

## 图文 67 生产案例：从 RocketMQ 全链路分析一下为什么用户支付后包？

424 人次阅读 2019-12-27 07:00:00

[详情](#) [评论](#)

### 生产案例：从 RocketMQ 全链路分析一下为什么用户支付后没收到红包？

石杉老哥重磅力作：《互联网java工程师面试突击》（第3季）【强烈推荐】：



全程真题驱动，精研Java面试中**6大专题的高频考点**，从面试官的角度剖析面试

(点击下方蓝字试听)

[《互联网Java工程师面试突击》（第3季）](#)

#### 1、客服反馈的一个奇怪问题：支付之后没有收到红包

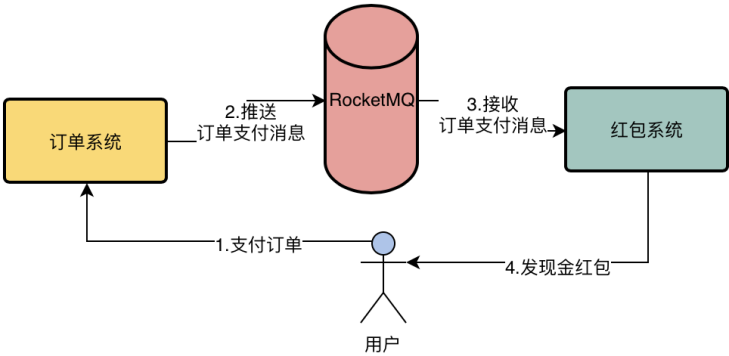
小猛在前段时间把MQ技术引入到自己的团队之后，花了一段时间去研究RocketMQ比较底层的原理，希望未来遇到一些MQ使用中的技术难题的时候，可以有足够的RocketMQ技术功底去解决。

果不其然，这一天客服给小猛的技术团队反馈了一个问题，有用户反馈说，按照规则应该是在支付之后可以拿到一个现金红包的，但是他在支付了一个订单之后，却并没有收到这个现金红包，于是就反馈给了客服。

这个问题可就奇怪了，大家都不知道问题是出在哪儿了。

经过一通排查，找了系统中打印的很多日志之后，发现了一个奇怪的现象

按理来说，订单系统在完成支付之后，会推送一条消息到RocketMQ里去，然后红包系统会从RocketMQ里接收那条消息去给用户发现金红包，我们看下图。



但是从订单系统和红包系统当天那个时间段的日志来看，居然只看到了订单系统有推送消息到RocketMQ的日志，但是并没有看到红包系统从RocketMQ中接收消息以及发现金红包的日志。

于是大家推测，问题可能就出在这儿了，是不是支付订单消息在传输的过程中丢失了？导致现金红包没有派发出去！

那么接着我们就来一步一步分析一下，对于MQ的使用过程中，到底有哪些地方会导致消息丢失？

2、订单系统推送消息到MQ的过程会丢失消息吗？

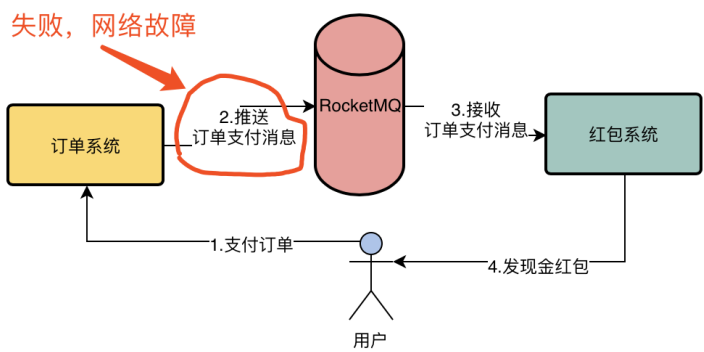
我们先看第一个问题，订单系统在接收到订单支付成功的通知之后，必然会去推送一条订单支付成功的消息到MQ的，那么在这个过程中，会出现丢失消息的问题吗？

其实答案是显而易见的，有可能会丢失

举一个比较常见的例子，比如订单系统在推送消息到RocketMQ的过程中，是通过网络去进行传输的，但是这个时候恰巧可能网络发生了抖动，也就是网络突然有点问题，导致这次网络通信失败了。

于是这个消息必然就没有成功投递给MQ

我们看下面的图里的示意，我用红圈标志出来了订单系统投递消息到MQ可能因为网络问题导致失败的情况。



除此之外，大家觉得还有没有其他的原因可能会导致订单系统推送消息到MQ失败的？

那是相当的多了，比如MQ确实是收到消息了，但是他的网络通信模块的代码出现了异常，可能是他内部的网络通信的bug，导致消息没成功处理。

或者是在你写消息到RocketMQ的过程中，刚好遇到了某个Leader Broker自身故障，其他的Follower Broker正在尝试切换为Leader Broker，这个过程中也可能会有异常。类似的问题可能还有其他的。

所以首先我们在使用任何一个MQ的时候，无论是RocketMQ，还是RabbitMQ，或者是Kafka，大家都要明确一点：不一定你发送消息出去就一定会成功，有可能就会失败，此时你的代码里可能会抛出异常，也可能不会抛出异常，这都不好说，具体要看到底什么原因导致的消息推送失败。

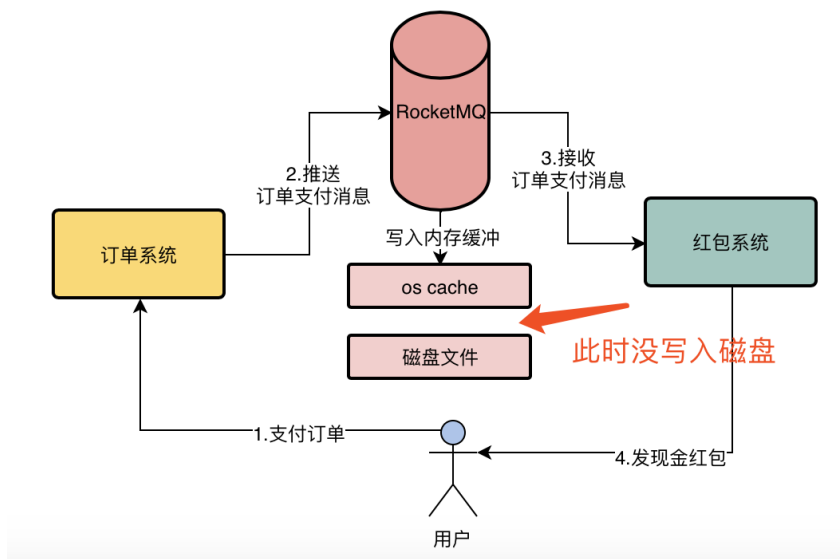
### 3、消息到达MQ了，MQ自己会导致消息丢失吗？

接着我们来看下一个问题，即使我们的订单系统成功的把消息写入了MQ，此时我们就可以想当然的认为你写成功了，消息就一定不会丢失了吗？

这个也是未必的，我们来分析一下为什么

因为通过之前的RocketMQ的底层原理的分析，我们现在都明确了一点，就是你的消息写入MQ之后，其实MQ可能仅仅是把这个消息给写入到page cache里，也就是操作系统自己管理的一个缓冲区，这本质也是内存

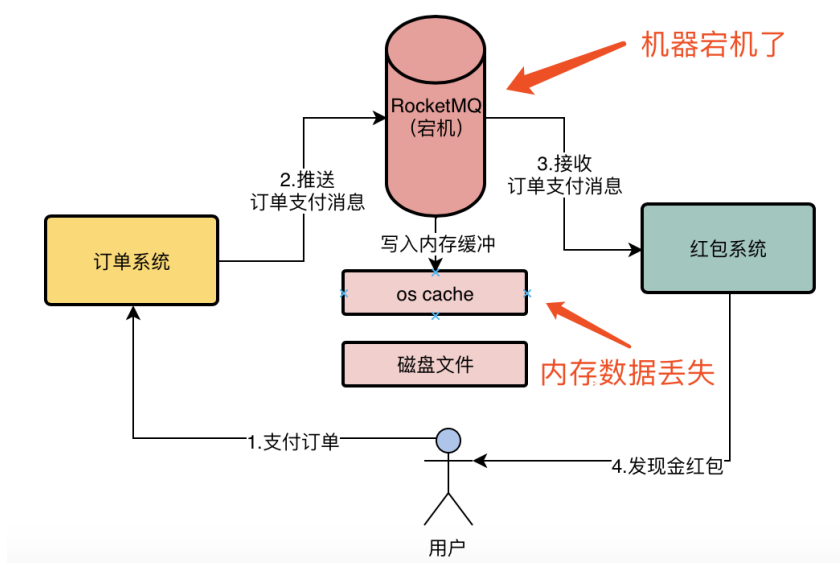
我们看下面的图示



大家注意图里的示意，可能你认为写成功了一个消息，但是此时仅仅进入了os cache，还没写入磁盘呢。

然后这个时候，假如要是出现了Broker机器的崩溃，大家思考一下，机器一旦宕机，是不是os cache内存中的数据就没了？

我们看下图

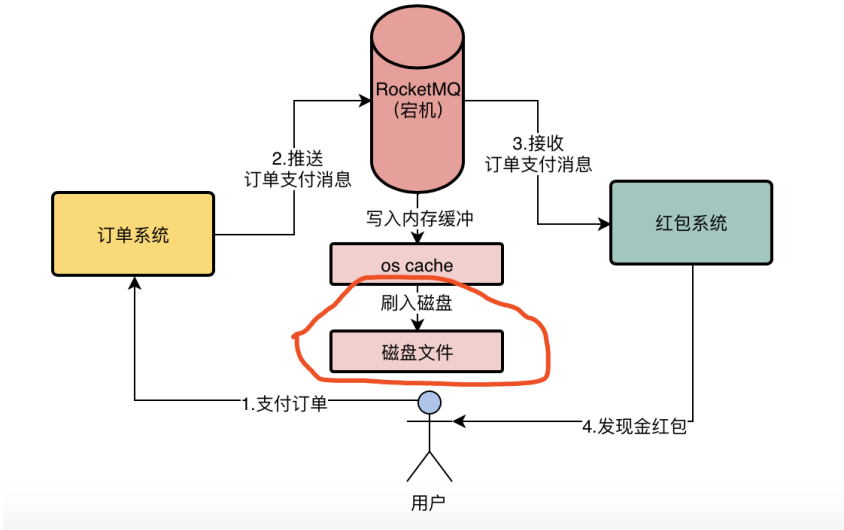


看到这里，想必每个人都会倒吸一口气，还真是，要是没有研究过MQ底层原理的人，可能还真不知道数据要是进入os cache的时候，碰上机器宕机，内存里的数据必然就会丢失了，你机器即使重启了，然后重启broker进程，此时这个数据也没了。

### 4、就算消息进入磁盘了，你以为真的万无一失吗？

我们接着来看下一个问题，我们通过之前的学习也都知道一点，那就是Broker把消息写入os cache之后，其实操作系统自己在一段不太确定的时间之后，他自己是会把数据从内存刷入磁盘文件里去的

我们看下图



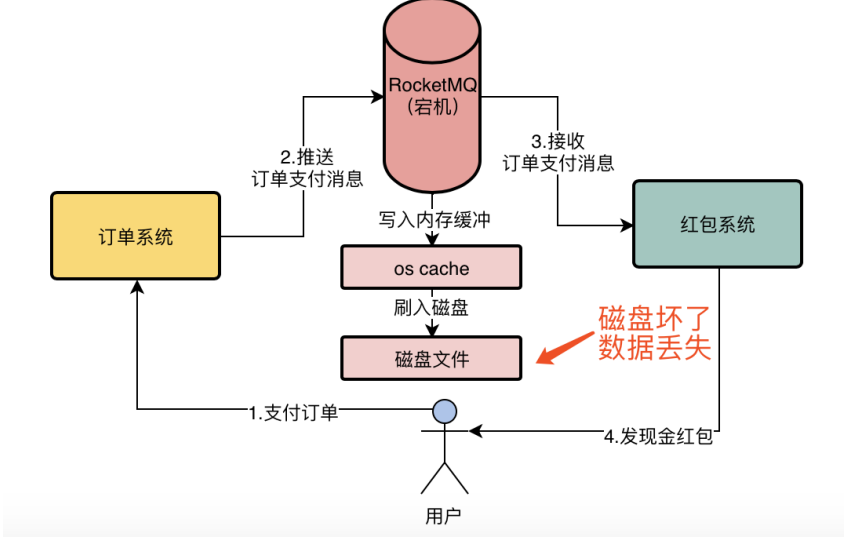
好，假设现在我们写入MQ的一条消息已经稳稳进入Broker所在机器的磁盘文件里了，大家觉得这个时候数据一定不会丢失吗？

显然不能想的那么简单，因为如果你的磁盘出现故障，比如磁盘坏了，你上面存储的数据还是会丢失。

如果大家注意留意近两年的一些技术行业的不起眼的新闻，就会知道，有的互联网公司就是把数据存储在服务器的磁盘上，但是因为没做完善的冗余备份，结果机器磁盘故障就导致那个互联网公司运营几年的核心数据都没了，找不回来了，大量的投资都打了水漂，血淋淋的教训。

所以如果消息进入了broker机器的磁盘之后，万一你实在是点儿背，赶上机器刚好磁盘坏了，可能上面的消息也就都丢失了

我们看下图，清晰的标识出来，磁盘坏了也会导致上面存储的数据丢失。



5、即使红包系统拿到了消息，就一定会丢失了吗？

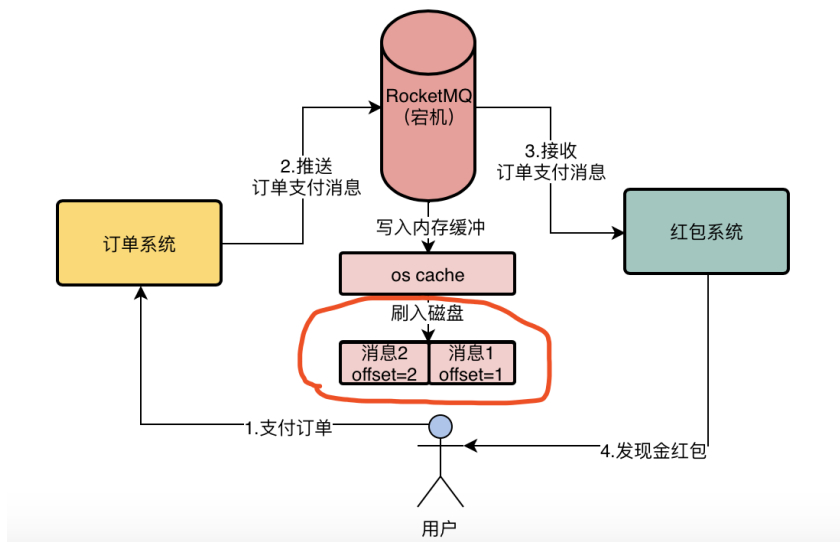
好，我们接着往后看，即使红包系统这个时候顺利从MQ里拿到了一条消息，然后他就一定能安稳的把现金红包发出去吗？

这也是未必的。要解释这个问题，我们就要牵扯到消息的offset这个概念了。

之前其实我们已经给大家在底层原理分析的部分详细解释了MQ底层的存储结构，包括消息的offset的概念

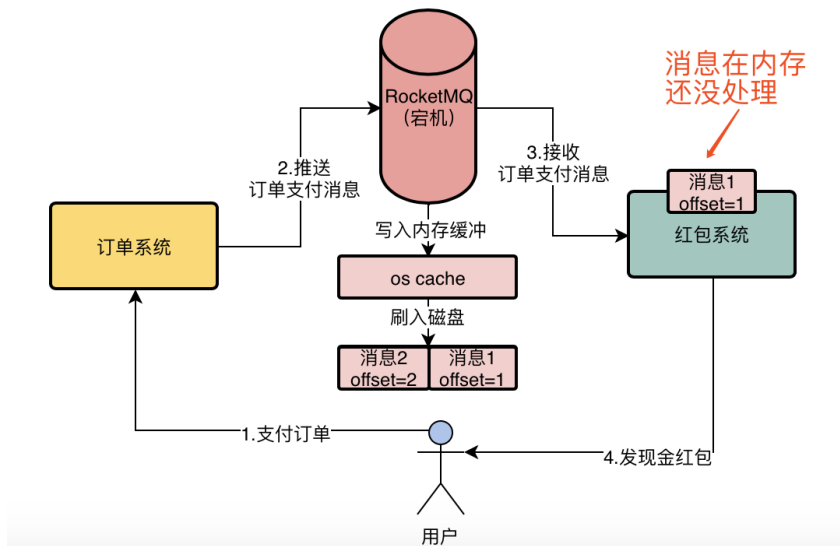
说白了，offset就是代表了一个消息的标识，代表了他的位置

我们给大家举个例子，看下图，假设现在有两个消息，offset分别为1和2。

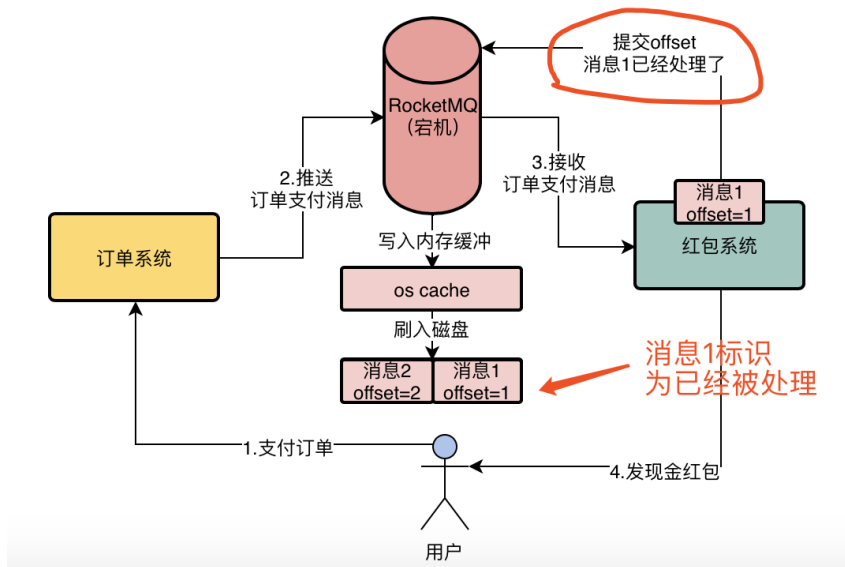


现在我们假设红包系统已经获取到了消息1了，然后消息1此时就在他的内存里，正准备运行代码去派发现金红包呢，但是要注意，此时还没派发现金红包

我们先看下图

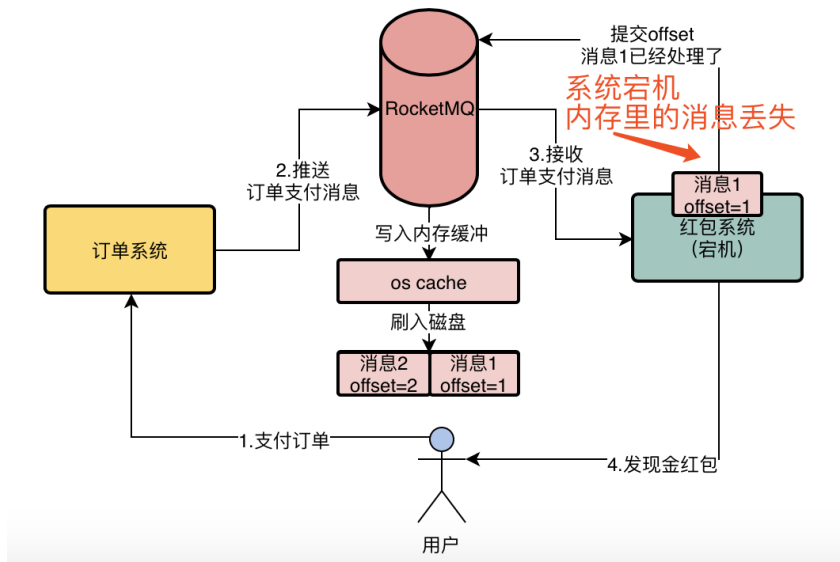


我们都知道，默认情况下，MQ的消费者有可能会自动提交已经消费的offset，那么如果此时你还没处理这个消息派发红包的情况下，MQ的消费者可能直接自动给你提交这个消息1的offset到broker去了，标识为你已经成功处理了这个消息，我们看下图。



接着恰巧在这个时候，我们的红包系统突然重启了，或者是宕机了，或者是可能在派发红包的时候更新数据库失败了，总之就是他突然故障了，红包系统的机器重启了一下，然后此时内存里的消息1必然就丢失了，而且红包也没发出去

我们看下图



## 6、用户支付之后红包到底为什么没发出去呢？

我们学习了今天的文章就可以来解答这个问题了，为什么用户支付了，但是红包没发出去呢？

其实原因有多种可能，比如订单系统推送消息到MQ就失败了，压根儿就没推送过去；

或者是消息确实推送到MQ了，但是结果MQ自己机器故障，把消息搞丢了；

或者是红包系统拿到了消息，但是他把消息搞丢了，结果红包还没来得及发。

如果真的在生产环境里要搞明白这个问题，就必须要打更多的日志去一点点分析消息到底是在哪个环节丢失了？

如果订单系统推送了消息，结果红包系统连消息都没收到，那可能消息根本就没发到MQ去，或者MQ自己搞丢了消息。

如果红包系统收到了消息，结果红包没派发，那就是红包系统搞丢了消息。

## 7、授人以渔：一个小小的思考题

今天给大家留的一个小小的思考题，是跟每个人的工作环境相关的，希望大家想想如下一些问题：

你的系统里用到了MQ吗？

如果用到了，有没有可能会出现消息丢失的问题？

你们之前是否出现过消息丢失的问题？

如果没有遇到过消息丢失问题，你想想你们有没有可能遇到？

如果遇到过，你们是怎么解决的？

这个是跟大家工作相关的问题，没有标准答案，大家踊跃在评论区交流。

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

狸猫技术窝其他**精品专栏**推荐：

[《从零开始带你成为JVM实战高手》](#)

[《21天Java 面试突击训练营》（分布式篇）](#)（现更名为：[互联网Java工程师面试突击第2季](#)）

[互联网Java工程师面试突击（第1季）](#)

**重要说明：**


**如何提问：**每篇文章都有评论区，大家可以尽情在评论区留言提问，我会逐一答疑

**如何加群：**购买了狸猫技术窝专栏的小伙伴都可以加入**狸猫技术交流群**

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）

Copyright © 2015-2019 深圳小鹅网络技术有限公司 All Rights Reserved. 粤ICP备15020529号

---

 小鹅通提供技术支持