虚拟机性能监控与故障处理工具

笔记本： jvm
创建时间： 2018/10/3/周三 12:13 **更新时间：** 2018/10/4/周四 8:55
作者： 1634896520@qq.com
URL： http://xianqi-h.iteye.com/blog/1346491

虚拟机性能监控与故障处理工具

一、JDK命令行工具

（一）**jps** :虚拟机进程状况工具

①列出正在运行的虚拟机进程，并显示虚拟机执行主类名称以及这些进程的本地虚拟机唯一ID。
②jps命令格式
jps [options] [hostid]
③jps工具主要选项

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

（二）**jstat** :虚拟机统计信息监视工具

①用于监视虚拟机各种运行状态信息的命令行工具。它可以显示本地或者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。
②主要命令

jstat -class pid	显示加载class的数量，及所占空间等信息。
jstat -compiler pid	:显示VM实时编译的数量等信息。
jstat -gc pid	:可以显示gc的信息，查看gc的次数，及时间。其中最后五项，分别是young gc的次数，young gc的时间，full gc的次数，full gc的时间，gc的总时间。
jstat -gccapacity	:可以显示，VM内存中三代（young,old,perm）对象的使用和占用大小，如：PGCMN显示的是最小perm的内存使用量，PGCMX显示的是perm的内存最大使用量，PGC是当前新生成的perm内存占用量，PC是但前perm内存占用量。其他的可以根据这个类推， OC是old内纯的占用量。
jstat -gcnew pid	:new对象的信息。
jstat -gcnewcapacity pid	:new对象的信息及其占用量
jstat -gcold pid	:old对象的信息。
jstat -gcoldcapacity pid	:old对象的信息及其占用量。
jstat -gcpermcapacity pid	: perm对象的信息及其占用量。
jstat -util pid	:统计gc信息统计。
jstat -printcompilation pid	:当前VM执行的信息。

③除了以上一个参数外，还可以同时加上 两个数字，如：jstat -printcompilation 3024 250 6是每250毫秒打印一次，一共打印6次，还可以加上-h3每三行显示一下标题。

（三）**jinfo** :Java配置信息工具

①实时的查看和调整虚拟机各项参数

②常用命令：

- jps的-v 参数可以查看虚拟机启动时显式指定的参数列表。
- jinfo的 - flag选项进行查询未被显式指定的参数的系统默认值。
- 使用 -flag name=value 或者 -flag [+|-] name 修改一部分运行期可写的虚拟机参数值

（四）**jmap** : Java内存映像工具

①用于生成堆转储快照

②还可以查询finalize执行队列、Java堆和永久代的详细信息，如空间使用率、当前用的是哪种收集器等。

③主要命令：

jmap -dump pid	生成Java堆转储快照
jmap -finalizerinfo pid	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
jmap -heap pid	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
jmap -histo pid	显示堆中对象统计信息，包括类、实例数量、合计容量
jmap -permstat pid	以ClassLoader为统计口径显示永久代内存状态，只在Linux、Solaris下有效
jmap -F pid	当虚拟机进程对-dump选项没有响应时，可使用这个选项强制生成dump快照，只在Linux/Solaris平台下有效

（五）**jhat**：虚拟机堆转储快照分析工具

- jhat与jmap配合使用，jhat内置了一个微型的HTTP/HTML服务器，生成dump文件的分析结果后，可以在浏览器中查看。
- 实例分析：

首先，通过jps 命令查到LVMID,然后执行：jmap -dump:format=b,file=eclipse.bin LVMID，返回值末尾为“Heap dump file create”

然后执行：jhat eclipse.bin，返回值末尾为：“Sever is ready”

然后在浏览器中输入：http://localhost:7000/ ,即可看到分析结果。

（六）**jstack** :Java堆栈跟踪工具

①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的集合，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，如线程死锁，死循环，请求外部资源导致的长时间等待等都是线程长时间停顿的常见原因。

②jstack工具主要选项

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

*：在jdk1.5中，java.lang.Thread类新增了一个getAllStackTraces()方法用于获取虚拟机中所有线程的StackTraceElement对象。使用这个方法可以通过简单的几行代码就完成jstack的大部分功能。

（七）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，它包含在HotSpot虚拟机的源代码之中，但没有提供编译后的程序。

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，同时还大量生成大量非常有价值的注释。

二、JDK可视化工具

JConsole、与VisualVM

（一）JConsole

①启动JConsole：通过JDK/bin目录下的“jconsole.exe”启动后，将自动搜索出本机运行的所有虚拟机进程，不需要用户自己再使用jps来查询了。

②“内存”页标签相当于可视化的jstat命令，用于监视受收集器管理的虚拟机内存（Java堆和永久代）的变化趋势。

③“线程”标签相当于可视化的jstack命令，遇到线程停顿时可以使用这个标签进行监控分析。

（三）VisualVM ： 多合一故障处理工具

①VisualVM 兼容范围与插件安装

VisualVM基于NetBeans平台开发，因此它一开始就具备了插件扩展功能的特性，通过插件支持，VisualVM可以做到：

- 显示虚拟机进程以及进程的配置、环境信息。
- 监视应用程序的CPU、GC、堆、方法区以及线程的信息。
- dump以及分析堆转储快照。
- 方法级的程序性能分析，找出被调用最多、运行时间最长的方法。
- 离线程序快照：收集程序的运行时配置、线程dump、内存dump等信息建立一个快照，可以将快照发送开发者处进行Bug反馈。
- 其他plugins的无限可能、、、

②使用教程：https://jingyan.baidu.com/article/e9fb46e172e3747521f76611.html

③BTrace动态日志跟踪

在开发中我们经常会遇到程序出现问题，但排查的话就不得不停掉服务，通过调试增量来加入日志代码以解决问题，当遇到生产环境服务无法停止的时候，缺一两句日志导致排错进行不下去，是非常麻烦的一件事情。

所以就有了BTrace插件，这个插件的作用在于，可以在不停止目标程序运行的前提下，通过HotSpot虚拟机的HotSwap技术动态加入原本并不存在的调试代码。