



《手写OS操作系统》小班二期招生，全程直播授课，大牛带你掌握硬核技术！

点此查看

·模式 – 函数

下载

插件开发、

n插件

·包装设计

·生成工具开

马结构

从所有教程的词条中查询...

首页 > 慕课教程 > Go工程师体系课全新版 > 16. grpc的超时和重试



bobby · 更新于 2022-11-16

15. 什么是服务雪崩 17. 常见的幂等性解决方案

超时

timeout是为了保护服务，避免consumer服务因为provider 响应慢而也变得响应很慢，这样consumer可以尽量保持原有的性能。

重试

如果provider只是偶尔抖动，那么超时后直接放弃，不做后续处理，就会导致当前请求错误，也会带来业务方面的损失。对于这种偶尔抖动，可以在超时后重试一下，重试如果正常返回了，那么这次请求就被挽救了，能够正常给前端返回数据，只不过比原来响应慢一点。重试可以考虑切换一台机器来进行调用，因为原来机器可能由于临时负载高而性能下降，重试会更加加剧其性能问题，而换一台机器，得到更快返回的概率也更大一些。

幂等

如果允许consumer重试，那么provider就要能够做到幂等。同一个请求被consumer多次调用，对provider产生的影响是一致的。而且这个幂等应该是服务级别的，而不是某台机器层面的，重试调用任何一台机器，都应该做到幂等。

15. 什么是服务雪崩 17. 常见的幂等性解决方案

我要提出意见反馈

