

第八章第二节 动态程序分析方法与工具

Outline

- Java性能调优工具:
 - jstat
 - jmap
 - jhat
 - Visual VM
 - MAT
- Memory dump
- Stack trace

Notes

Java性能调优工具

- **Jstat**: 获取JVM的Heap使用和GC的性能统计数据, 命令如-gcutil
- **Jmap**: 输出内存中的对象分布情况 如: jmap -clstats
- **Jhat**: 导出heap dump, 浏览/查询其中的对象分布情况
- **jstack**: 获取Java线程的stack trace 具体用途如下:
 - 定位线程出现长时间停顿的原因, 如多线程间死锁、死循环、请求外部资源 导致的长时间等待等。
 - 线程出现停顿的时候通过jstack来查看各个线程的调用堆栈, 就可以知道没 有响应的线程到底在后台做什么事情, 或者等待什么资源。
- **Visual VM**: 提供了一个可视化界面, 用于查看Java应用程序在JVM上运行时的详细信息, 使用各种技术, 包括jvmstat, JMX, Serviceability Agent (SA) 和Attach API等
- **MAT**: 内存堆导出文件的分析工具, 生成饼状图等, 能够对问题发生时刻的系统内存状态获取一个整体印象, 找到最有可能导致内存泄露的对象, 进一步查看其是否有异常行为。

Memory Dump (堆转储文件)

转自 [爪哇堂的博客](#)

正如Thread Dump文件记录了当时JVM中线程运行的情况一样，Heap Dump记录了JVM中堆内存运行的情况，可使用jmap或JConsole命令生成，jhat分析。

【使用 jmap 命令生成】

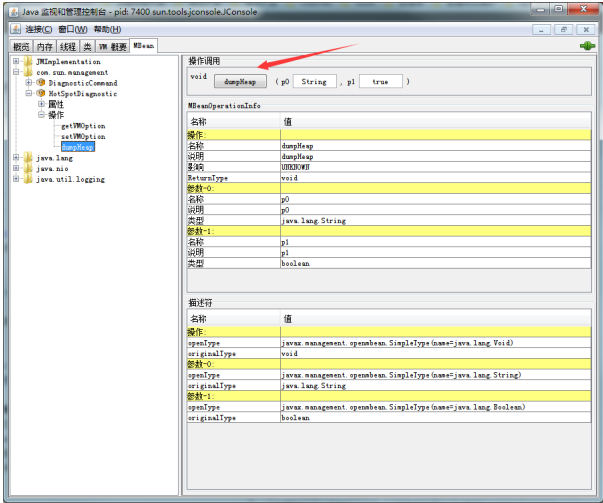
jmap 命令是JDK提供的用于生成堆内存信息的工具，可以执行下面的命令生成Heap Dump：

```
1 jmap -dump:live,format=b,file=heap-dump.bin <pid>
```

其中的pid是JVM进程的id， heap-dump.bin是生成的文件名称，在执行命令的目录下面。推荐此种方法。

【使用JConsole生成】

JConsole是JDK提供的一个基于GUI查看JVM系统信息的工具，既可以管理本地的JVM，也可以管理远程的JVM，可以通过下图的 dumpHeap 按钮生成 Heap Dump文件。



【使用jhat分析】

[jhat](#) 是JDK自带的用于分析JVM Heap Dump文件的工具，使用下面的命令可以将堆文件的分析结果以HTML网页的形式进行展示：

```
1 jhat <heap-dump-file>
```

其中 heap-dump-file 是文件的路径和文件名，可以使用 -J-Xmx512m 参数设置命令的内存大小。执行成功之后显示如下结果：

```
Snapshot resolved.
Started HTTP server on port 7000
Server is ready.
```

这个时候访问 <http://localhost:7000/> 就可以看到结果了。

Stack Trace

可使用jstack查看，定位线程出现长时间停顿的原因。