

天启项目 jportal-auth 是认证中心, 整个平台大部分接口会经过这里

业务系统首先暴出问题, 说开发环境崩了, 我立马着手查询服务器日志, 发现堆溢出, 整个服务器内存也只剩下几十兆,

为了尽快回复使用, **马上重启服务**, 以解燃眉之急. 以为重启后就能慢慢排查问题了,

重启后还是无法使用, 马上**再检查内存**, 发现内存情况正常, 再转向**检查磁盘**, 结果显示占用 99%, 磁盘无法写入了, 马上用 **find 和 xargs** 命令查询较大的日志文件并删除, 让磁盘恢复一些, 这才能让业务系统勉强使用

现在转身排查堆溢出的问题, 不到半小时, 堆再次溢出, 群里再次炸开锅, 短短半小时连续堆溢出, 发现服务器又只剩下几十兆了, 通过**top命令**看到, 大部分内存一直被 jportal-auth 占用, 业务系统的人也反馈, 校验 token 接口失败, 监听了一下日志, 发现 token 验证接口被频繁调用, 大致上每秒一百多次, 日志不停的刷, (后来发现是业务系统有轮询机制). 可是校验 token 接口逻辑十分简单, 就是从 redis 查出来然后再比对, 所以初步判断是请求太过频繁, 导致对象无法释放, 所以直接**调大 jvm 参数 -xms 和 -xmx 参数**, 先让他们用着,

然后用 mat 分析工具, 把 jportal-auth.hprof 文件打开, 分析里面具体哪个对象不释放, 打开后发现全是 http, output 这种, 完全没有自定义对象的影子, 打开对象树 (dominator tree) 看到是每个请求占用内存大, 全是 byte, 打开它的对象引用, 发现是 inputbuffer 占用特别大, 很奇怪为什么是这种, 这种应该是 tomcat 级别的问题, 多看了几个请求, 发现每个请求都占用同样大小的内存, 接近 10M, 这就很奇怪了, 怎么每次都刚好是 10M 呢, 联想到刚好又是 inputbuffer 占用大, 检查了一下每个请求的参数, 发现都是正常的基本参数, 那就巧合了, 忽然想到是请求头的大小固定了, 赶紧检查配置, 发现真的设置了最大的请求头大小, 麻溜的赶紧减小这个数值

总结:

原因:

1. 设置了太大的请求头
2. 业务系统十分频繁的调用
3. 磁盘爆满

解决思路:

1. 检查内存/磁盘使用情况
2. 适当调整最大/小 堆大小
3. 使用工具分析具体原因, 对症下药

MAT 工具使用

在jdk启动加参数中加： `-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=D:\dump\`，然后应用启动后出现内存异常则会自动导出dump文件，默认的文件名是：`java_pid<进程号>.hprof` (也可以指定名称, 在路径中指定名称)。

<https://blog.csdn.net/aileitianshi/article/details/90147109>

<https://www.jianshu.com/p/c6e2abb9f657>