种情况将集群减少Broker节点，对原有的Topic信息进行迁移，重新分配broker节点

案例将01,02节点上的topic test01（两个副本） 转移到03,04broker上（重新分配）

1. 进入kafka的bin目录
2. 新建文件 vi topics-to-move.json
3. 执行脚本，生成分配计划 –broker-list “03,04”是目标broker
4. 
5. 本次案例将原本存在于两个borker节点（86,87）上的topic：test01（两副本），转移到87,88两个broker上（重新分配）。
6. 生成分配计划
7. 进入kafka的bin目录：
8. cd /opt/cloudera/parcels/KAFKA/bin
9. 新建文件：
10. vi topics-to-move.json
11. {"topics":
12. [{"topic":"test01"}],
13. "version": 1
14. }
15. 执行脚本，生成分配计划：（--broker-list "87,88"是目标borker）
16. kafka-reassign-partitions --zookeeper bigdata102:2181,bigdata103:2181,bigdata104:2181 --topics-to-move-json-file topics-to-move.json --broker-list "87,88" --generate
17. 执行结果如下：
18. Current partition replica assignment #当前分区的副本分配
19. {"version":1,"partitions":[{"topic":"test01","partition":0,"replicas":[86,87]},{"topic":"test01","partition":1,"replicas":[86,87]},{"topic":"test01","partition":2,"replicas":[86,87]}]}
20. Proposed partition reassignment configuration #建议的分区配置
21. {"version":1,"partitions":[{"topic":"test01","partition":0,"replicas":[87,88]},{"topic":"test01","partition":1,"replicas":[88,87]},{"topic":"test01","partition":1,"replicas":[87,88]}]}
22. 新建文件topic-reassignment.json，将建议的分区设置copy进去：
23. vi topic-reassignment.json
24. {"version":1,"partitions":[{"topic":"test01","partition":0,"replicas":[87,88]},{"topic":"test01","partition":1,"replicas":[88,87]},{"topic":"test01","partition":1,"replicas":[87,88]}]}
25. 执行分配计划
26. 根据建议的分区设置，重新分配topic的分区到broker节点
27. kafka-reassign-partitions --zookeeper bigdata102:2181,bigdata103:2181,bigdata104:2181 --reassignment-json-file topic-reassignment.json --execute
28. 查看执行状态
29. kafka-reassign-partitions --zookeeper bigdata102:2181,bigdata103:2181,bigdata104:2181 --reassignment-json-file topic-reassignment.json --verify
30. 最终完成分配
31. 

**Kafka上对一个Topic增加Partition**

通过kafka-topics.sh工具的alter命令，将topic\_test的partitions从12增加到20；

./bin/kafka-topics.sh –zookeeper 192.168.2.225:2183/config/mobile/mq –alter –partitions 20 –topic topic\_test

**迁移数据到其他broker上**

Kafka提供了kafka-reassign-partitions.sh进行数据迁移

* --generate: 根据给予的Topic列表和Broker列表生成迁移计划。generate并不会真正进行消息迁移，而是将消息迁移计划计算出来，供execute命令使用。
* --execute: 根据给予的消息迁移计划进行迁移。
* --verify: 检查消息是否已经迁移完成。

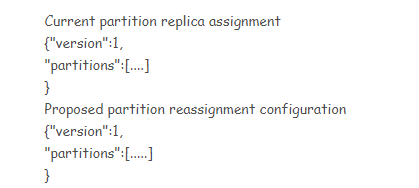
**1、**topic为test目前在broker id为1,2,3的机器上，现又添加了两台机器，broker id为4,5，现在想要将压力平均分散到这5台机器上。

**手动生成一个json文件topic.json**



#### 2、调用--generate生成迁移计划，将test扩充到所有机器上

./bin/kafka-reassign-partitions.sh --zookeeper vlnx111122:2181 --topics-to-move-json-file topic.json --broker-list"1,2,3,4,5" --generate



3、Current partition replica assignment表示当前的消息存储状况。Proposed partition reassignment configuration表示迁移后的消息存储状况。  
将迁移后的json存入一个文件reassignment.json，供--execute命令使用。

#### 执行--execute进行扩容。

./bin/kafka-reassign-partitions.sh --zookeeper vlnx111122:2181 --reassignment-json-file reassignment.json --execute

Current partition replica assignment

...

Save this to use as the --reassignment-json-file option during rollback

#### 使用--verify查看进度

./bin/kafka-reassign-partitions.sh --zookeeper vlnx111122:2181 --reassignment-json-file reassignment.json --verify

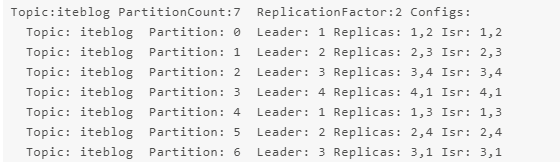
1. ./bin/kafka-console-producer.sh --broker-list vlnx111111:9092 --topic test
2. ./bin/kafka-console-consumer.sh --zookeeper vlnx111122:2181 --topic test --from-beginning
3. ./bin/kafka-topics.sh --zookeeper vlnx111122:2181 --list
4. ./bin/kafka-topics.sh --zookeeper vlnx111122:2181 --create --replication-factor 2 --partition 6 --topic test
5. ./bin/kafka-topics.sh --zookeeper vlnx111122:2181 --delete --topic test
6. ./bin/kafka-topics.sh --zookeeper vlnx111122:2181 --describe --topic test

## Kafka集群的扩展以及重新分布分区

1. 将已经部署好的kafka节点中复制相应 的配置文件 把 broker id 改成全局唯一，启动节点加入Kafka集群
2. 新加入的节点不会自动分配数据，无法负载，除非创建新的topic
3. 手动将部分分区移动到新的kafka节点上
   1. 查看topic各个分区分布位置

bin/kafka-topics.sh --zookeeper hadoop102:2181,hadoop103:2181,hadoop104:2181 –-list

b）查看topic分区情况bin/kafka-topics.sh --topic GMALL\_STARTUP --describe --zookeeper hadoop102:2181,hadoop103:2181,hadoop104:2181



C）将1-4节点上的分区分不到1-5节点

创建文件，规定哪些topic需要重新分区，文件内容如下

cat topics-to-move.json

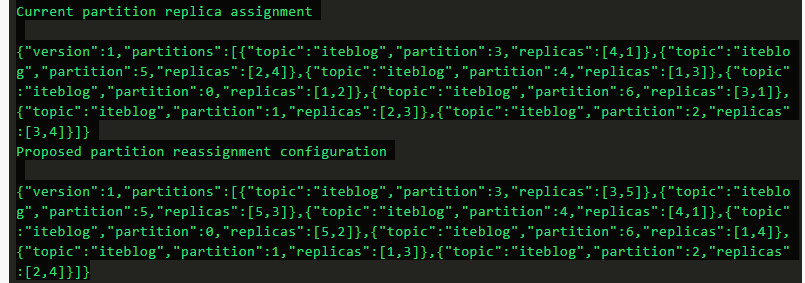
{"topics": [{"topic": "iteblog"}],

  "version":1

}

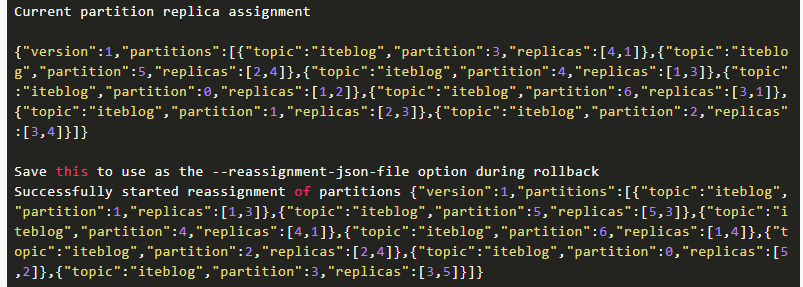
D) 然后使用kafka-reassign-partitions.sh工具生成reassign plan

bin/kafka-reassign-partitions.sh --zookeeper www.iteblog.com:2181 --topics-to-move-json-file topics-to-move.json --broker-list "1,2,3,4,5" --generate



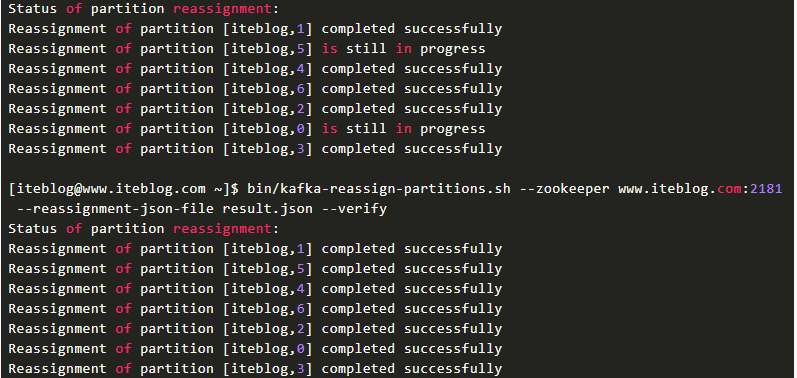
E) 将这些内容保存到名为result.json文件里面（文件名不重要，文件格式也不一定要以json为结尾，只要保证内容是json即可），然后执行这些reassign plan：

bin/kafka-reassign-partitions.sh --zookeeper www.iteblog.com:2181 --reassignment-json-file result.json –execute



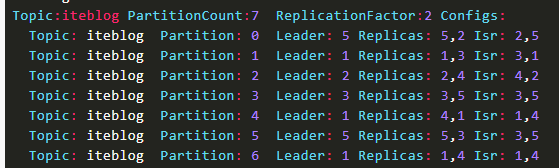
这样Kafka就在执行reassign plan，我们可以校验reassign plan是否执行完成：

bin/kafka-reassign-partitions.sh --zookeeper www.iteblog.com:2181 --reassignment-json-file result.json --verify



可以看出，分区正在Reassignment的状态是still in progress；如果分区Reassignment完成则completed successfully，然后我们就可以看到分区已经按照生成的reassign plan进行，我们可以看下topic各个分区现在的分布情况

./bin/kafka-topics.sh --topic iteblog --describe --zookeeper www.iteblog.com:2181



分区的分布的确和操作之前不一样了，broker 5上已经有分区分布上去了。但是仔细的同学应该可以发现，broker 4上居然没有分区的Leader，这肯定不是我们想要的！所以使用kafka-reassign-partitions.sh工具生成的reassign plan只是一个建议，方便大家而已。其实我们自己完全可以编辑一个reassign plan，然后执行它，如下：

{

    "version": 1,

    "partitions": [

        {

            "topic": "iteblog",

            "partition": 0,

            "replicas": [

                1,

                2

            ]

        },

        {

            "topic": "iteblog",

            "partition": 1,

            "replicas": [

                2,

                3

            ]

        },

        {

            "topic": "iteblog",

            "partition": 2,

            "replicas": [

                3,

                4

            ]

        },

        {

            "topic": "iteblog",

            "partition": 3,

            "replicas": [

                4,

                5

            ]

        },

        {

            "topic": "iteblog",

            "partition": 4,

            "replicas": [

                5,

                1

            ]

        },

        {

            "topic": "iteblog",

            "partition": 5,

            "replicas": [

                1,

                3

            ]

        },

        {

            "topic": "iteblog",

            "partition": 6,

            "replicas": [

                2,

                4

            ]

        }

    ]

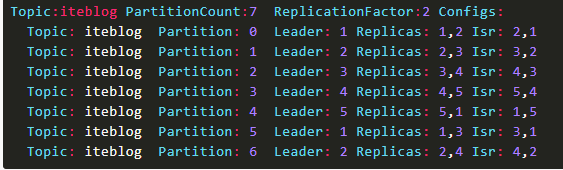
}

将上面的json数据文件保存到result.json文件中，然后也是执行它：

bin/kafka-reassign-partitions.sh --zookeeper www.iteblog.com:2181 --reassignment-json-file result.json --execute

等这个reassign plan执行完，我们再来看看分区的分布：

./bin/kafka-topics.sh --topic iteblog --describe --zookeeper [www.iteblog.com:2181](http://www.iteblog.com:2181)



## Kafka集群的扩容（Topic迁移）

1. kafka的集群扩容实际上就是把 topic 的 partition 移动到新加的集群上。
2. 生成 topic 移动 json文件有两种方式:

通过 **--topics-to-move-json-file** 和 **--broker-list** 批量生成新的topic分区信息，然后根据该信息执行转移操作。

手动写要移动的topic信息，更灵活，但是在大量 topic 和 partition 的情况下非常繁琐并且容易出错。

**1. 启动新节点**

将原节点上的 kafka 目录通过 scp 命令拷贝到新节点，只需要修改配置文件中的 broker\_id 和 ip 地址，然后依次启动 kafka 服务。

**2. 移动topic**

移动 topicA , topicB

sudo vim topics-to-move.json

{"topics": [{"topic": "topicA"},

{"topic": "topicB"}],

"version":1

}

**3.将上述 topic 移动到 broker 3,4,5上，用 generate 命令生成 partition 分配 json 串。**

bin/kafka-reassign-partitions.sh --zookeeper zk1:2181

--topics-to-move-json-file topics-to-move.json --broker-list "3,4,5" --generate



**4、将 Proposed partition reassignment configuration 下方的数据写入 expand-cluster-reassignment.json上文说的第二种方式其实就是手动写该文件，而不用命令生成的。**

**vim expand-cluster-reassignment.json**

**{"version":1,"partitions":[{"topic":"topicA","partition":2,"replicas":[3,4]},**

**{"topic":"topicA","partition":1,"replicas":[5,3]},**

**{"topic":"topicB","partition":1,"replicas":[3,4]},**

**{"topic":"topicA","partition":0,"replicas":[4,5]},**

**{"topic":"topicB","partition":0,"replicas":[5,3]},**

**{"topic":"topicB","partition":2,"replicas":[4,5]}]}**

**5、用 execute 命令开始正式执行，将会把上述2个topic按 expand-cluster-reassignment 中的设置进行分配，该过程可能会影响其它topic的读取（延时）**

**6、查询执行状态**

bin/kafka-reassign-partitions.sh --zookeeper zk1:2181

--reassignment-json-file expand-cluster-reassignment.json –verify

**Reassignment of partition [topicA,4] is still in progress # 转移中**

**Reassignment of partition [topicB,2] completed successfully # 转移结束**

如果你看到一大堆的 ERROR 信息：**ERROR: Assigned replicas (0,5,1,2,3,4) don't match the list of replicas for reassignment (5,3,4) for partition [topicA,3]**

**Reassignment of partition [topicA,3] failed**

**过会再看就好了**