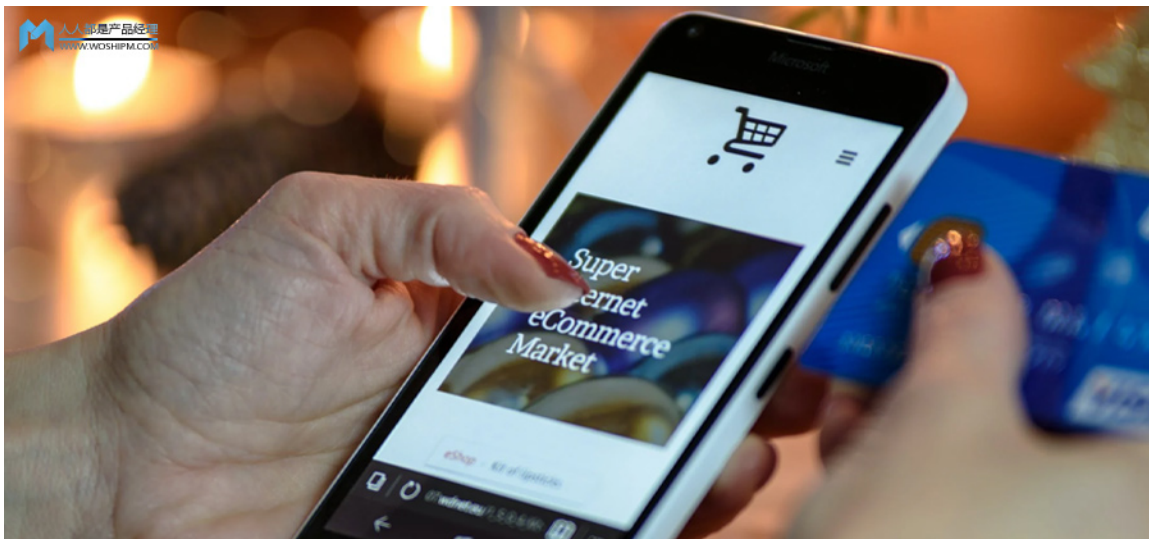


编辑导语：在订单支付的过程中，我们常常会遇到这样的问题：明明付了钱，也扣了款，但是订单却并没有成功。上文中，作者为我们分享了一次解决方案。在本篇文章中，作者又结合实际情况和案例，总结出了两种系统设计方案。



上次在文章《钱被扣走了，但是订单却未成功！支付掉单异常最全解决方案》提到，支付过程会出现「掉单、卡单」的情况，这种情况对于用户来讲，体验非常差，明明自己付了钱，扣了款，但是订单却未成功。

上篇文章我们简单说了下解决方案，这次小黑哥就结合生产实际碰到的情况，给出两种详细设计的方案：

定时轮询补偿方案 延迟消息补偿方案

大家可以根据自己系统的实际情况，选择性参考。

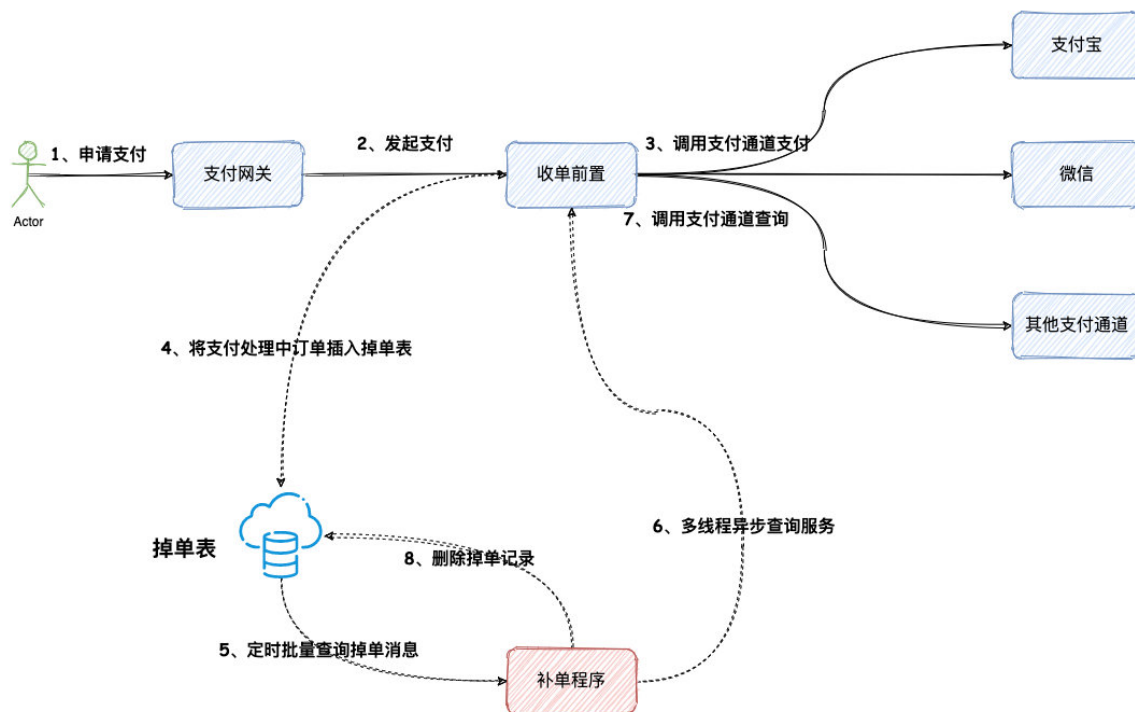
「当然了，以下设计方案可能并不完美，如果各位读者还有其他解决方案，欢迎留言指出，一起讨论，一起成长~」

一、定时轮询补偿方案

1. 整体流程

这个方案主要采用定时任务，批量查询掉单记录，从而驱动查询具体支付支付结果，然后更新内部订单。

整体方案流程图如下：



定时任务补偿

前三步流程没什么好说的，正常的支付流程，咱们针对后面几步具体详细说下。

第三步调用支付通道之后，如果支付通道端返回「支付受理成功或者支付处理中」，我们就需要调用第四步，将这类订单插入掉单表。

如果支付直接成功了，那就正常流程返回即可。

复习一下，网关类支付，比如支付宝、微信支付、网银支付，这种支付模式，支付通道仅仅返回支付受理成功，具体支付结果需要接收支付通道端的支付通知，这类支付我们将其称为异步支付。

相应的还有同步支付，比如银行卡支付，微信、支付宝代扣类支付，这类支付，同步就能返回支付结果。

第五步，补单应用将会定时查询数据库，批量查询掉单记录；第六步，补单应用使用线程池，多线程异步的方式发起掉单查询；第七步，调用支付通道支付查询接口。

重点来了，如果第七步支付结果查询为以下状态：

「支付结果为扣款成功」 「支付结果为明确失败」 「掉单记录查询达到最大次数」

「第八步就会删除掉单记录。」

最后，如果掉单查询依旧还是处理中，那么经过一定的延时之后，重复第五步，再次重新掉单补偿，直到成功或者查询到达最大次数。

2. 相关问题

「为什么需要新建一张掉单表？不能直接使用支付订单表，查询未成功的订单吗？」

这个问题，实际上确实可以直接使用的支付订单表，然后批量查询当天未成功的订单，补单程序发起支付查询。

那为什么需要新建一张掉单表？

主要是因为数据库查询效率问题，因为支付订单表每天都会大量记录新增，随着时间，这张表记录将会越来越多，越来越大。

「支付记录越多，批量范围查询效率就会变低，查询速度将会变慢。」

所以为了查询效率，新建一张掉单表。这张表里仅记录支付未成功的订单，所以数据量就会很小，那么查询效率就会很高。

另外，掉单表里的记录，不会被永久保存，只是临时性。当支付结果查询成功，或者支付结果明确失败，又或者查询次数到达规定最大次数，就会删除掉单记录。

「这就是第八步为什么需要删除掉单表的原因。」

如果需要保存每次掉单表查询详情，那么这里建议再新增一张掉单查询记录表，保存每一次的查询记录。针对这个方案，如果还有其他问题，欢迎留言。

3. 方案优缺点

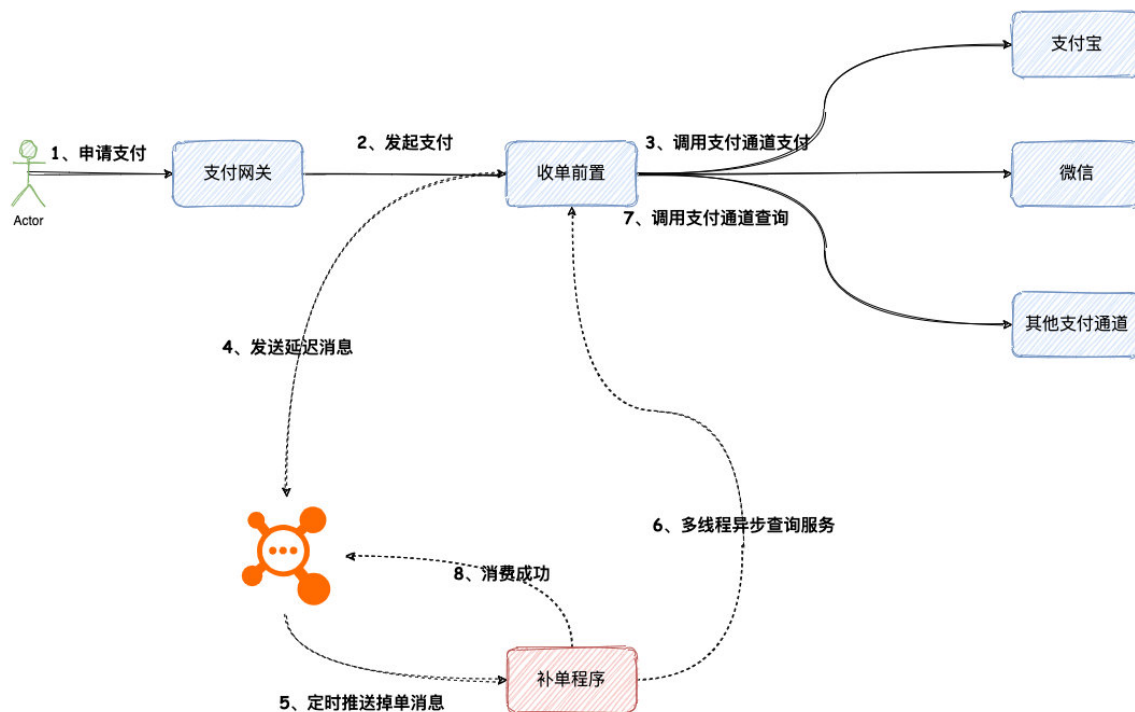
定时轮询补偿方案，最大的优点可能就是系统架构方案比较简单，比较容易实施，那么这个方案的缺点主要在于「定时任务」上。定时任务轮询方案天然会存在以下不足：

「轮询效率稍低」；每次查询数据库，已经被执行过记录，仍然会被扫描（补单程序将会根据一定策略决定是否发起支付通道查询），有「重复计算」的嫌疑；「时效性不够好」，如果每小时轮询一次，最差的情况下，时间误差会达到 1 小时；如果为了解决时效性问题，增加定时任务查询效率，那么 1 中查询效率跟 2 的重复计算问题将会更加明显。

二、延迟消息补偿方案

下面介绍另外一种掉单补偿方案，延迟消息补偿方案，这个方案整体流程与定时任务方案类似，最大区别可能在于，从一种「拉模式」变成一种「推模式」。

整体方案流程图如下：



这个方案主要流程跟定时方案类似，主要区别在于第四步、第五步、第八步。

第四步的流程从插入掉单表变更为往「延迟队列发送掉单消息」；第五步，补单程序接收掉单消息，然后触发支付掉单查询；第八步，如果第七步支付结果查询为以下状态：

支付结果为扣款成功 支付结果为明确失败 掉单记录查询达到最大次数

补单程序将会告知延迟队列消费成功，延迟队列将会删除这条掉单消息。其他状态将会告知消费失效，延迟队列将会在一定延时之后，再次发送掉单消息，然后继续重复第五步。

方案优缺点：

延迟消息的方案相对于定时轮询方案来讲：

无需再查询全部订单 效率高时效性较好

不过延迟消息这种方案，需要基于「延迟队列」，实现起来比较复杂，目前开源实现也比较少。

三、小结

支付掉单、卡单是支付过程中经常会碰到的事，我们可以采用异步补偿的方案，解决该问题，异步补偿方案可以采用如下两种：

定时轮询补偿方案 延迟消息补偿方案

定时轮询补偿方案实现起来比较简单，但是时效性稍差。而延迟消息补偿方案总体来说比较优秀，但是实现起来比较复杂。如果没有自定义的延迟时间的需求，可以直接采用 RocketMQ 延迟消息，简单快捷。

另外「延迟队列」使用场景还是比较多，不仅仅能用在掉单补偿上，还可以用于支付关单等场景。所以有能力开发的团队，可以开发一个通用的延迟队列。

历史支付文章推荐：

1. “轻轻一扫，立刻扣款”，解读付款码背后的原理
2. 产品设计：解读银行卡支付背后的原理
3. 手机没网了，却还能支付，这是什么原理？
4. 钱被扣走了，但是订单却未成功！支付掉单异常最全解决方案
5. 一笔订单，但是误付了两笔钱！这种重复付款异常到底该如何解决？

作者：楼下小黑哥；微信公号@程序通事，支付行业，后端技术

题图来自 Unsplash，基于 CC0 协议