

图文 087 由于消费系统故障导致的RocketMQ百万消息积压问题，应该如何处理？

73 人次阅读 2020-02-04 08:33:42

[详情](#) [评论](#)



狸猫技术

由于消费系统故障导致的RocketMQ百万消息积压问题，应该如何处理？

The banner features a large blue title '从零开始' (From Zero to Hero) with 'MySQL实战优化高手' (MySQL Practical Optimization Expert) below it. It includes a QR code and a contact number: '642600657'. The background has a grid pattern.

相关频道

从 0 开始
中间件实践
已更新 9

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

专栏作者：救火队队长
阿里资深技术专家

继《从零开始带你成为JVM实战高手》后，救火队长携新作再度出山，重磅推荐：

(点击下方蓝字试听)

[《从零开始带你成为MySQL实战优化高手》](#)

今天同样是给大家讲解两个比较轻松的技术话题，因为最近几天的内容处于本专栏的一个新旧过渡期，对于原先大纲的90讲的内容，我们基本都已经快要讲完了，但是我们的新版大纲里又新增了30讲的RocketMQ源码分析的内容，所以这两天我们老大纲里最后几讲都是比较简单轻松的话题，很快我们将要进入非常硬核的30讲源码分析的环节。

先给大家说一个RocketMQ如果有百万消息积压了，应该怎么来处理呢？

说到这里，先给大家推荐一下石杉老师的《互联网Java工程师面试突击（第一季）》，在里面就有分析到MQ中间件的百万消息积压的问题以及常见的解决方案。大家可以在狸猫技术窝公众号的知识店铺里免费观看

我这里主要是基于RocketMQ的技术原理给大家分析一下，如果使用RocketMQ遇到百万消息积压的时候，我们应该怎么处理和解决呢？

我们先来思考一下，遇到百万消息积压大概是个什么场景。

先来一个比较真实的生产场景，我们曾经有一个系统，他就是由生产者系统和消费者系统两个环节组成的，生产者系统会负责不停的把消息写入RocketMQ里去，然后消费者系统就是负责从RocketMQ里消费消息。

这个系统在生产环境是有高峰和低谷的，在晚上几个小时的高峰期，大概就会有100多万条消息进入RocketMQ。然后消费者系统从RocketMQ里获取到消息之后，会依赖一些NoSQL数据库去进行一些业务逻辑的实现。

然后有一天晚上就出现了一个问题，消费者系统依赖的NoSQL数据库就挂掉了，导致消费者系统自己也没法运作了，此时就没法继续从RocketMQ里消费数据和处理了，消费者系统几乎就处于停滞不动的状态。

然后生产者系统在晚上几个小时的高峰期，就往MQ里写入了100多万的消息，此时都积压在MQ里了，根本没人消费和处理。

针对这种紧急的线上事故，一般来说有几种方案可以快速搞定他，如果这些消息你是允许丢失的，那么此时你就可以紧急修改消费者系统的代码，在代码里对所有的消息都获取到就直接丢弃，不做任何的处理，这样可以迅速的让积压在MQ里的百万消息被处理掉，只不过处理方式就是全部丢弃而已。

但是往往对很多系统而言，不能简单粗暴的丢弃这些消息，所以最常见的办法，还是先等待消费者系统底层依赖的NoSQL数据库先恢复了，恢复之后，就可以根据你的线上Topic的MessageQueue的数量来看看如何后续处理。

假如你的Topic有20个MessageQueue，然后你只有4个消费者系统在消费，那么每个消费者系统会从5个MessageQueue里获取消息，所以此时如果你仅仅依靠4个消费者系统是肯定不够的，毕竟MQ里积压了百万消息了。

所以此时你可以临时申请16台机器多部署16个消费者系统的实例，然后20个消费者系统同时消费，每个人消费一个MessageQueue的消息，此时你会发现你消费的速度提高了5倍，很快积压的百万消息都会被处理完毕。

但是这里你同时要考虑到你的消费者系统底层依赖的NoSQL数据库必须要能抗拒临时增加了5倍的读写压力，因为原来就4个消费者系统在读写NoSQL，现在临时变成了20个消费者系统了。 慕课都有联系微信642600657

当你处理完百万积压的消息之后，就可以下线多余的16台机器了。

这是一个最最常见的处理百万消息积压的办法，大体思路跟石杉老师在《互联网Java工程师面试突击（第一季）》里讲解的方案是一样的，只不过这里细化到根据RocketMQ的技术原理来讲解。

那么如果你的Topic总共就只有4个MessageQueue，然后你就只有4个消费者系统呢？

这个时候就没办法扩容消费者系统了，因为你加再多的消费者系统，还是只有4个MessageQueue，没法并行消费。

所以此时往往是临时修改那4个消费者的系统的代码，让他们获取到消息然后不写入NoSQL，而是直接把消息写入一个新的Topic，这个速度是很快的，因为仅仅是读写MQ而已。

然后新的Topic有20个MessageQueue，然后再部署20台临时增加的消费者系统，去消费新的Topic后写入数据到NoSQL里去，这样子也可以迅速的增加消费者系统的并行处理能力，使用一个新的Topic来允许更多的消费者系统并行处理。

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）](#)

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](#)）

Copyright © 2015-2020 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

慕课都有联系微信642600657