

在通往牛逼的路上，我一路狂奔！

永久域名 <http://chenhongwei0924.iteye.com>



chenhongwei0924

浏览: 1938 次

性别:

来自: 北京

我现在离线

详细资料 留言簿

搜索本博客

最近访客 [>>更多访客](#)

- [haitaophehe](#)
- [yan578351314](#)
- [hawk0708](#)
- [xredman](#)

- 博客分类
- [全部博客 \(61\)](#)
 - [Java \(1\)](#)
 - [Oracle \(9\)](#)
 - [Linux \(0\)](#)

我的相册

[http_imgload](#)
共 1 张

我的留言簿 [>>更多留言](#)

- 路过打酱油

-- by [Firklaag](#)

tomcat内存溢出总结3

文章分类:Java编程

每一个32位的进程最多可以使用2G的可用内存，因为另外2G被操作系统保留。这里假设使用1.5G给JVM，那么还余下500M可用内存。这500M内存中的一部分必须用于系统dll的加载，那么真正剩下的也许只有400M，现在关键的地方出现了：当你使用Java创建一个线程，在JVM的内存里也会创建一个Thread对象，但是同时也会在操作系统里创建一个真正的物理线程(参考JVM规范)，操作系统会在余下的400兆内存里创建这个物理线程，而不是在JVM的1500M的内存堆里创建。在jdk1.4里头，默认的栈大小是256KB，但是在jdk1.5里头，默认的栈大小为1M每线程，因此，在余下400M的可用内存里边我们最多也只能创建400个可用线程。

这样结论就出来了，要想创建更多的线程，你必须减少分配给JVM的最大内存。还有一种做法是让JVM宿主在你的JNI代码里边。

给出一个有关能够创建线程的最大个数的估算公式：

(MaxProcessMemory - JVMMemory - ReservedOsMemory) / (ThreadStackSize) = Number of threads

对于jdk1.5而言，假设操作系统保留120M内存：
1.5GB JVM: (2GB-1.5Gb-120MB)/(1MB) = ~380 threads
1.0GB JVM: (2GB-1.0Gb-120MB)/(1MB) = ~880 threads
在2000/XP/2003的boot.ini里头有一个启动选项，好像是：/PAE /3G，可以让用户进程最大内存扩充至3G，这时操作系统只能占用最多1G的虚存。那样应该可以让JVM创建更多的线程。
因此这种情况需要结合操作系统进行相关调整。

因此：我们需要结合不同情况对tomcat内存分配进行不同的诊断才能从根本上解决问题。

参考资料（从这些资料中受益良多）：
<http://www.javaeye.com/topic/80620?page=1>
<http://ggmm.blog.sohu.com/117545379.html>
<http://hi.baidu.com/hexiong/blog/item/16dc9e518fb10c2542a75b3c.html>
http://www.wujianrong.com/archives/2006/12/javalangoutofmemoryerror_permg.html

11:24 | [浏览 \(164\)](#) | [评论 \(0\)](#) | [相关推荐](#) [▶ MORE](#)

评论

发表评论

文章已被作者锁定，不允许评论。

- [chenhongwei0924](#) 写道 来给老
爷唱个 怒了 你妹
-- by [Firklaag](#)

其他分类

- [我的收藏](#) (197)
- [我的代码](#) (0)
- [我的论坛主题帖](#) (0)
- [我的所有论坛帖](#) (20)
- [我的精华良好帖](#) (0)

最近加入群组

- [高级语言虚拟机](#)
- [javaeye湖南帮](#)
- [Ubuntu For Fun](#)

存档

- [2011-04](#) (1)
- [2011-03](#) (14)
- [2011-02](#) (19)
- [更多存档...](#)

评论排行榜

- [抽象工厂模式【ABSTRACT
FACTORY PATTERN】...](#)
- [抽象工厂模式【ABSTRACT
FACTORY PATTERN】...](#)
- [sql1](#)
- [抽象工厂模式【ABSTRACT
FACTORY PATTERN】...](#)
- [sql7](#)

