

java程序性能分析之thread dump和heap dump - nickTimer - 博客园

一. dump基本概念

在故障定位(尤其是out of memory)和性能分析的时候,经常会用到一些文件来帮助我们排除代码问题。这些文件记录了JVM运行期间的内存占用、线程执行等情况,这就是我们常说的dump文件。常用的有heap dump和thread dump(也叫javacore,或java dump)。我们可以这么理解:heap dump记录内存信息的,thread dump是记录CPU信息的。

heap dump:

heap dump文件是一个二进制文件,它保存了某一时刻JVM堆中对象使用情况。HeapDump文件是指定时刻的Java堆栈的快照,是一种镜像文件。Heap Analyzer工具通过分析HeapDump文件,哪些对象占用了太多的堆栈空间,来发现导致内存泄露或者可能引起内存泄露的对象。

thread dump:

thread dump文件主要保存的是java应用中各线程在某一时刻的运行的位置,即执行到哪一个类的哪一个方法哪一个行上。thread dump是一个文本文件,打开后可以看到每一个线程的执行栈,以stacktrace的方式显示。通过对thread dump的分析可以得到应用是否“卡”在某一点上,即在某一点运行的时间太长,如数据库查询,长期得不到响应,最终导致系统崩溃。单个的thread dump文件一般来说是没有什么用处的,因为它只是记录了某一个绝对时间点的情况。比较有用的是,线程在一个时间段内的执行情况。

两个thread dump文件在分析时特别有效,因为它可以看出在先后两个时间点上,线程执行的位置,如果发现先后两组数据中同一线程都执行在同一位置,则说明此处可能有问题,因为程序运行是极快的,如果两次均在某一点上,说明这一点的耗时是很大的。通过对这两个文件进行分析,查出原因,进而解决问题。

二. 利用JDK自带的工具获取thread dump文件和heap dump文件

使用的JDK工具在JDK_HOME/bin/目录下,使用到jmap和jstack这两个命令。

1. 获取heap dump文件

windows下切换到JDK_HOME/bin/,执行以下命令: jmap -dump:format=b,file=heap.hprof 2576

linux下切换到JDK_HOME/bin/,执行以下命令: ./jmap -dump:format=b,file=heap.hprof 2576

这样就会在当前目录下生成heap.hprof文件,这就是heap dump文件。

2. 获取thread dump文件

windows下执行: jstack 2576 > thread.txt

linux下执行: ./jstack 2576 > thread.txt

windows/linux则会将命令执行结果转储到thread.txt,这就是thread dump文件。有了dump文件后,我们就能借助性能分析工具获取dump文件中的信息。

3. 如果我们只需要将dump中存活的对象导出,那么可以使用:live参数

jmap -dump:live,format=b,file=heapLive.hprof 2576

执行完后,我们在当前目录C:\Java\jdk1.6.0_27\bin下看到刚生成的三个文件,如下所示:

说明:如上实例的2576是我当前需要分析的java进程PID,关于Windows下如何获得指定的JAVA时空程PID可参考: <http://bijian1013.iteye.com/blog/2221238>

三. 使用工具分析java heap dump文件

现在我们使用一些图形化工具,来帮助我们分析文件中的信息,有效地定位问题。

1. 使用JDK自带的jhat命令

jhat是用来分析java堆的命令,可以将堆中的对象以html的形式显示出来,包括对象的数量,大小等等,并支持对象查询语言。

jhat -port 5000 heap.hprof

当服务启动完成后,我们就可以在浏览器中,通过http://localhost:5000/进行访问,如下所示:

2. 使用eclipse MAT工具

一般来说,应用程序的dump文件都是很大的,jdk自带命令难以分析这些大文件。在实际的生产环境下,我们必

须要借助第三方工具，才能快速打开这些大文件，进行分析定位。eclipse memory analyzer是一款优秀的heap分析工具，能够帮我们快速定位内存泄露问题。