

原 问题分析: java.lang.OutOfMemoryError:unable to create new native thread

标签: JAVA jvm 线程

2015-09-08 09:52 444人阅读 评论(0) 收藏 举报

版权声明: 本文为博主原创文章, 未经博主允许不得转载。

目录(?)

[-]

1. 背景
2. 分析
3. 结论

背景

后台系统抛出OOM错误, 操作无法正常进行。调用后台日志, 发现错误信息

为: java.lang.OutOfMemoryError:unable to create new native thread

分析

- 1、一般OOM比较常见的原因是堆内存不够或者持久代内存不够导致, 但这个问题是因为无法创建本地线程。
- 2、最初怀疑JVM的堆内存或持久代内存耗尽了, 使用jmap查看JVM内存信息, 发现一切正常。(有可能jmap等命令根本无法执行。)
- 3、通过free、top命令查看系统内存, 发现系统物理内存耗尽了。使用jmap命令将JVM的信息dump到文件, 使用JVisualVM分析, 发现有大量的线程处于等待状态, 并且这些线程都是ThreadPool。通过执行下面命令, 可以获知该进程下的线程数:

```
01. ps -L -p [pid] | wc -l
```

发现线程数量有几千个, 线程数量没有达到操作系统限制的数量, 但是内存不够用了。

- 4、根据获得的线索, 怀疑代码里对线程池没有正确使用。分析代码后发现, 线程池对象没有使用static标识, 导致每次在new一个新的对象时就会创建一个线程池。由于使用的是50个线程的固定线程池, 每次任务运行完后, 线程并不会关闭, 而是处于等待状态。因此, 这样一样, 调用次数过多, 就会导致创建很多个线程, 直至OOM。

线程池对象持有的线程泄漏引发OOM
线程池对象必须声明为类变量

结论

- 1、其实分析中的第二步没有必要进行。分析JVM的内存区域: 堆内存、持久代、虚拟机栈、本地方法栈、Direct内存。我们用jmap只能查看堆内存和持久代内存。线程直接使用的是系统物理内存, 用jmap无法看出来。
- 2、我们在配置-Xms、-Xmx参数时, 虽然指定了堆内存的大小, 但是JVM在启动时, 并不会一下申请这么多内存。而是在创建新的对象时, 找操作系统申请内存。同时, 在GC后, JVM并不会将申请下来的内存返回给

操作系统，而是由它自身来进行管理。

3、堆内存配置过大，JVM可用的剩余物理内存就会少。因此，需要综合考虑系统的情况，合理分配JVM的堆内存，不然就容易出现各类OOM了。本次遇到的这个问题就是因为配置有问题：操作系统共有4G的内存，结果堆内存就设置了3G，如果JVM把3G的内存吃完了，也就剩下1G的内存供线程、缓冲区等其他用户使用，所以明显容易出问题。