

https://github.com/futurewei-cloud/alcor-int/pull/37/files

# 以Host Group为Topic

## 一、key shared模式

1、每个topic（host group）拥有一个subscription。

2、使用key shared模式时，生产者发消息时必须指定key；我们的方案中生产者以host\_id为key，将key-val存入一个topic中。

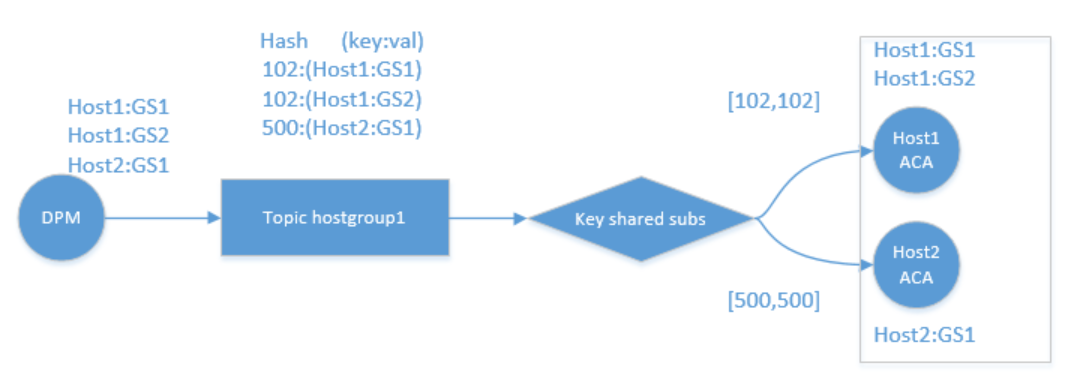
3、消费者采用Key shared模式订阅相应的topic，每个消费者组内所有消费者订阅同一个subscription，每个消费者会指定一个范围（[0,65535]的子集），key值通过Murmur3\_32Hash计算出的值若属于该范围，则可以拿到该key对应的消息。

4、每个ACA是一个消费者，它知道自己的host\_id，将[Murmur3\_32Hash(host\_id)%65535, Murmur3\_32Hash(host\_id)%65535]作为范围可以只拿到本host的消息，减少冗余。

5、由于key shared模式下topic内每个GS只能被一个消费者拿到。组播消息需要被复制多份，每份标上不同的key后发送给MQ。

~~6、某个ACA失效时，重启后可以从MQ中获取其range内包含的key-val。~~

7、当某一个consumer失效，属于它的GS会被其他consumer消费。我们可以不acknowledge这些GS。并开启broker的unacknowledged message automatic re-delivery mechanism，使这些unacknowledged GS在超时后重传



发送端：

client = PulsarClient.builder()

.serviceUrl("pulsar://localhost:6650")

.build();

producer = client.newProducer()

.topic("my-topic")

.create();

producer.newMessage().key("key-1").value("message-1-1\n".getBytes()).send();

接收端：

client = PulsarClient.builder()

.serviceUrl("pulsar://localhost:6650")

.build();

int hashcode = Murmur3\_32Hash.getInstance().makeHash("key") % 65536;

consumer = client.newConsumer()

.topic("my-topic")

.ackTimeout(30, TimeUnit.SECONDS)

.subscriptionName("my-subscription")

.subscriptionType(SubscriptionType.Key\_Shared)

.keySharedPolicy(KeySharedPolicy.stickyHashRange().ranges(Range.of(hashcode, hashcode)))

.subscribe();

## 二、Failover模式

1、每个topic拥有一个subscription。

2、host group内各host的ACA以failover模式订阅对应topic。

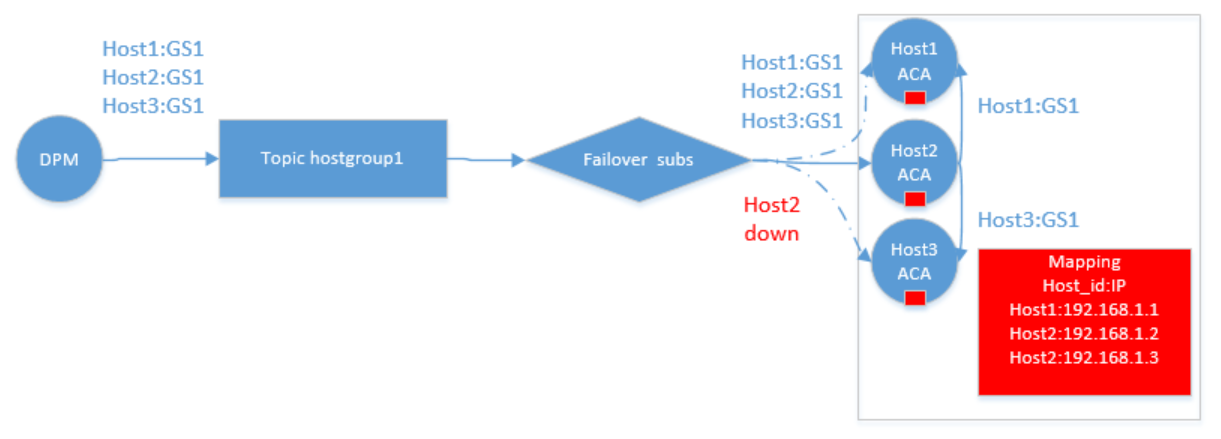
3、这些ACA以consumer name为依据排序，第一个ACA为master，接收该topic内所有msg，其他为slave ACA。当master断开连接后，下一个ACA成为master。

4、各ACA需要维护本group内所有其他host的IP。Master ACA与本group内其他ACA建立TCP连接，Master ACA接收到本组的全部GS后，根据host id区分该GS是否属于本host，属于master host的GS被相应模块处理。不属于master host的GS通过TCP协议发送给本group内其他对应的Slave ACA。当收到Slave ACA的ack时Master ACA对本GS进行acknowledge。否则不acknowledge该GS。

5、组播消息需要被复制多份，每份标上不同的key后发送给MQ。~~还可以在topic内只保留一份，在ACA侧进行分发，此方案需要ACA知道各vpc与group内各host的映射关系。~~

6、当host group内失效的host重新启动并订阅该subscription时，将作为后备消费者。

7、当slave host宕机时，master 无法收到相应GS的ack，因此不会acknowledge这些GS，这些unacknowledge GS超时会触发broker重发GS。



## 三、多级topic

Pulsar broker的逻辑结构是//tenant/namespace/topic

每个host group用一个ns，该ns下各topic对应一个具体topic

# 以VPC为topic

## Key shared模式

1、每个vpc topic拥有一个subscription。

2、使用key shared模式时，生产者发消息时必须指定key；我们的方案中生产者以host\_id为key，将key-val存入一个topic中。

3、消费者采用Key shared模式订阅多个vpc topic的subscription，包含同一vpc的多个host作为一个消费者组，每个消费者组内所有消费者订阅同一个subscription，每个消费者会指定一个范围（[0,65535]的子集），key值通过Murmur3\_32Hash计算出的值若属于该范围，则可以拿到该key对应的消息。

4、每个ACA是一个消费者，它知道自己的host\_id，将[Murmur3\_32Hash(host\_id)%65535, Murmur3\_32Hash(host\_id)%65535]作为范围可以只拿到本host的消息，减少冗余。

5、由于key shared模式下topic内每个GS只能被一个消费者拿到。组播消息需要被复制多份，每份标上不同的key后发送给MQ。

6、当某一个consumer失效，属于它的GS会被其他consumer消费。我们可以不acknowledge这些GS。并开启broker的unacknowledged message automatic re-delivery mechanism，使这些unacknowledged GS在超时后重传

## 二、Failover模式（一个VPC相关的主机不在一个rack，不合适）

1、每个VPC拥有一个subscription。

2、与某vpc相关的host的ACA以failover模式订阅多个相关的vpc topic。

3、这些ACA以consumer name为依据排序，第一个ACA为master，接收该topic内所有msg，当master断开连接后，下一个ACA成为master。

4、各ACA需要维护本vpc内所有其他host的IP。Master ACA接收到本组的全部GS后，根据host id区分该GS是否属于本host，属于本host的GS被相应模块处理；不属于本host的GS发送给本group内其他对应的host。

5、对于组播消息，除了可以按照方案一的做法。还可以在ACA侧进行分发，此方案需要aca知道各vpc与group内各host的映射关系。

6、当host group内失效的host重新启动并订阅该subscription时，将作为后备消费者。

7、当slave host宕机时，master 无法收到相应GS的ack，因此不会acknowledge这些GS，这些unacknowledge GS超时会触发broker重发GS。