

## 76\_眼前一亮：再看 MQ 如何基于延时消息机制，优化大量订单的定时取消问题

---

**儒猿架构官网上线**，内有石杉老师架构课最新大纲，儒猿云平台详细介绍，敬请浏览

官网：[www.ruyuan2020.com](http://www.ruyuan2020.com)（建议 PC 端访问）

---

### 1.开篇

上节课介绍了在未引入 MQ 之前如何完成超时订单取消的功能，通过创建 Job 不断监听订单的状态和超时时间，并且在满足设定条件的情况下对订单进行取消。虽然这个方案可行，但是也会遇到维护和性能的问题。本节课来看 MQ 如何基于延时消息机制，优化订单取消的问题。今天课程的内容包括以下几个部分：

- RocketMQ 的延迟消息机制
- 调整取消订单业务流程图

### 2.RocketMQ 的延迟消息机制

如果说引入 RocketMQ 接受订单生成的消息，同时发送一个延迟 30 分钟的消息给消费者。在 30 分钟以后消费者接受到消息在进行订单状态的判断，不仅减少了扫描的工作，而且简化的实现逻辑，同时还将两个功能进行了解耦，不是挺好吗。顺着这个思路思考，我们通过引入 RocketMQ 的方式在创建订单的时候就发送一个延迟消息，这个消息会发送到 RocketMQ 内部延时的 topic 中，在 30 分钟后 RocketMQ 内部会重新投递到原 topic 给订阅该订单延时 topic 消费者消费，在消费的时候去判断订单是否被支付，如果没有支付就取消订单。如图 1 所示，将上述想法具体化为如下步骤：

1. 在创建订单的时候，就调用消息生产者生成一条延迟消息。
2. 这个延迟消息生产者的任务就是发送一条消息，消息在 30 分钟以后会被消费。
3. 延迟消息消费者会在 30 分钟以后获取消息通知，此时判断“订单是否支付？”

- 如果此时订单没有支付，就满足了订单创建以后 30 分钟都没有支付的条件，于是调用取消订单的操作。完成系统自动取消订单的功能。

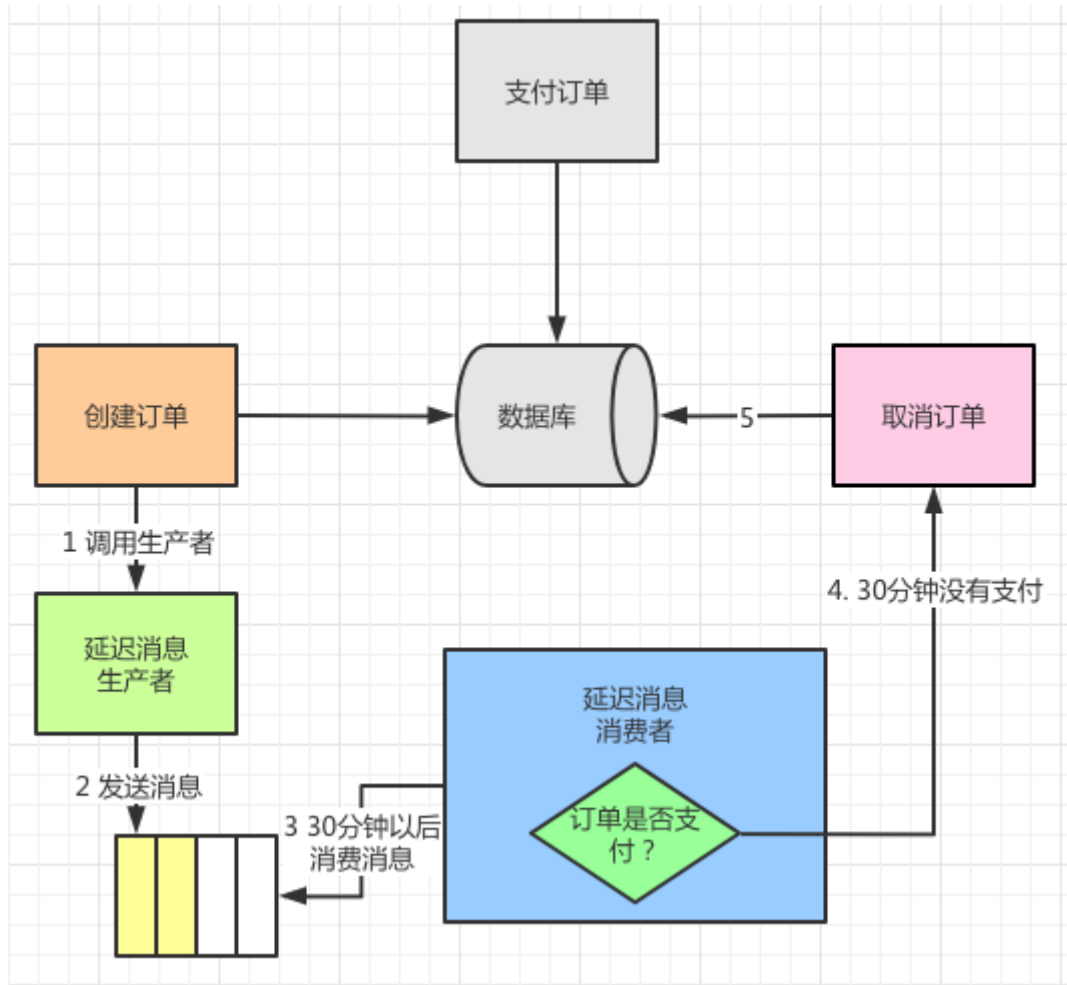


图 1 RocketMQ 的延迟消息机制

### 3.调整取消订单业务流程图

由于定时任务面对大量超时未付款订单的处理效率问题，我们将其替换成了 RocketMQ 的方案。如图 2 所示，方案按照以下几个步骤进行：

- 订单生成之后通过延迟消息生产者发送延迟消息，也就是上面提到的 30 分钟被消费的订单消息，这个消息会被发送到 RocketMQ 的队列中。
- 在 30 分钟的延迟时间以后，延迟消息消费者会获取到这个消息，并且对消息内容进行判断，如果还没有支付。
- 那么就满足了 30 分钟以后依旧没有支付的订单条件，于是执行取消订单的操作。

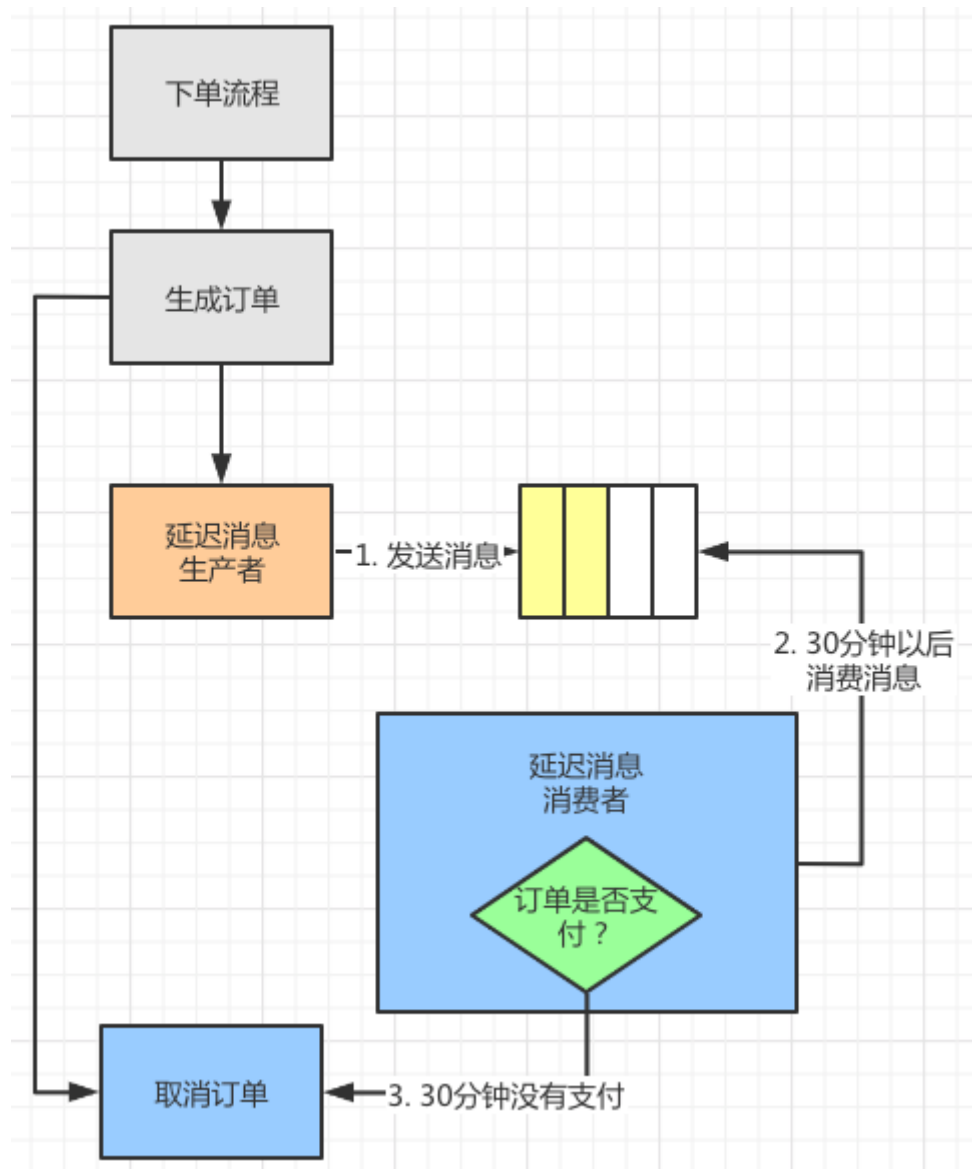


图 2 调整后的取消订单业务流程图

#### 4.总结

本节课为了解决大量未支付超时订单扫表性能问题，基于 RocketMQ 的延时消息进行优化。在创建订单的时候发送延迟消息（30 分钟），消费者在接受到延迟消息的时候再判断是否支付，未支付则取消订单。下节课会带大家进行取消订单业务代码的编写。下期见，拜拜。