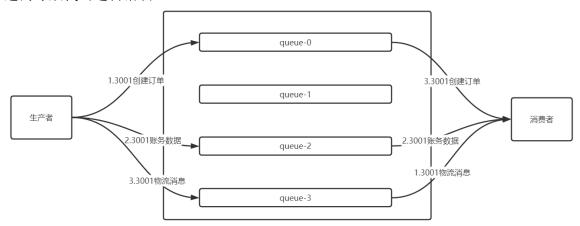
RocketMQ发送消息之有序消息

假设没有分区时有什么问题?

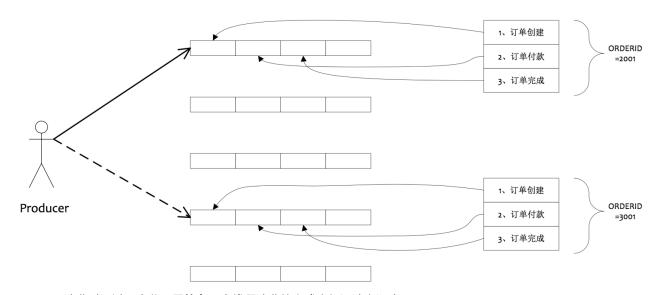
如果某一笔业务产分为多条普通消息同时发送,消费者无法保证按按生产者预期的顺序进行消费,进而导致代码逻辑错误。



分区有序消息

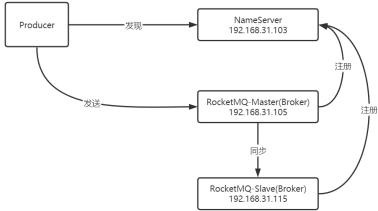
分区有序消息:与Kafka中的分区类似,把一个Topic消息分为多个分区"保存"和消费,在一个分区内的消息就是传统的队列,遵循FIFO(先进先出)原则。

全局有序消息:如果把一个 Topic 的分区数设置为 1,那么该 Topic 中的消息就是单分区,所有消息都遵循FIFO(先进先出)的原则。



Consumer消费时通过一个分区只能有一个线程消费的方式来保证消息顺序。

部署拓扑



```
源码讲解
pom. xm1
<dependency>
  <groupId>org.apache.rocketmq</groupId>
  <artifactId>rocketmq-client</artifactId>
  <version>4.9.2</version>
</dependency>
MessageType2
package com.itlaoqi.rocketmq.mtype;
import java.util.List;
@Slf4j
//发送分区顺序消息
public class MessageType2 {
  public static void main(String[] args) {
    DefaultMQProducer producer = new DefaultMQProducer("ProducerGroupName");
    producer.setNamesrvAddr("192.168.31.103:9876");
    producer.setRetryTimesWhenSendAsyncFailed(2);
    try {
      producer.start();
      Integer id = 4465;
      String data = "{\"id\":" + id+", + \"title\":\"X市2021年度第四季度税务汇总数据\"}";
      Message message = new Message("tax-data", "2021S4",id.toString(),
data.getBytes(RemotingHelper.DEFAULT CHARSET));
      //分区有序消息最大的区别便是调用send方法是,需要实现MessageQueueSelector接
口,确定使用哪个队列投递消息
      SendResult result = producer.send(message, new MessageQueueSelector() {
         @Override
         public MessageQueue select(List<MessageQueue> mqs, Message msq, Object
arg) {
           log.info("当前队列数量: " + mqs.size() + ",明细: " + mqs.toString());
           log.info("Message对象: " + msg.toString());
           int dataId = Integer.parseInt(msg.getKeys());
           int index = dataId % mqs.size();
           MessageQueue messageQueue = mgs.get(index);
           log.info("分区队列:" + messageQueue);
           return messageQueue;
```

```
}
},null);

log.info("消息已发送: Msgld:" + result.getMsgld() + ", 发送状态:" + result.getSendStatus());
}catch (Exception e){
    e.printStackTrace();
}finally {
    try {
        producer.shutdown();
        System.out.println("连接已关闭");
}catch (Exception e){
        e.printStackTrace();
    }
}
```

输出结果

15:04:49.407 [main] INFO com.itlaoqi.rocketmq.mtype.MessageType2 - 当前队列数量: 4,明细: [MessageQueue [topic=tax-data, brokerName=broker-a, queueld=0], MessageQueue [topic=tax-data, brokerName=broker-a, queueld=1], MessageQueue [topic=tax-data, brokerName=broker-a, queueld=2], MessageQueue [topic=tax-data, brokerName=broker-a, queueld=3]]

```
15:04:49.409 [main] INFO com.itlaoqi.rocketmq.mtype.MessageType2 - Message对象:
Message{topic='tax-data', flag=0, properties={KEYS=4465, WAIT=true, TAGS=2021S4}, body=[123, 34, 105, 100, 34, 58, 52, 52, 54, 53, 32, 44, 32, 43, 32, 34, 116, 105, 116, 108, 101, 34, 58, 34, 88, -27, -72, -126, 50, 48, 50, 49, -27, -71, -76, -27, -70, -90, -25, -84, -84, -27, -101, -101, -27, -83, -93, -27, -70, -90, -25, -88, -114, -27, -118, -95, -26, -79, -121, -26, -128, -69, -26, -107, -80, -26, -115, -82, 34, 125], transactionId='null'}
15:04:49.409 [main] INFO com.itlaoqi.rocketmq.mtype.MessageType2 - 分区队列:MessageQueue [topic=tax-data, brokerName=broker-a, queueld=1]
15:04:49.712 [main] INFO com.itlaoqi.rocketmq.mtype.MessageType2 - 消息已发送:Msgld:7F0000013C4018B4AAC20D891D290000,发送状态:SEND OK
```

全局有序消息

代码解析

在实现MessageQueueSelector接口时,固定选择某个队列就代表全局有序。注意:这里的全局有序代表broker中全局有序。如果消息被分发到不同的broker中,不保证有序,当然这种使用方法是错误的。

```
SendResult result = producer.send(message, new MessageQueueSelector() {
    @Override
    public MessageQueue select(List<MessageQueue> mqs, Message msg, Object arg) {
        MessageQueue messageQueue = mqs.get(0);
        return messageQueue;
    }
},id);
```

输出结果

15:08:18.295 [main] INFO com.itlaoqi.rocketmq.mtype.MessageType2 - 当前队列数量: 4,明细: [MessageQueue [topic=tax-data, brokerName=broker-a, queueld=0], MessageQueue [topic=tax-data, brokerName=broker-a, queueld=1], MessageQueue [topic=tax-data, brokerName=broker-a, queueld=2], MessageQueue [topic=tax-data, brokerName=broker-a, queueld=3]]

15:08:18.297 [main] INFO com.itlaoqi.rocketmq.mtype.MessageType2 - Message对象: Message{topic='tax-data', flag=0, properties={KEYS=4465, WAIT=true, TAGS=2021S4}, body=[123, 34, 105, 100, 34, 58, 52, 52, 54, 53, 32, 44, 32, 43, 32, 34, 116, 105, 116, 108, 101, 34, 58, 34, 88, -27, -72, -126, 50, 48, 50, 49, -27, -71, -76, -27, -70, -90, -25, -84, -84, -27, -101, -101, -27, -83, -93, -27, -70, -90, -25, -88, -114, -27, -118, -95, -26, -79, -121, -26, -128, -69, -26, -107, -80, -26, -115, -82, 34, 125], transactionId='null'} 15:08:18.297 [main] INFO com.itlaoqi.rocketmq.mtype.MessageType2 - 分区队列:MessageQueue [topic=tax-data, brokerName=broker-a, queueld=0] 15:08:18.602 [main] INFO com.itlaoqi.rocketmq.mtype.MessageType2 - 消息已发送: Msgld:7F00000134C418B4AAC20D8C4D220000,发送状态:SEND_OK