

图文 82 生产案例：基于延迟消息机制优化大量订单的定时退款扫描问题！

255 人次阅读 2020-01-20 09:15:04

[详情](#) [评论](#)**生产案例：基于延迟消息机制优化大量订单的定时退款扫描问题！**

狸猫技术

The image is a promotional graphic for a MySQL optimization course. It features a large, stylized blue '零' character with the text '从零开始' (From Zero to Hero) overlaid. Below the character, the text '带你成为 MySQL 实战优化高手' (Lead you to become a MySQL practical optimization expert) is displayed. At the bottom, it says '继爆款专栏《从零开始带你成为 JVM 实战高手》后，又一重磅力作' (After the blockbuster column 'From Zero to Hero: Lead you to become a JVM practical expert', another重量级 work). The author is identified as '专栏作者：救火队队长 阿里资深技术专家' (Column author: Firefighting Team Leader, Senior Technical Expert at Alibaba). On the right side, there is a sidebar with the text '相关频道' (Related Channels) and a thumbnail for the course, which includes the title '从零开始', a progress bar showing '已更新9', and a '立即购买' (Buy Now) button.

继《从零开始带你成为JVM实战高手》后，又一重磅力作

(点击下方蓝字试听)

[《从零开始带你成为MySQL实战优化高手》](#)

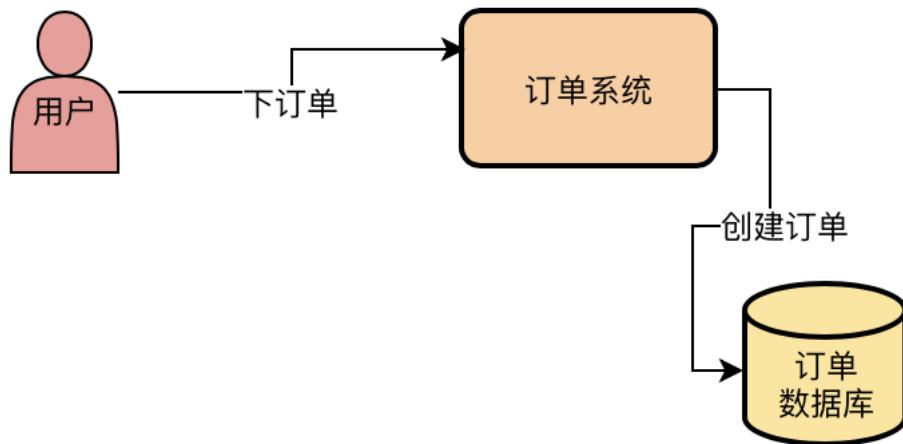
这篇文章以及接下来的几篇文章，将会是2020年春节前的最后几篇文章，考虑到很多人在看文章的时候都回老家准备过年了，可能也没太多精力在学习上，所以我们这几篇文章将会写的短小精干一些，避免耗费大家过多的时间。

在2020年春节过后，我还会讲解更多的MQ在实际生产环境下的一些技术问题以及对应的技术方案，比如MQ中百万消息积压解决方案、MQ消息轨迹、MQ监控报警、MQ中间件双写迁移方案、MQ权限方案，等等。

先来看看之前提到的一个订单退款扫描的问题。

我们先考虑一个正常的电商购物流程，一般来说我们作为用户在一个电商APP上都会选择一些商品加入购物车，然后对购物车里选择的一些商品统一下一个订单，此时后台的订单系统必然会在订单数据库中创建一个订单。

我们看下面的图



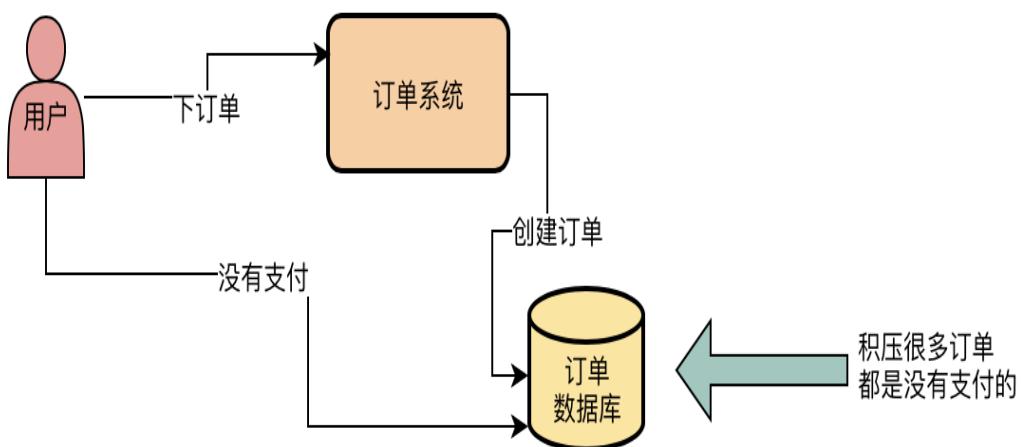
但是我们下了一个订单之后，虽然订单数据库里会有一个订单，订单的状态却是“待支付”状态，因为此时你其实还没有支付这个订单，我们的订单系统其实也在等待订单用户完成这个订单的支付。

这里就有两种可能了，一种可能是用户下单之后立马就支付掉了，那么接着订单系统可以走后续的流程，比如通过MQ发送消息通知优惠券系统给用户发优惠券，通知仓储系统进行调度发货，等等。

另外一种可能就是用户下单之后，一直在犹豫，迟迟没有下订单。

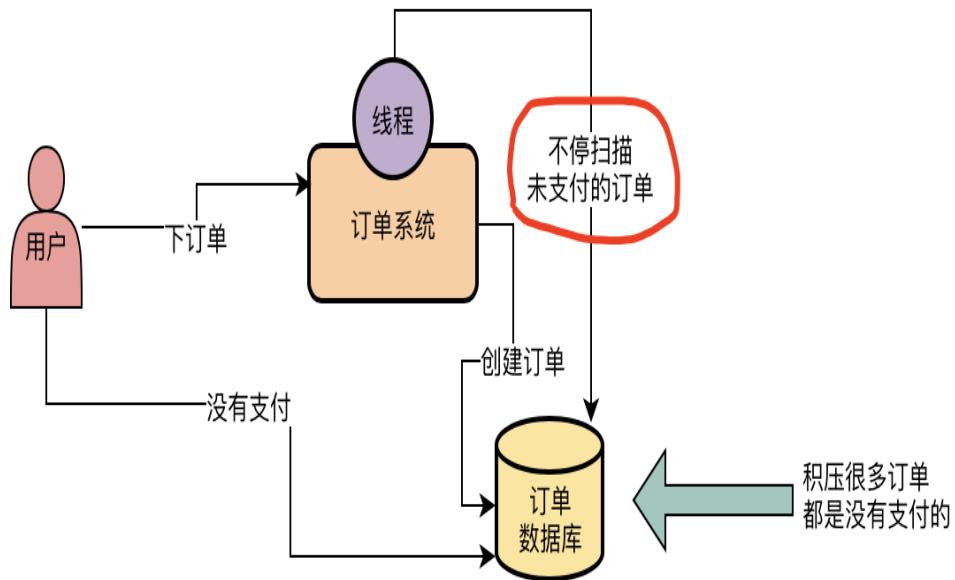
因此在实际情况中，其实APP的大量用户每天会下很多订单，但是不少订单可能是一直没有进行支付的。可能他下单之后犹豫了，可能是他忘了支付了！

我们看下图示，订单数据库中有很多订单是没有支付的。



所以一般订单系统都必须设置一个规则，当一个订单下单之后，超过比如30分钟没有支付，那么就必须订单系统自动关闭这个订单，后续你如果要购买这个订单里的商品，就得重新下订单了。

我们看下图，可能你的订单系统就需要有一个后台线程，不停的扫描订单数据库里所有的未支付状态的订单，看他如果超过30分钟了还没支付，那么就必须自动把订单状态更新为“已关闭”



但是这里就引入了一个问题，就是订单系统的后台线程必须要不停的扫描各种未支付的订单，这种实现方式实际上并不是很好。

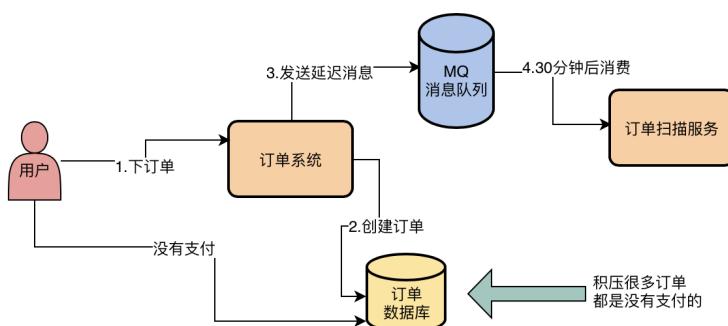
一个原因是未支付状态的订单可能是比较多的，然后你需要不停的扫描他们，可能每个未支付状态的订单要被扫描N多遍，才会发现他已经超过30分钟没支付了。

另外一个是很难去分布式并行扫描你的订单。因为假设你的订单数据量特别的多，然后你要是打算用多台机器部署订单扫描服务，但是每台机器扫描哪些订单？怎么扫描？什么时候扫描？这都是一系列的麻烦问题。
联系微信642600657

因此针对类似这种场景，MQ里的延迟消息往往就会出场了，他是特别适合在这种场景里使用的，而且在实际项目中，MQ的延迟消息使用的往往是很的。

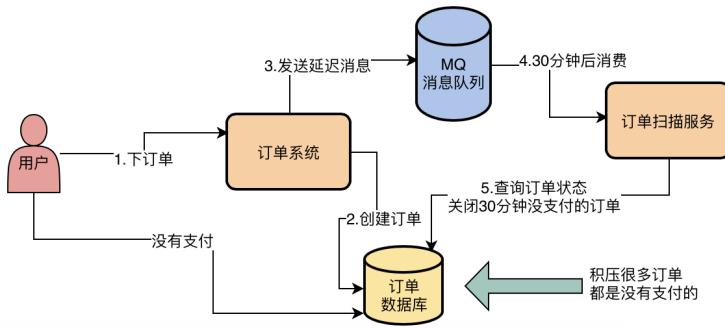
所谓延迟消息，意思就是说，我们订单系统在创建了一个订单之后，可以发送一条消息到MQ里去，我们指定这条消息是延迟消息，比如要等待30分钟之后，才能被订单扫描服务消费到

我们如下图所示



这样当订单扫描服务在30分钟后消费到了一条消息之后，就可以针对这条消息的信息，去订单数据库里查询这个订单，看看他在创建过后都过了30分钟了，此时他是否还是未支付状态？

如果此时订单还是未支付状态，那么就可以关闭他，否则订单如果已经支付了，就什么都不用做了，我们看下图



这种方式就比你用后台线程扫描订单的方式要好的多了，一个是对每个订单你只会在他创建30分钟后查询他一次而已，不会反复扫描订单多次。

另外就是如果你的订单数量很多，你完全可以让订单扫描服务多部署几台机器，然后对于MQ中的Topic可以多指定一个MessageQueue，这样每个订单扫描服务的机器作为一个Consumer都会处理一部分订单的查询任务。

所以MQ的延迟消息，是非常常用并且非常有用的一个功能。

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

狸猫技术窝精品专栏及课程推荐：

[《从零开始带你成为JVM实战高手》](#)
[《21天互联网Java进阶面试训练营》（分布式篇）](#)
[《互联网Java工程师面试突击》（第1季）](#)
[《互联网Java工程师面试突击》（第3季）](#)

慕课都有联系微信642600657

重要说明：

如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑

如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《付费用户如何加群》（[购买后可见](#)）