

kafka中partition和消费者对应关系

1个partition只能被同组的一个consumer消费，同组的consumer则起到均衡效果

消费者多于partition

topic: test 只有一个partition

创建一个topic——test,

```
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor
```

在g2组中启动两个consumer,

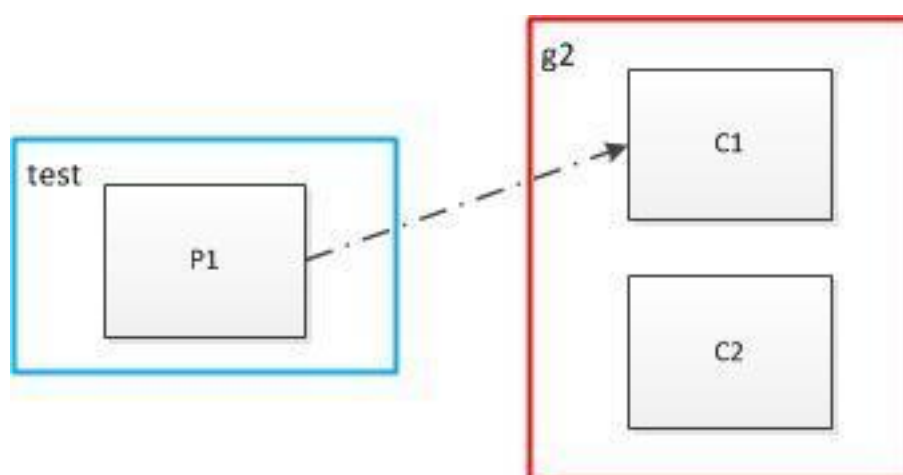
1. bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic
2. bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic

消费者数量为2大于partition数量1，此时partition和消费者进程对应关系如下：

```
bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe -
```

```
TOPIC PARTITION CURRENT-OFFSET LOG-END-OFFSET LAG CONSUMER-ID HOST CLIENT-I
test 0 9 9 0 consumer-1-4a2a4aa8-32f4-4904-9c16-1c0bdf7128a2 /127.0.0.1 con
- - - - - consumer-1-fd7b120f-fd21-4e07-8c23-87b71c1ee8a5 /127.0.0.1 consum
```

消费者consumer-1-fd7b120f-fd21-4e07-8c23-87b71c1ee8a5无对应的partition。
用图表示为



如上图，向test发送消息：1，2，3，4，5，6，7，8，9
只有C1能接收到消息，C2则不能接收到消息，即同一个partition内的消息只能被同一个组中的一个consumer消费。当消费者数量多于partition的数量时，多余的消费者空闲。
也就是说如果只有一个partition你在同一组启动多少个consumer都没用，partition的数量决定了此topic在同一组中被可被均衡的程度，例如partition=4，则可在同一组中被最多4个consumer均衡消费。

消费者少于和等于partition

topic：test2包含3个partition

```
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 test2
```

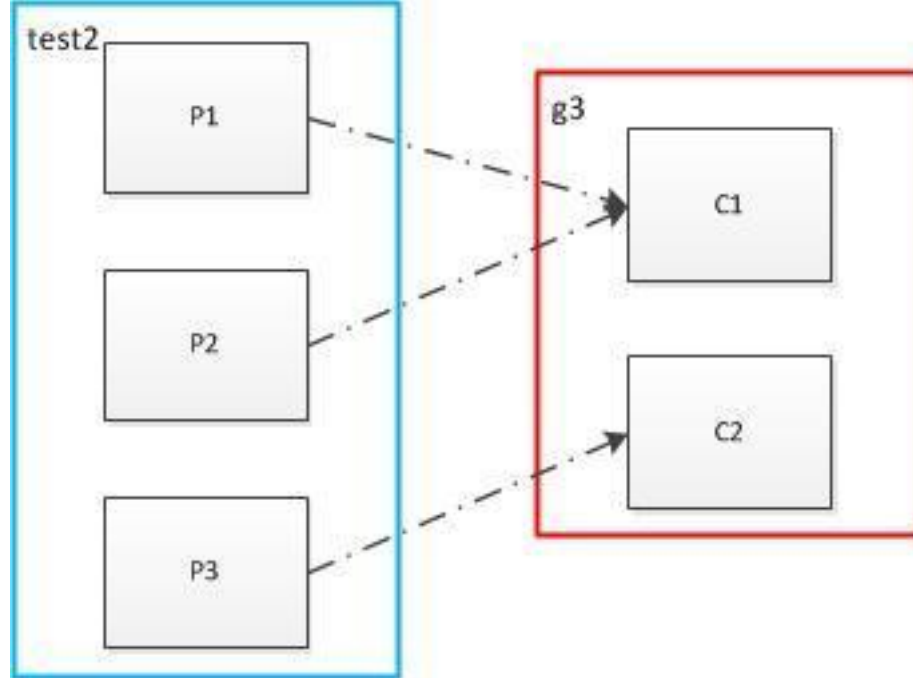
开始时，在g3组中启动2个consumer，

```
1.bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test2 --zookeeper localhost:2181
2.bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test2 --zookeeper localhost:2181
```

则对应关系如下：

TOPIC	PARTITION	CURRENT-OFFSET	LOG-END-OFFSET	LAG	CONSUMER-ID	HOST	CLIENT-ID
test2	0	8	8	0	consumer-1-8b872ef7-a2f0-4bd3-b2a8-7b26e4d8ab2c	/127.0.0.1	consumer-1-8b872ef7-a2f0-4bd3-b2a8-7b26e4d8ab2c
test2	1	7	7	0	consumer-1-8b872ef7-a2f0-4bd3-b2a8-7b26e4d8ab2c	/127.0.0.1	consumer-1-8b872ef7-a2f0-4bd3-b2a8-7b26e4d8ab2c
test2	2	8	8	0	consumer-1-f362847d-1094-4895-ad8b-1e1f1c88936c	/127.0.0.1	consumer-1-f362847d-1094-4895-ad8b-1e1f1c88936c

其中，consumer-1-8b872ef7-a2f0-4bd3-b2a8-7b26e4d8ab2c对应了2个partition
用图表示为：



生产者消费者对应关系2.jpg

消费者数量2小于partition的数量3，此时，向test2发送消息1，2，3，4，5，6，7，8，9

C1接收到1，3，4，6，7，9

C2接收到2，5，8

此时P1、P2对对应C1，即多个partition对应一个消费者，C1接收到消息量是C2的两倍

然后，在g3组中再启动一个消费者，使得消费者数量为3等于topic2中partition的数量

```
3.bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic t
```

对应关系如下：

TOPIC	PARTITION	CURRENT-OFFSET	LOG-END-OFFSET	LAG	CONSUMER-ID	HOST	CLIENT-I
test2	0	8	8	0	consumer-1-8b872ef7-a2f0-4bd3-b2a8-7b26e4d8ab2c	/127.0.0.1	co
test2	1	7	7	0	consumer-1-ab472ed5-de11-4e56-863a-67bf3a3cc36a	/127.0.0.1	co
test2	2	8	8	0	consumer-1-f362847d-1094-4895-ad8b-1e1f1c88936c	/127.0.0.1	co

此时，partition和消费者是一对一关系，向test2发送消息1，2，3，4，5，6，7，8，9

C1接收到了：2，5，8

C2接收到了：3，6，9

C3接收到了：1，4，7

C1，C2，C3均分了test2的所有消息，即消息在同一个组之间的消费者之间

均分了!

多个消费者组

启动g4组，仅包含一个消费者C1，消费topic2的消息，此时消费端有两个消费者组

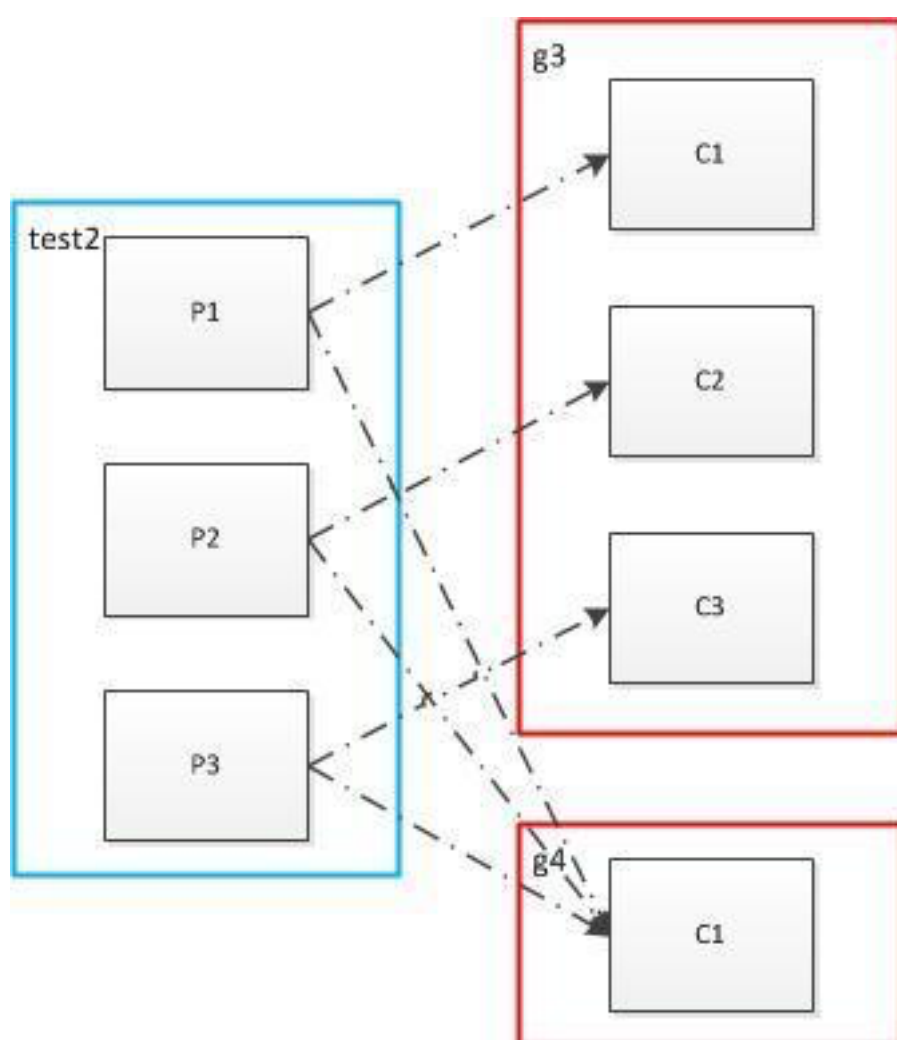
```
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic tes
```

g4组的C1的对应了test2的所有partition:

```
bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe -
```

TOPIC	PARTITION	CURRENT-OFFSET	LOG-END-OFFSET	LAG	CONSUMER-ID	HOST	CLIENT-I
test2	0	36	36	0	consumer-1-befc9234-260d-4ad3-b283-b67a2bf446ca	/127.0.0.1	
test2	1	35	35	0	consumer-1-befc9234-260d-4ad3-b283-b67a2bf446ca	/127.0.0.1	
test2	2	36	36	0	consumer-1-befc9234-260d-4ad3-b283-b67a2bf446ca	/127.0.0.1	

用图表示为



生产者消费者对应关系3.jpg

如上图，向test2发送消息1，2，3，4，5，6，7，8，9

那么g3组各个消费者及g4组的消费者接收到的消息是怎样地呢？欢迎思考！！

答案：

消息被g3组的消费者均分，g4组的消费者在接收到了所有的消息。

g3组：

C1接收到了：2，5，8

C2接收到了：3，6，9

C3接收到了：1，4，7

g4组：

C1接收到了：1，2，3，4，5，6，7，8，9

启动多个组，则会使同一个消息被消费多次