11. 祭天的支付故障：大促高峰运营后台操作拖死在线交易

大家好，我是隐墨星辰，专注境内/跨境支付架构设计十余年。

今天我们继续聊聊那些“可以拖出去祭天”的线上故障案例。这次不是领导人好，保住我小命一条，而是我不够资格，大老板把我领导的领导给祭天了。

这也是很多年的故事，当时的微服务框架还没有现在这么成熟，限流与服务隔离也没有现在这么先进。

起因只是一次小小的后台查询操作，却在大促高峰期引发了连锁反应，让整个系统陷入瘫痪。

一次大促开始不久，一位运营小姐姐在后台发起了一个后台查询操作，很不幸，触发了慢查询。本来所有在线交易系统都使用缓存以顶住一些读压力，但仍然有小部分请求在缓存中找不到数据，穿透到了数据库，却遇上这条正在阻塞资源的慢查询。结果越来越多请求堆积，数据库CPU瞬间飙升至100%。

DBA尝试kill连接却无济于事，不得已只能重启数据库。可重启后，因为在线交易的请求仍在，缓存依旧被打穿，数据库一恢复服务就再度陷入满载状态，陷入恶性循环。

只能人工先限流，再次重启，慢慢恢复流量。等服务完全恢复，大促高峰已经过去大半。

事后复盘发现：当时在线交易对那部分数据其实并非强依赖，即使拿不到那些数据，也能继续走主流程。但由于缺少弱依赖分析和对应的降级策略，这块数据竟成了全局堵点。

虽然是一次有点久远的故障，背后的思路对现在的系统设计仍然有一些借鉴意义。

先看看问题。

首先是**后台操作与线上数据库耦合**。大促期间的后台查询没有隔离环境，直接在生产DB上执行引发慢SQL，将数据库资源长时间占用，影响在线交易。

其次**缺乏弱依赖降级策略**。这部分数据其实是“弱依赖”，在线交易不拿到也能继续推进。但系统实现成了强依赖，导致前端请求死等结果，让线程和资源耗尽。

同时**还缺少自动化限流手段**，当慢SQL导致数据库处理能力下降，大量请求同时穿透到后端加剧问题。如果有自动化限流策略，可在异常时快速对请求进行削峰填谷，避免资源被透支。

当然最重要是**缓存防击穿策略欠缺**。在高并发场景下，若缓存未命中数据且没有防击穿策略，一旦后端阻塞，所有请求将同时穿透数据库，极易造成资源枯竭。应有预热缓存、单线程代理查

询、请求队列等措施来防止缓存失效时大量请求直击后端。

看到问题，对应的措施就比较简单了。

首先是**后台与线上隔离**。在大促期间禁止非所有必要的后台操作，即使需要也要单独环境或读副本、读缓存，不要直接跑在核心库上。可以在大促时切换后台操作权限，菜单折叠、权限下线，减少意外。

其次是**弱依赖识别与降级**。对主链路中的数据需求进行强弱依赖分析。对于弱依赖数据，一旦获取超时或异常，即刻跳过，不阻塞整个主流程。有损服务好过无服务可用。

然后是**限流和熔断**。引入限流工具，对下游资源的请求数量进行控制。当下游变慢或挂掉时，限流可阻止请求洪流加剧问题。

最后是老生常谈的**缓存防击穿策略**。在高并发下，缓存穿透会将数千上万请求砸向数据库。应提前预热数据，让缓存中长期保留重要信息；对请求加锁或队列，让同一时间只有一个线程去真正请求后端数据，其余等待结果，从而避免瞬间集中冲击。

在这个事故之前，我是没有想到一个简单的后台操作也会搞死在线交易。从另外一个侧面看，当请求量足够大时，很多隐藏的问题就会暴露，也就是所谓的量变引起质变。

一将功成万骨枯，一个个线上故障成就了一个个经验丰富的工程师。

这是《支付通识》专栏系列文章中的第（11）篇。

深耕境内/跨境支付架构设计十余年，欢迎关注并星标公众号“隐墨星辰”，和我一起深入解码支付系统的方方面面。

专栏系列文章PDF合集不定时更新，欢迎关注我的公众号“隐墨星辰”，留言“PDF”获取。

隐墨星辰 公众号

10年顶尖境内/跨境支付公司架构经验

著有《图解支付系统设计与实现》

和我一起解码支付系统方方面面

有个支付系统设计与实现讨论群，添加个人微信（yinmon_sc）备注666进入。

隐墨星辰 个人微信

10年顶尖境内/跨境支付公司架构经验

著有《图解支付系统设计与实现》

备注666进支付讨论群