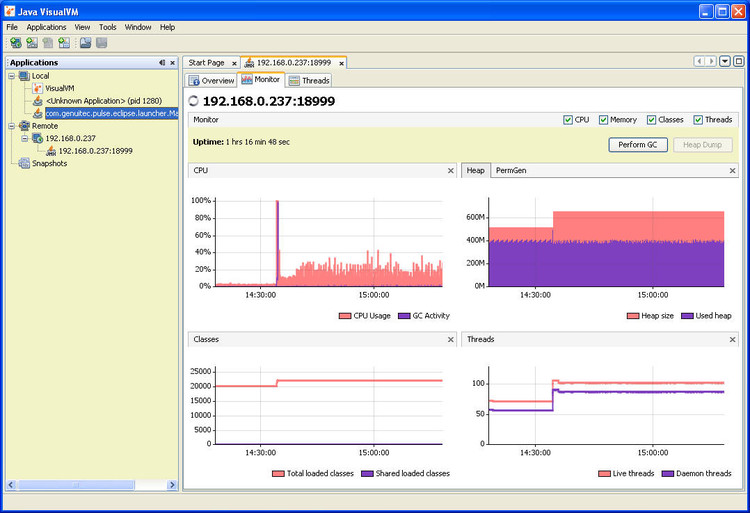
# 使用Java VisualVM监控远程JVM

我们经常需要对我们的开发的软件做各种测试, 软件对系统资源的使用情况更是不可少, 目前有多个监控工具, 相比JProfiler对系统资源尤其是内存的消耗是非常庞大,JDK1.6开始自带的VisualVM就是不错的监控工具.

这个工具就在JAVA\_HOME\bin\目录下的jvisualvm.exe, 双击这个文件就能看到一个比较直观的界面



从左边Applications树中可以知道，不光可以监控本地JVM运行情况， 还可以监控远程机器上的JVM运行情况.

本地监控：只要打开某个JAVA程序就会自动的加入到本地监控中.

因为本地监控无需配置, 所以这里主要介绍监控远程JVM

要进行远程监控, 本机的VisualVM就必须和远程的JVM要进行通信, Visualvm目前支持两种remote connection方式.

分别是jstatd和JMX方式: 这里我主要介绍的是通过JMX方式.

通过JMX连接远程机器, 需要经过下面的配置:

1. 修改远程机器JDK配置文件 (我这里远程机器是linux).

a.进入JAVA\_HOME\jre\lib\management\目录

b.拷贝jmxremote.password.template这个文件到当前目录, 并改名为jmxremote.password

c.打开jmxremote.password文件，去掉 # monitorRole QED 和 # controlRole R&D 这两行前面的注释符号

2. 修改远程机器上需要被监控的程序的配置文件 (我这里是监控Tomcat容器内部署的应用).

a.进入TOMCAT\_HOME\bin目录

b.打开catalina.sh文件，加入如下信息：

JAVA\_OPTS="$JAVA\_OPTS -Djava.rmi.server.hostname=192.168.0.237

-Dcom.sun.management.jmxremote.port=18999

-Dcom.sun.management.jmxremote.ssl=false

-Dcom.sun.management.jmxremote.authenticate=false"

c.重启Tomcat服务.

3. 客户端VisualVM配置 (我客户端用的是WinXP).

a.直接反键点击Remote，选择Add Remote Host...

b.在弹出的界面中输入远程机器的IP地址(192.168.0.237)，这个IP地址会加入到Remote节点下.

c.反键点击这个IP地址，选择Add JMX Connection, 在弹出的界面中输入刚配置的端口号(18999), 这个连接会加入到该IP节点下.

d.反键点击这个连接，选择Open.

此时就可以看到监控的界面, 从界面上我们可以看到CPU信息， 内存信息， 统计加载类数量，线程信息.

# [LoadRunner性能测试入门教程](http://www.51zxw.net/list.aspx?cid=497)

http://www.51zxw.net/list.aspx?cid=497