



面向互联网的性能测试案例分享

王磊@百度
wanglei@baidu.com





测试结果与实际情况不符



✓ 案例重现

– 测试对象：检索模块

– 表现

- 线下测试，当请求量为400条/秒时，机器的cpu idle为40%
- 线上服务，当请求量为250条/秒时，机器的cpu idle为20%

– 后果

- 本次升级项目失败，回滚项目



- ✓ 疑问？
 - 线下测试中为何没有测试出来？

- ✓ 如何找到原因
 - 测试方法
 - 测试数据
 - 词表
 - 数据量
 - 测试环境



✓ 问题复现

– 方法

- 对比排查

– 顺序

- 由极可能发生到最不可能发生，逐一排查

✓ 本案例的结论

– 上有模块一个配置项错误





OS对性能测试的影响



✓ 案例重现

– 测试对象：提供垂直类检索的子系统

– 表现

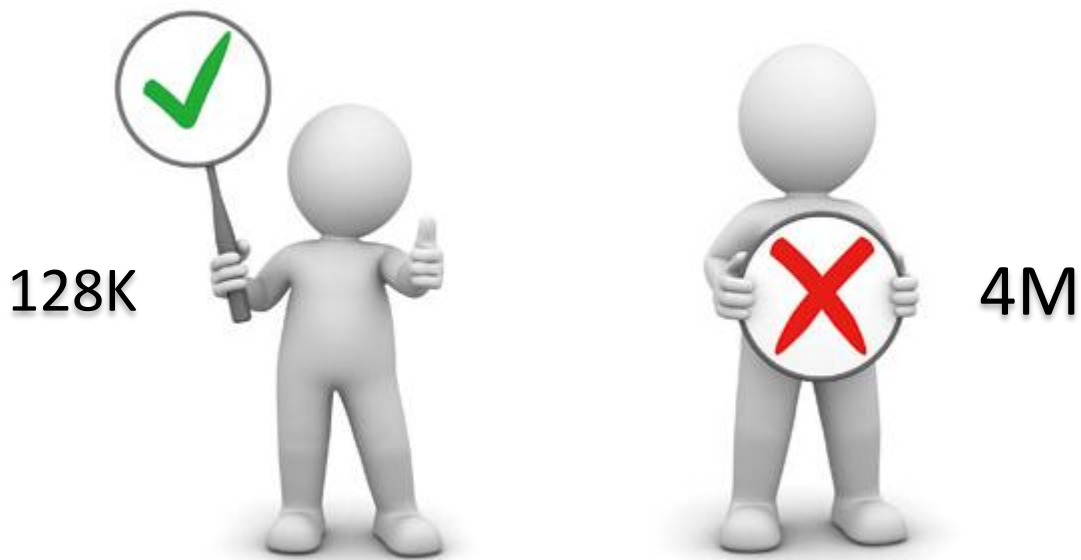
- 线下做性能测试，发现3台机器的IO负载特别高
- 换到其他机器上，没有问题

✓ 初步排查

- Raid 一样
- 异常机器：r/s大约1000多次，带宽rkB/s大约140MB，svctm约0.5ms，因此IO利用率接近100%（%util）
- 正常机器：r/s大约150多次，带宽rkB/s大约5MB，svctm约4ms，因此IO利用率接近60%（%util）

✓ 进一步分析

- 该模块的功能
- 读取的方式





意想不到的
配置

系统知识



网络环境对性能的影响



✓ 案例重现

– 测试对象：2个模块组成的检索子系统

– 表现

- 上游模块（A），请求下游模块数据总是连接超时（B）

– 做过什么？

- 数据量变大后，增加B模块数量
- 一个A模块连接B模块的数量变多



- ✓ 超时问题排查顺序
 - 模块的超时时间配置
 - 网络带宽是否合适
 - 网络模型是否合适
 - 程序是否有bug
 - OS的限制
 - **网络硬件？**



✓ 具体分析

– 交换机的情况

- A连接100个B，A的接收带宽1000Mbps，100个B的发送带宽为 100×1000 Mbps

– 原理

- 原因何在
- 以前为何没事





意想不到的
配置

系统知识

网络环境



不良代码对性能的影响



✓ 案例重现

– 测试对象：一个检索模块

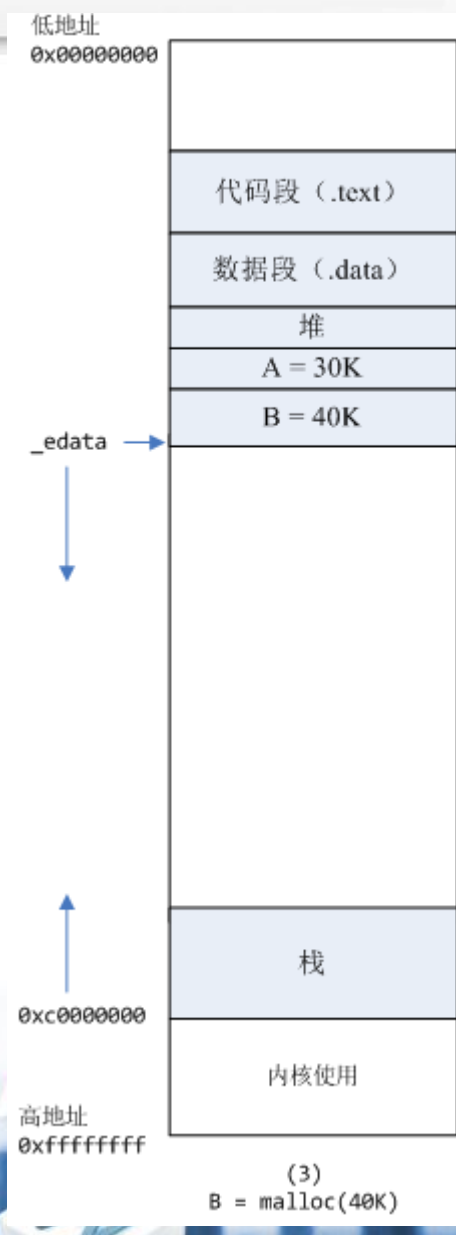
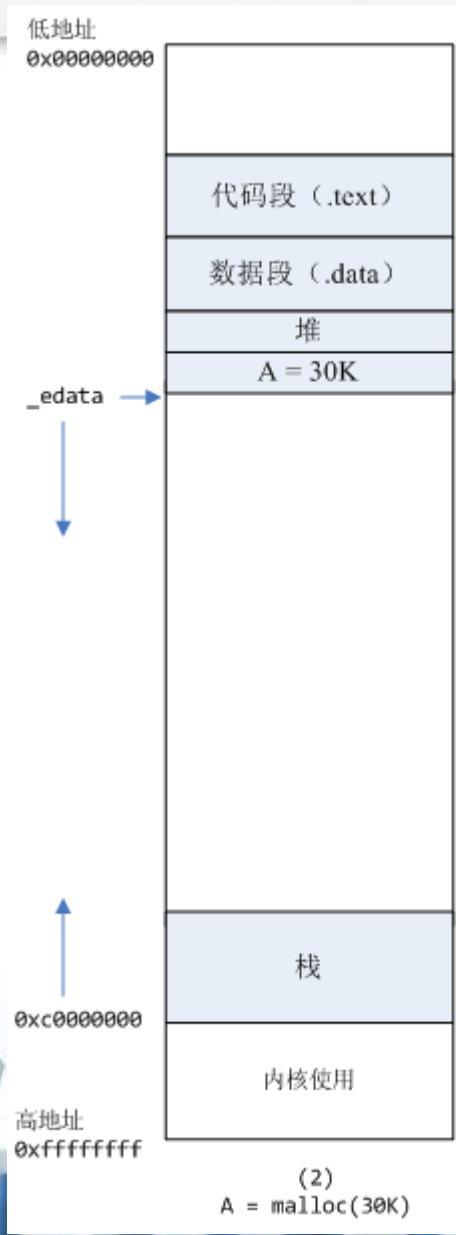
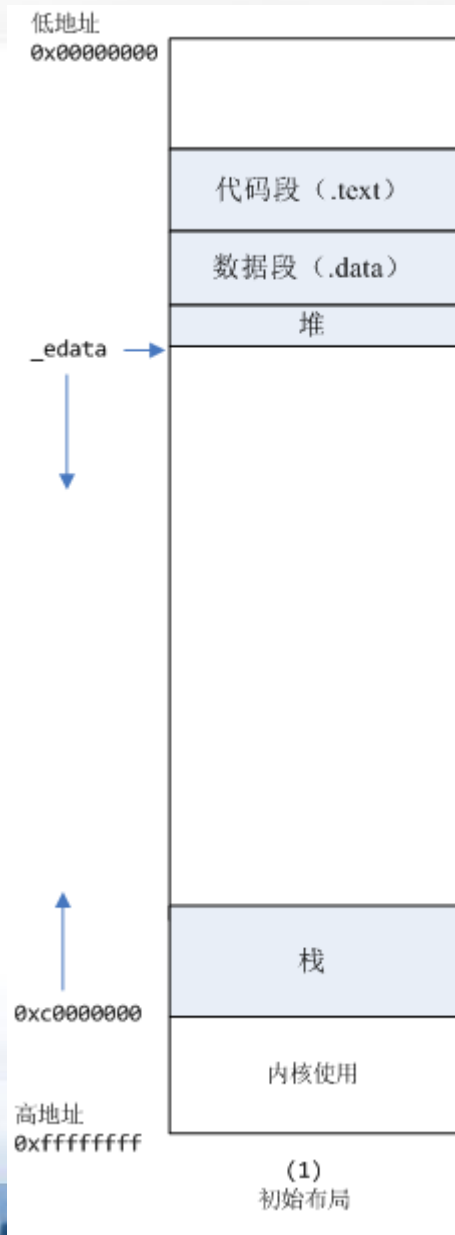
– 表现

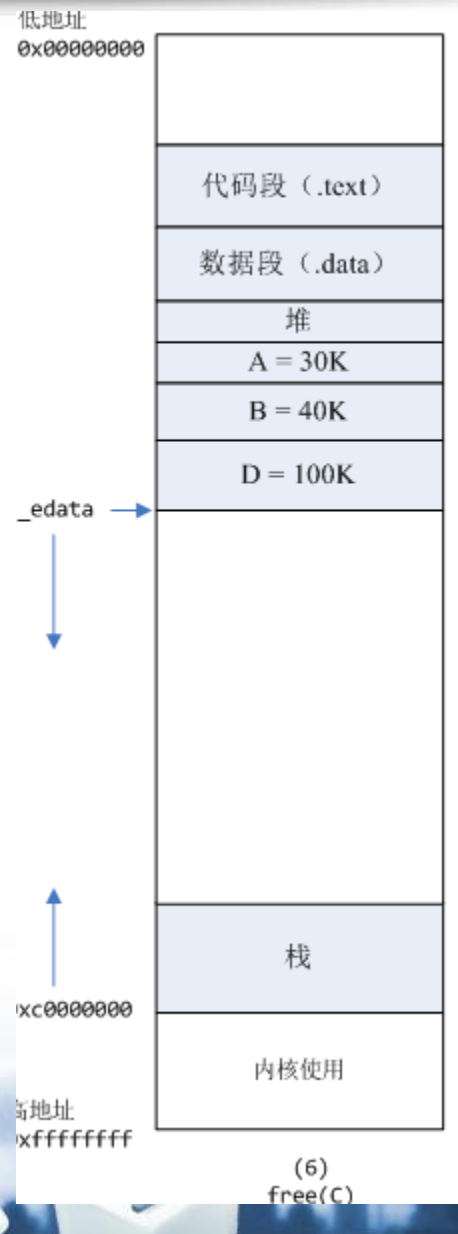
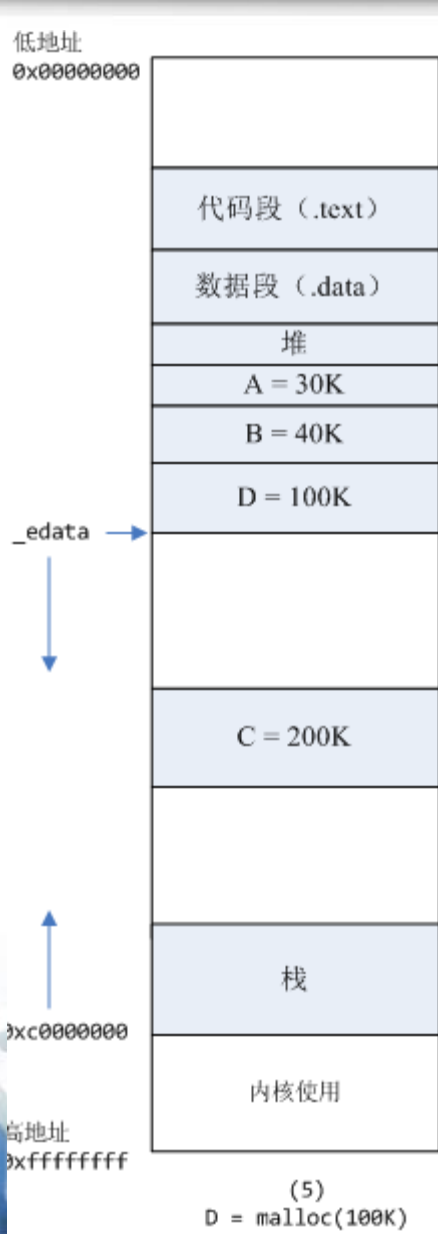
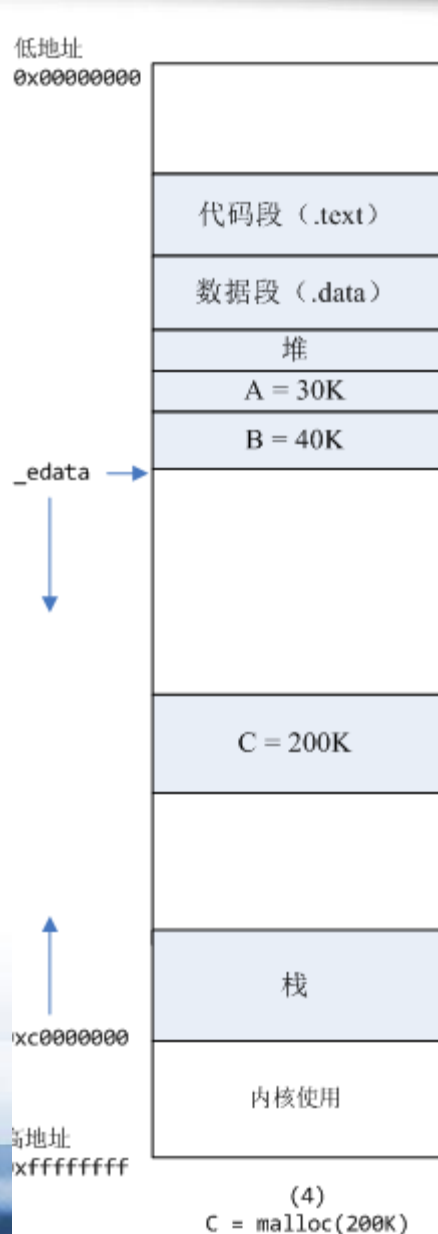
- 性能表现与预期的相差较大
- 进程的系统态CPU消耗20，用户态CPU消耗10，系统idle大约70
- 发现majflt每秒增量为0，而minflt每秒增量大于10000

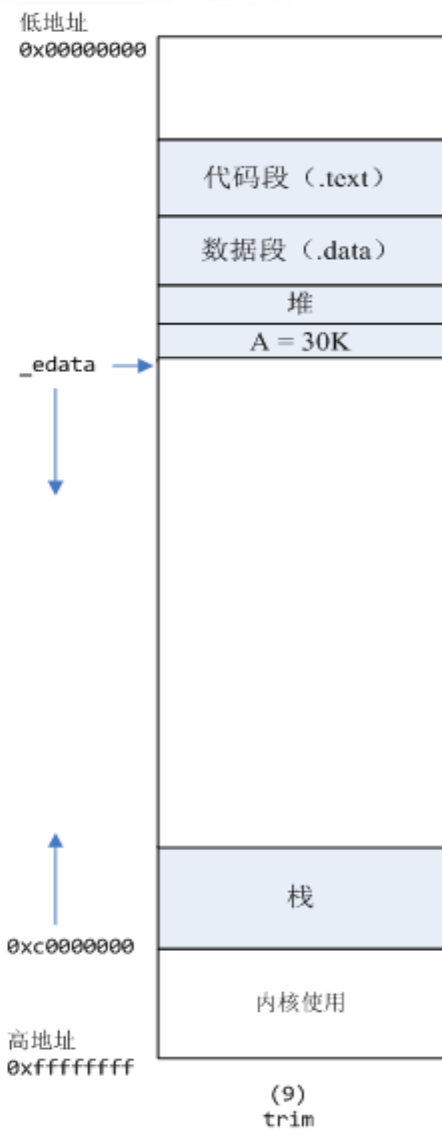
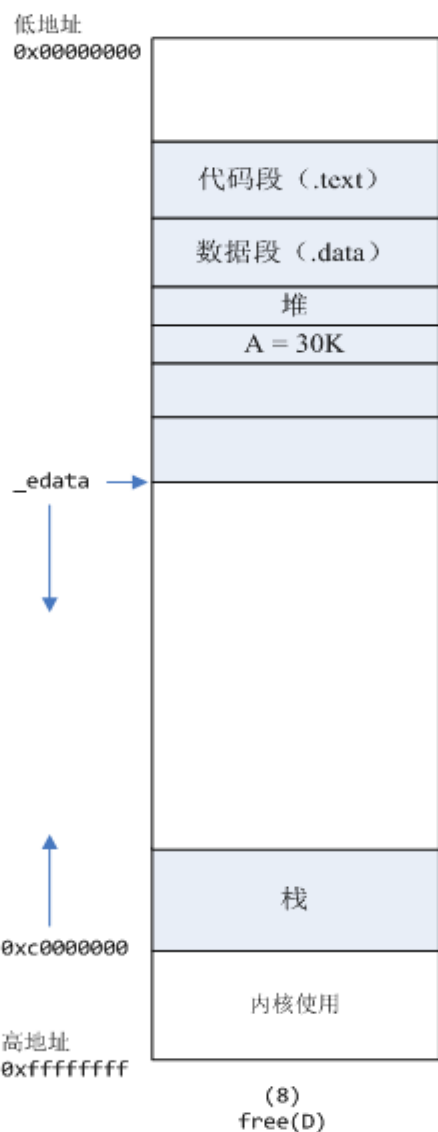
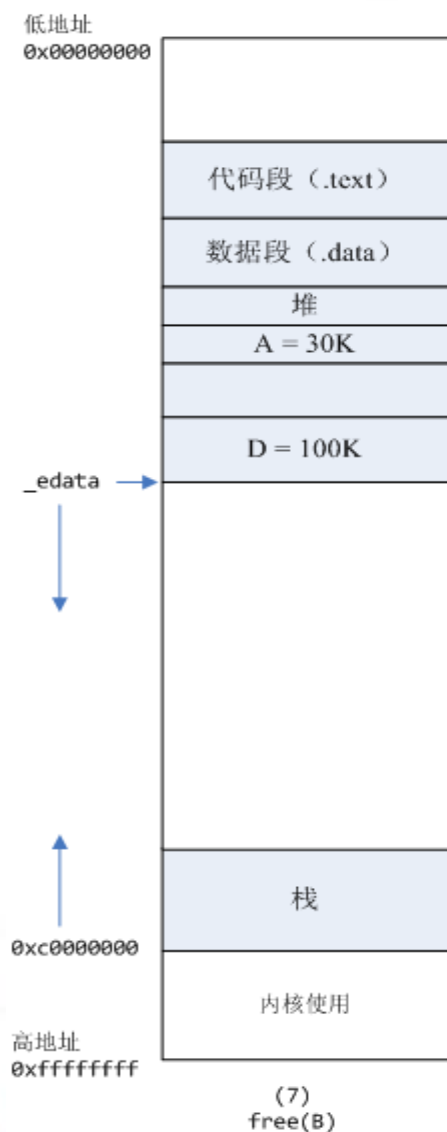
Majflt minflt



- 1 检查要访问的虚拟地址是否合法
- 2 查找/分配一个物理页
- 3 填充物理页内容（读取磁盘）
- 4 建立映射关系(虚拟地址到物理地址)
- 5 重新执行发生缺页中断的那条指令







✓ 代码

- 一个请求来，用malloc分配2M内存，请求结束后free这块内存。
- 日志显示分配内存语句耗时10us，平均一条请求处理耗时1000us

✓ 性能差的原因

- 内存申请造成缺页中断
- 因为这种缺页不需要磁盘，所以是minfit
- 缺页中断在内核态执行，进程的内核态cpu消耗很大



意想不到的
配置

系统知识

网络环境

过硬的编程



谢谢！

