10. 祭天的支付故障：雪崩

大家好，我是隐墨星辰，专注境内/跨境支付架构设计十余年。

我是从传统行业转互联网支付，刚进入支付行业那几年，经历了很多线上故障。

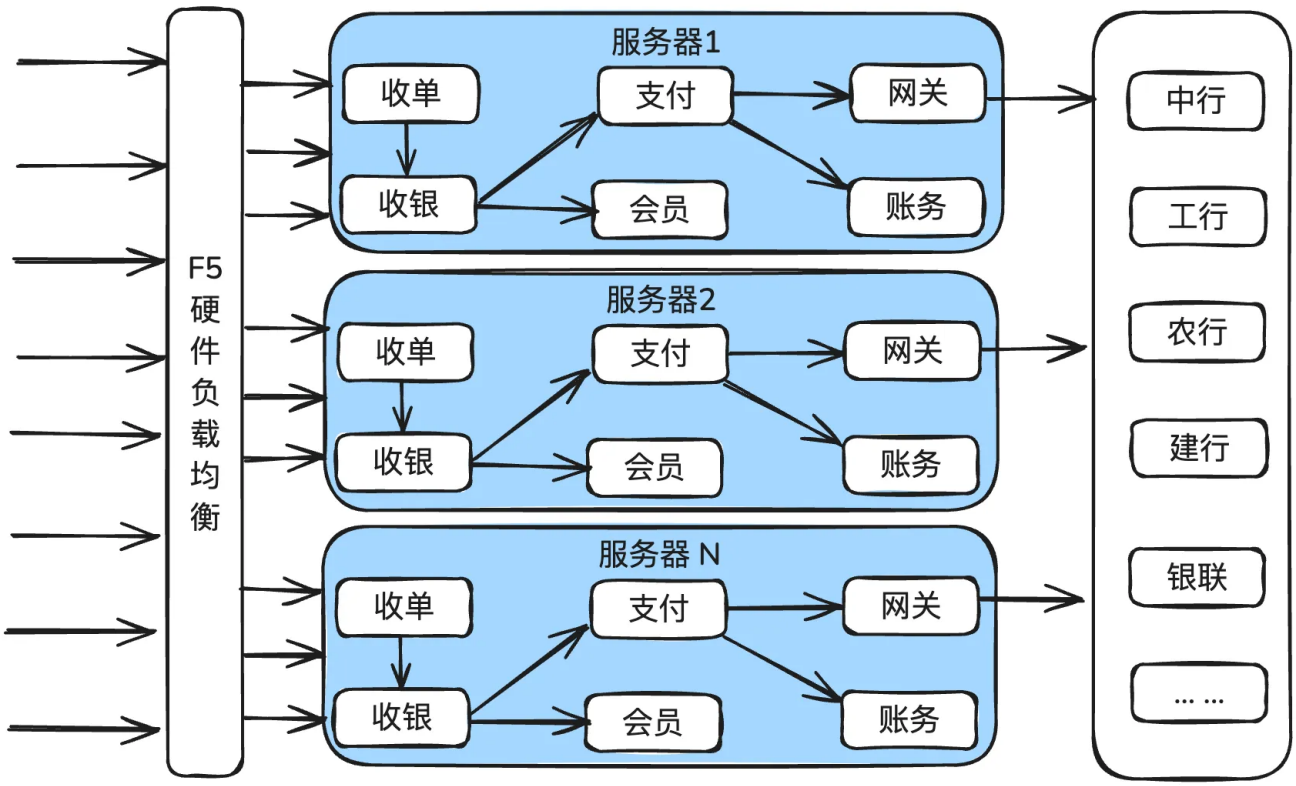
除了上一篇说到的渠道短号导致资损几十万的故障，还有一个可以拖出去祭天的故障，导致的后果严重性远超上一个：**不但整个支付系统宕机，错乱的数据就有几十万条，修数就修了四天三夜**。

好在领导真的人好，保住我小命一条，但是领导自己的年终和晋升给祭天了。

故事从好些年前说起，那时分布式应用已经起来，但是微服务还没有成气候。

支付系统由台服务器组成集群服务，每台服务器都有完整的全应用部署，包括收银支付，收单，渠道网关，会员，商服等全部子应用，从入口开始直到调用渠道，全部在一台服务器内部搞定，各子应用之间的服务是独立的，通过socket调用。

整体架构如下图所示。



那时还是直连银行，日常没有什么问题，经过几次大促的洗礼都平安度过。

直到有一次大促，流量特别高，大促开始没几分钟，就有一家银行出现慢处理问题，具体表现为**平时1S就返回，变成了平均2到5分钟才返回结果**。

很不幸，当前**网关的配置有缺陷，导致配置的超时时间没有生效**，所有与这家银行的请求平均耗时2到5分钟才释放，导致**网关的线程耗尽**。

悲剧由此拉开序幕。

首先是网关的线程耗尽，**导致其它银行的请求也得不到及时的处理**。

然后是内部各子应用之间全部是**同步调用，网关的问题快速蔓延到了上游应用，上游各子应用的线程数也被耗尽，雪崩出现，整个支付系统无法正常处理业务请求**。

那也是我职业生涯中第一次听说“雪崩”这个名词可以用在技术领域。

如前面所说，当时数据订正就花了四天三夜，每天在公司的行军床上躺2小时。

前车之鉴，后车之师。还是有一些教训可以总结一下。

1. **强制设置合理的超时时间，并验证有效**。这里面包含2层意思：
   a. 合理的超时时间。比如不同的外部渠道以及同一渠道不同的接口，响应时间都是不一样的，需要统计90分位，95分位，98分位等多个时间。一般覆盖95分位就差不多。
   b. 需要验证有效。很多技术参数在表面上看是设置了，但是实际可能不是预期那样。就拿http来说，就有连接超时，写入超时，读取超时等多个超时参数。一定需要模拟测试验证，达到预期效果。
2. **服务隔离**。比如为不同的渠道做线程池隔离。一个渠道挂了不影响另外一个渠道。
3. **健全的服务降级、熔断、限流能力**。现在的微服务框架基本都有自动化的服务降级、熔断、限流能力，但是需要提前做好配置。提供有损服务好过完全无服务可用。
4. **用好同步受理异步处理机制**。比如最外部的渠道网关，因为外部渠道的耗时都比较长，就采用同步受理异步处理的模式：先把交易信息收下来，落库，马上返回给上游受理成功，然后再起异步线程把请求发出去。

   这样有两个好处：

   1）**可以为网关单独扩大线程池的最大线程配置**。因为网关已经变为IO密集型应用。

   2）**网关的慢处理（比如耗时2S），不影响上游**，上游可以毫秒级就处理完自己的业务。
5. **压测和预案**。前者提前发现问题，后者是问题出现后可以指导快速响应。

当交易量足够大，一个小小的问题也有可能被放大到无法承受之重。

无知的人，给平台带来的伤痛最深。

这是《支付通识》专栏系列文章中的第（10）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