

## 第4章 虚拟机性能监控与故障处理

### 4.2 JDK命令行工具

JDK命令行工具大多数是jdk/lib/tool.jar类库的一层薄包装，主要功能代码实在tools类库实现的。

Sun JDK监控和故障处理工具如下表：

名称	主要作用
jps	JVM Process Status Tool,显示指定系统内所有的HotSpot虚拟机进程
jstat	JVM Statistics Monitoring Tool,用于手机HotSpot虚拟机各方面的运行数据
jinfo	Configuration Info for Java,显示虚拟机配置信息
jmap	Memory Map for Java,生成虚拟机的内存转储快照
jhat	JVM Heap Dump Browser,用于分析heapdump文件，它会建立一个HTTP/HTML服务器，让用户可以在浏览器上查看分析结果。
jstack	Stack Trace for Java，显示虚拟机的线程快照

#### 4.2.1 jps:虚拟机进程状况工具

功能：可以列出正在运行的虚拟机进程，并显示虚拟机执行主类（Main Class，main函数所在的类）名称以及这些进程的本地虚拟机唯一ID（LVMID）。

对于本地虚拟机进程来说，LVMID与操作系统的进程ID（Process Identifier,PID）是一致的，使用Windows的任务管理器或者UNIX的ps命令也可以查询到虚拟机进程的LVMID，但如果启动了多个虚拟机进程就麻烦了。

jps命令格式：

jps [option][hostid]

jps工具的主要选项：

选项	作用
-q	只输出LVMID，省略主类名称
-m	输出虚拟机进程启动时传递给主类main()函数的参数
-l	输出主类的全名，如果进程执行的是Jar包，输出Jar路径
-v	输出虚拟机进程启动时的JVM参数

#### 4.2.2 jstat:虚拟机统计信息监视工具

作用：用于监视虚拟机各种运行状态信息的命令行工具。显示本地或远程虚拟机进程中的类装载、内存、垃圾收集、JIT编译等运行数据，在没有GUI图形界面，只提供纯文本控制台环境的服务器上，它将是运行期定位虚拟机性能问题的首选工具。

jstat命令格式：

jstat [option vmid [interval[s|ms]][count]]

对于命令格式中的VMID与LVMID需要特别说明一下：如果是本地虚拟机进程，VMID与LVMID是一致的，如果是远程虚拟机进程，那VMID的格式为：

[protocol:][/]lvmid[@hostname[:port]/servername]

参数interval和count代表查询间隔和次数，如果省略这两个参数，说明值查询一次。

选项option代表着用户希望查询的虚拟机信息，主要分为3类：类装载、垃圾收集、运行期编译状况。

jstat工具的主要选项：

选项	作用
-class	监视类装载、卸载数量、总空间以及类装载所耗费的时间
-gc	监视Java堆状况，包括Eden区、两个Survivor、老年代、永久代等的容量、已用空间、GC时间合计等信息
-gccapacity	监视内容和-gc基本相同，但输出主要关注Java堆各个区域使用到的最大最小空间
-gcutil	监视内容和-gc基本相同，但输出主要关注已使用空间占总空间的百分比
-gccause	与-gcutil功能一样，但是会额外输出导致上一次GC产生的原因
-gcnew	监视新生代GC状况
-gcnewcapacity	监视内容与gcnew基本相同，输出主要关注使用到的最大、最小空间
-gcold	监视老年GC状况
-gcoldcapacity	监视内容与-gcold基本相同，输出主要关注使用到的最大、最小空间
-gcpermcapacity	输出永久代使用到的最大最小的空间
-compiler	输出JIT编译器编译过的方法、耗时等信息
-printcompilation	输出已经被JIT编译的方法

#### 4.2.3 jinfo:Java配置信息工具

jinfo的作用是实时地查看和调整虚拟机各项参数。使用jps命令的-v参数可以查看虚拟机启动时西安市指定的参数列表。但如果想知道未被显示指定的参数的系统默认值，可以使用jinfo的-flag选项查询。如果JDK1.6版本及其以上的话，使用java -XX: +PrintFlagsFinal查看系统默认值也可以。jinfo还可以使用-sysprops选项吧虚拟机进程的System.getProperties()的内容打印出来。可以使用-flag [+|-] name或者-flag name=value修改一部分运行期可写的虚拟机参数值。

jinfo命令格式：

jinfo [option] pid

#### 4.2.4 jmap:Java内存映射工具

jmap(Memory Map for Java)命令用于生成用于堆转储快照(一般称为heapdump或者dump文件)

jmap还可以查询finalize执行队列、Java堆和永久代的详细信息，如空间使用了、当前用的是哪种收集器。

在windows平台上jmap工具的很多功能是受限的，除了生成dump文件和用于查看每个类的实例、空间占用统计的-histo选项在所有操作系统提供外，其余操作只在Linux/Solaris下使用：

jmap命令格式：

jmap [option] vmid

jmap工具的主要功能选项：

选项	作用
-dump	生成Java堆转储快照。格式为：-dump:[live, ]format = b,file=,其中live子参数说明是否只dump出存活的对象。
-finalizerinfo	显示在F-Queue中等待Finalizer线程执行finalize方法的对象。只在Linux/Solaris平台下有效
-heap	显示Java堆详细信息，如使用哪种回收器、参数配置、分代情况等，只在Linux/Solaris平台下有效
-histo	显示堆中对象统计信息，包括类、实例数量、合计容量
-permstat	以ClassLoader为统计口径显示永久代内存状态。只在Linux/Solaris平台下有效
-F	当虚拟机进程对-dump选项没有响应时，可使用这个选项生成dump快照，只在Linux/Solaris平台下有效

#### 4.2.5 jhat:虚拟机堆转储快照分析工具

Sun JDK 提供了jhat命令与jmap搭配使用，来分析jmap生成的对转储快照。但是一般不适用jhat命令来分析dump文件，因为：

1. 一般不会对部署应用程序的服务器上直接分析dump文件，即使可以这样做，也会尽量将dump文件复制到其他机器上分析，因为分析工作是一个耗时而且小号硬件资源的过程。
2. jhat的分析功能相对比较简陋。

使用jhat分析，当命令行显示Server is ready.的提示后，用户在浏览器中键入http://localhost:7000/就可以看到分析结果。

#### 4.2.6 jstack:Java堆栈跟踪工具

jstack命令用于生成虚拟机当前时刻的线程快照(一般称为Threaddump或者javacore文件)。线程快照就是当前虚拟机内每一条线程正在执行的方法堆栈集合，生成线程快照的主要目的是定位线程出现长时间停顿的原因，如线程间死锁、死循环、请求外部资源导致的长时间等待等都是导致线程长时间停顿的常见原因。

jstack命令格式：

jstack [option] vmid

jstack工具主要选项:

选项	作用
-F	当正常输出的请求不被响应时，强制输出线程堆栈
-l	除堆栈外，显示关于锁的附加信息
-m	如果调用到本地方法的话，可以显示C/C++堆栈

#### 4.2.7 HSDIS: JIT生成代码反汇编

HSDIS是一个Sun官方推荐的HotSpot虚拟机JIT编译代码的反汇编插件，它的作用是让HotSpot的-XX:+PrintAssembly指令调用它来把动态生成的本地代码还原为汇编代码输出，同时生成大量有价值的注释。

```
java -XX:+PrintAssembly -Xcomp -XX:CompileCommand=dontinline,*Bar.sum -  
XX:CompileCommand=compileonly,*Bar.sum test.Bar
```

参数-Xcomp是让虚拟机以编译模式执行代码。两个-XX:CompileCommand意思是让编译器不要内联sum()并且编译sum(),-XX:+PrintAssembly就是输出反汇编内容。

### 4.3 JDK的可视化工具

#### 4.3.1 JConsole:Java监视和管理控制台

JConsole是一种基于JMX的可视化监视、管理工具。它管理部分的功能是针对JMX MBean进行管理。

##### 1.启动JConsole