



《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

全部开发者教程

mono-repo)

7. go代码的检测工具

8. go中常见的错误

第22周 设计模式和单元测试

1. go最常用的设计模式 - 函数选项

2. 单例模式和懒加载

3. 测试金字塔

第23周 protoc插件开发、cobra命令行

1. protoc调试源码

2. protoc自定义gin插件

第24周 log日志包设计

日志源码

第25周 ast代码生成工具开发

错误码

第26周 三层代码结构

通用app项目启动

 bobby · 更新于 2022-11-16

◀ 上一节 14. go操作Rock... 16. grpc的超时... 下一节 ▶

定义

服务雪崩效应是一种因“服务提供者的不可用”（原因）导致“服务调用者不可用”（结果），并将不可用逐渐放大的现象。如下图所示：



上图中, A为服务提供者, B为A的服务调用者, C和D是B的服务调用者. 当A的不可用,引起B的不可用,并将不可用逐渐放大C和D时, 服务雪崩就形成了。

形成原因

服务雪崩的过程可以分为三个阶段：

1. 服务提供者不可用；
2. 重试加大请求流量；
3. 服务调用者不可用；

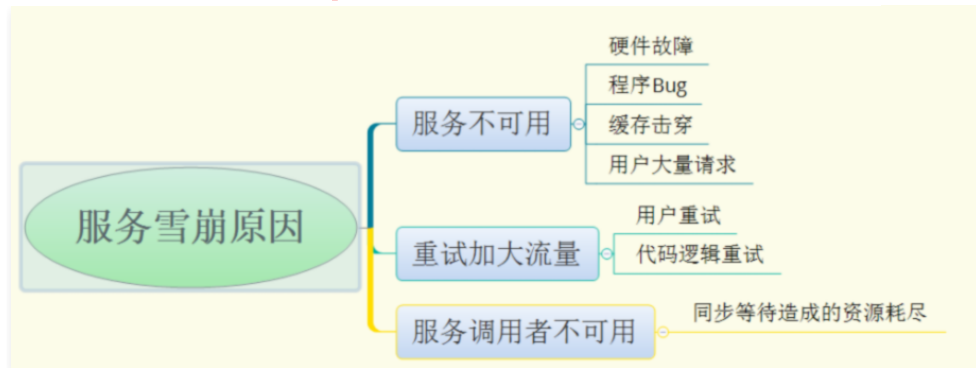
服务雪崩的每个阶段都可能由不同的原因造成，总结如下：

意见反馈

收藏教程

标记书签

↑



应对策略

一开始你的服务能考虑抗住高并发吗？

成本的增加： 开发成本 硬件成本 - 我要抗住高并发 - **10W - 500**

即使是淘宝这种 平时的流量和双11的流量差异就很大

我们现在考虑到我今年可能出现一次高并发 **1W** - 我全年的服务都部署成可以抗住**1W**并发 - **500 - 1k**

某个时候出现了流量的猛增 - **2k**

**

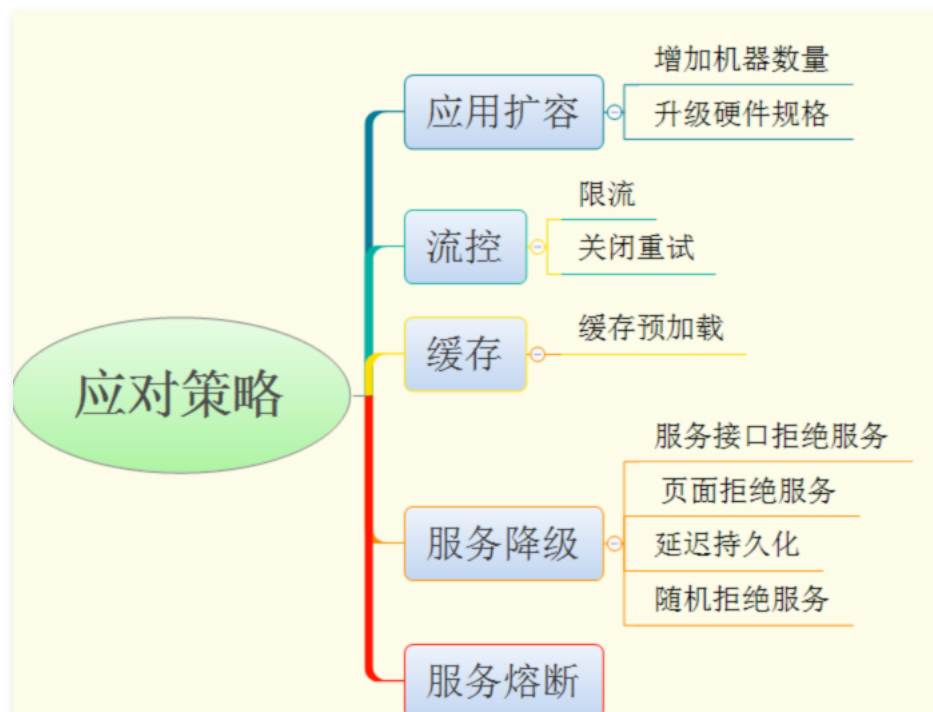
限流 - 2k 但是我的服务能力只有1k, 所以这个时候多出来的流量怎么办： 1. 拒绝 2. 排队等待
用户体验不太好： 当前访问用户过多，请稍后重试 和 你的服务直接挂了

用户体验降级了 - 原本是访问流畅，下单流畅 -> 当前访问用户过多，请稍后重试

熔断 - 比如A服务访问B服务，这个时候B服务很慢 - B服务压力过大，导致了出现了不少请求错误，调用方很容易出现一个问题： 每次调用都超时 2k, 结果这个时候数据库出现了问题， 超时重试 - 网络 2k的流量突然变成了3k

这让原本就满负荷的b服务雪上加霜，如果这个时候调用方有一种机制： 比如说 1. 发现了大部分请求很慢 - 50%请求都很慢， 2. 发现我的请求有50%都错误了 3. 粗我数量很多，比如1s出现了20个错误

熔断 - 1. 保险丝 2. 股市熔断



💬

?

📱

😊