

# VisualVM 入门指南

2009-12-9

修订历史:

时间	修订内容	里程碑版本号	修订人
2009-12-9	初稿	1.0.0	XXX

# 目 录

---

1	什么是 Visual VM? .....	4
1.1	Visual VM 简介 .....	4
1.2	如何获取 VisualVM.....	4
2	Visual VM 实战 .....	5
2.1	开启 Visual VM 之旅.....	5
2.1.1	启动问题.....	5
2.1.2	界面简介.....	6
2.1.3	安装插件.....	6
2.1.4	监控本地 Java 应用.....	8
2.1.5	监控远程 Java 应用.....	8
2.2	使用 Visual VM 查看 JVM 相关信息.....	11
2.3	使用 Visual VM 解决内存溢出问题.....	14
2.4	使用 Visual VM 查看 Tomcat 的线程状态 .....	16
2.5	使用 Visual VM 查看 CPU 消耗情况 .....	20

# 1 什么是 Visual VM?

---

## 1.1 Visual VM 简介

VisualVM 提供在 Java 虚拟机 (Java Virtual Machine, JVM) 上运行的 Java 应用程序的详细信息。在 VisualVM 的图形用户界面中, 您可以方便、快捷地查看多个 Java 应用程序的相关信息。(摘自官方)

简单说来, VisualVM 是一种集成了多个 JDK 命令行工具的可视化工具, 它能为您提供强大的分析能力。所有这些都是免费的! 它囊括的命令行工具包括 jstat, JConsole, jstack, jmap 和 jinfo, 这些工具与 JDK 的标准版本是一致的。

可以使用 VisualVM 生成和分析海量数据、跟踪内存泄漏、监控垃圾回收器、执行内存和 CPU 分析, 同时它还支持在 MBeans 上进行浏览和操作。尽管 VisualVM 自身要在 JDK6 这个版本上运行, 但是 JDK1.4 以上版本的程序它都能监控。

对于我们来说, 这些功能都是至关重要的。以前大家可能会用 YourKit 这样的付费软件来检测我的应用软件, 但现在可以用 VisualVM 这样一个开源工具来实现同样的功能了。更何况, 有谁比能 Sun 公司更了解 JVM 呢?

## 1.2 如何获取 VisualVM

VisualVM 的一个最大好处就是, 它已经在你的 JDK bin 目录里了, 只要你使用的是 JDK1.6 Update7 之后的版本。点击一下 jvisualvm.exe 图标它就可以运行了。

这里是 VisualVM 的官方网站: <https://visualvm.dev.java.net>, 资料很全, 同时提供 VisualVM 最近版本下载。

# 2 Visual VM 实战

---

## 2.1 开启 Visual VM 之旅

如果你使用的是 JDK 是 1.6Update7 之后的版本，那么 Visual VM 已经包含在 bin 目录下了，否则需要去官方下载，目前最新版本是 1.2.1。

### 2.1.1 启动问题

如果你在 windows 上使用 Visual VM，需要做的只是点一下 jvisualvm.exe，就能启动它；绿色，好用。但是 Visual VM 所在的分区如果是 NTFS 格式，那么第一个问题就出现了：sun 对 NTFS 格式的硬盘支持有问题！但可通过参数可避免，并完成启动。步骤如下：

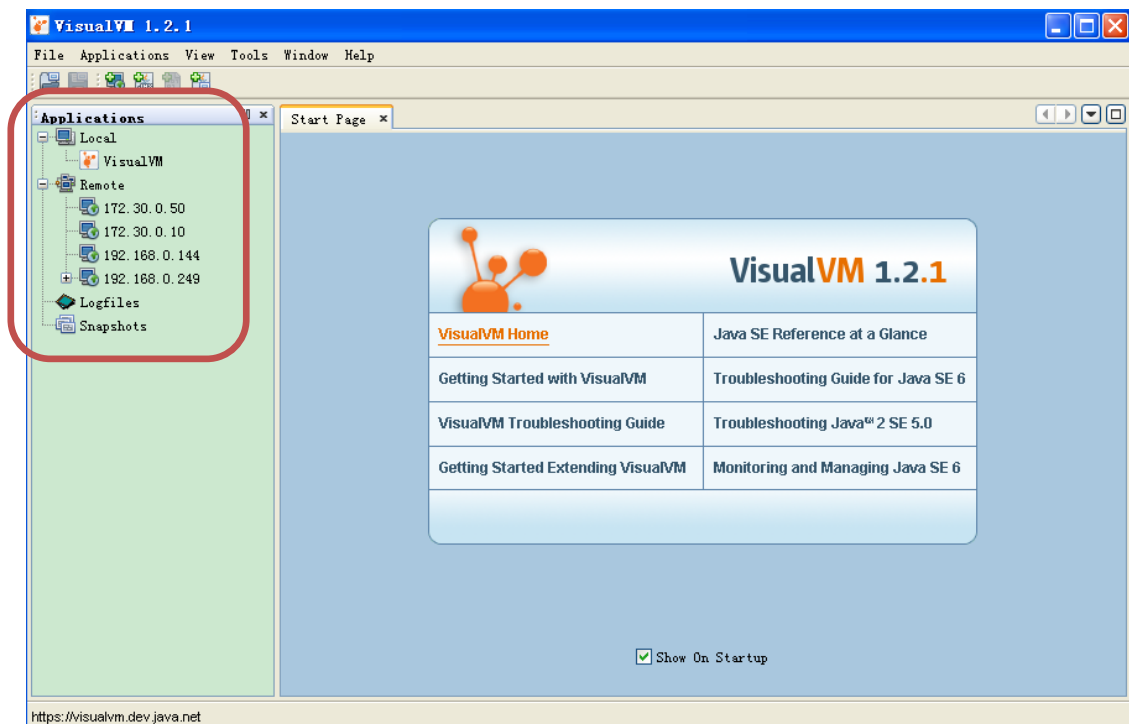
1. 创建一个 visualvm.exe 的快捷方式
2. 在“目标”中添加如下参数

```
-XX:+PerfBypassFileSystemCheck
```



## 2.1.2 界面简介

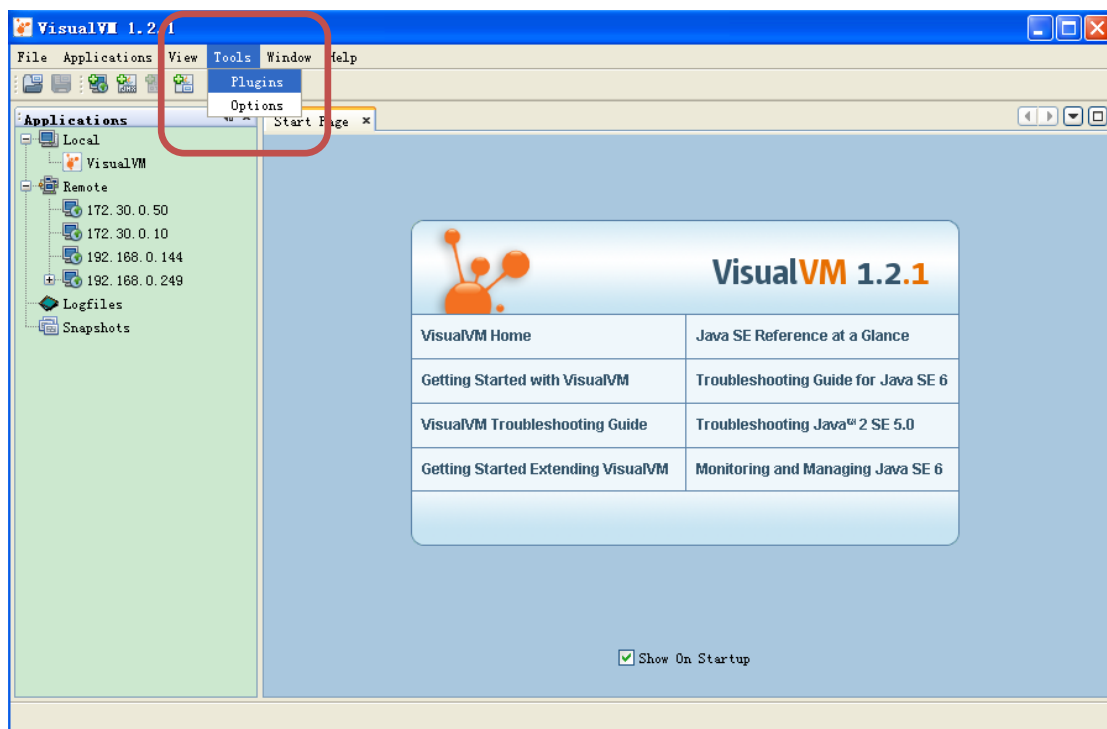
Visual VM 启动成功！可以看到 Visual VM 的界面了。通过 Visual VM 可以看到本机运行中的所有 Java 应用。你会发现根本不需要在 VisualVM 里为 Java 应用程序注册，它们就会自动显示出来。甚至还可以在导航栏里查看到远程的 Java 应用。导航栏即为 Applications，其中分为 Local（本地 Java 应用）和 Remote（远程的 Java 应用）。



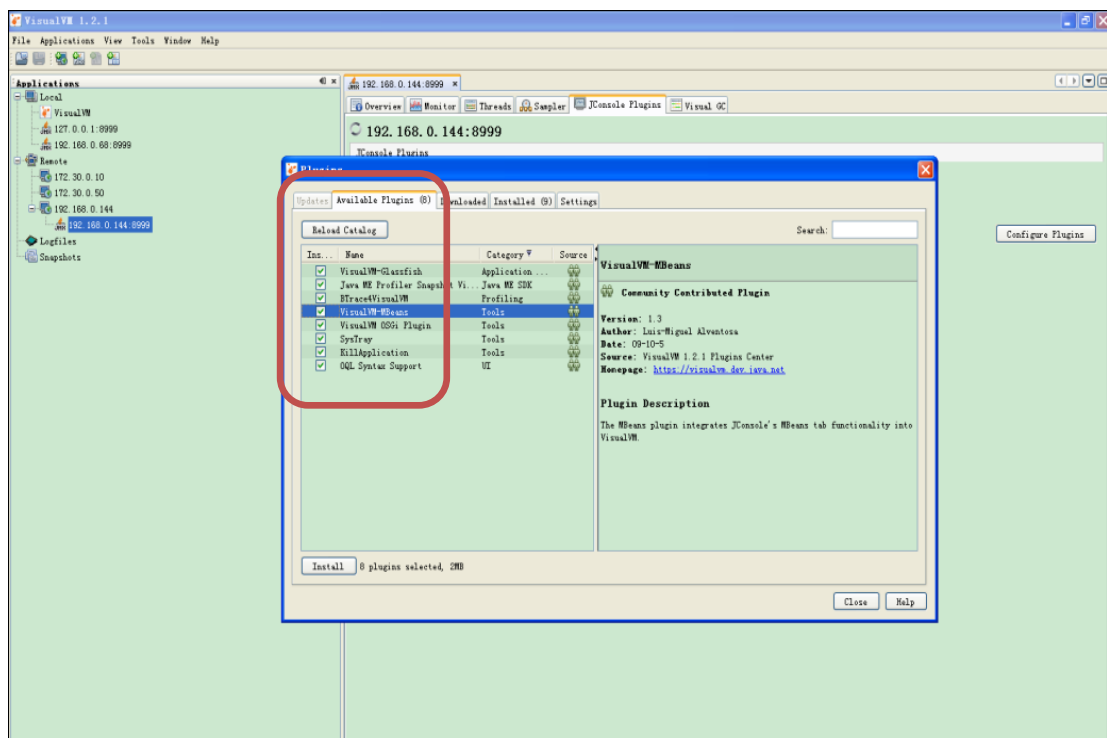
## 2.1.3 安装插件

Visual VM 有很多好用的插件；而且 Sun 为了推广 Visual VM，还举办的 Visual VM 插件开发大赛，一等奖奖金为 2000 刀，非常诱人。如何安装插件呢？步骤如下：

1. 点击 Tools -> Plugins



## 2. 推荐安装全部插件



## 2.1.4 监控本地 Java 应用

Visual VM 本身就是一个 Java 应用，所以打开 Visual VM 看到的第一个可监控应用就是 Visual VM 本身；可以用它热热身，小试下牛刀。在 Visual VM 可视化界面中可以监控到 Visual VM 本身的内存使用情况、线程情况、JVM 启动参数、cpu 消耗情况、垃圾回收情况等很多参数。当然如果在本地启一个 Tomcat 一样可以看到这些参数，可以方便我们在本地对 JVM 进行调优。但是且接如果你是在 windows 下起应用，如果你的 Java 应用是在 NTFS 格式的盘附上，记得加参数：

```
-XX:+PerfBypassFileSystemCheck
```

## 2.1.5 监控远程 Java 应用

监视远程 Java 应用稍微复杂些。Visual VM 是通过 JMX 来和远程 Java 应用联系的。

JMX: Java Management Extensions, 即 Java 管理扩展, 是一个为应用程序、设备、系统等植入管理功能的框架。JMX 可以跨越一系列异构操作系统平台、系统体系结构和网络传输协议, 灵活的开发无缝集成的系统、网络和服务管理应用。

下面以 192.168.0.249 上的 tomcat 为例, 叙述下监视远程 Java 应用的步骤如下:

1. 为 249 上的 tomcat 配置 jvm 启动参数。在 tomcat 的 catalina.sh 中添加如下参数:

```
JAVA_OPTS=' -Dcom.sun.management.jmxremote.port=8999  
-Dcom.sun.management.jmxremote.ssl=false  
-Dcom.sun.management.jmxremote.authenticate=false'
```

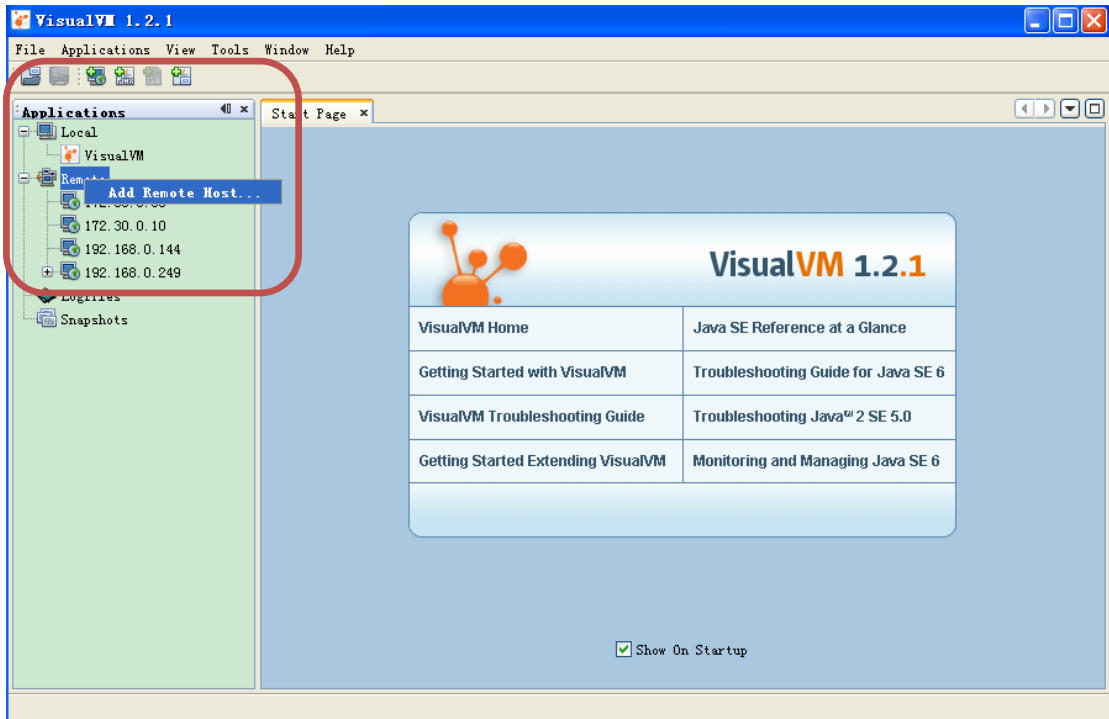
其中 `-Dcom.sun.management.jmxremote.port=8999` 指定了 JMX 启动的代理端口; 这个端口就是 Visual VM 要连接的端口

其中 `-Dcom.sun.management.jmxremote.ssl=false` 指定了 JMX 是否启用 ssl

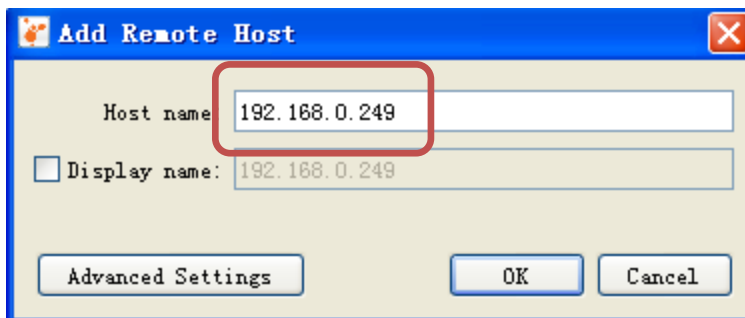


其中 `-Dcom.sun.management.jmxremote.port=8999` 指定了 JMX 是否启用鉴权（需要用户名，密码鉴权）

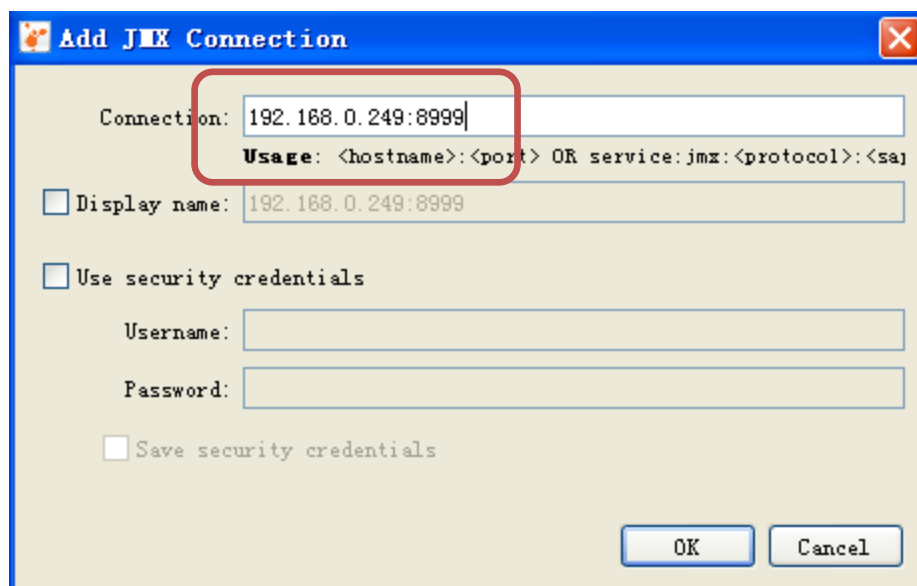
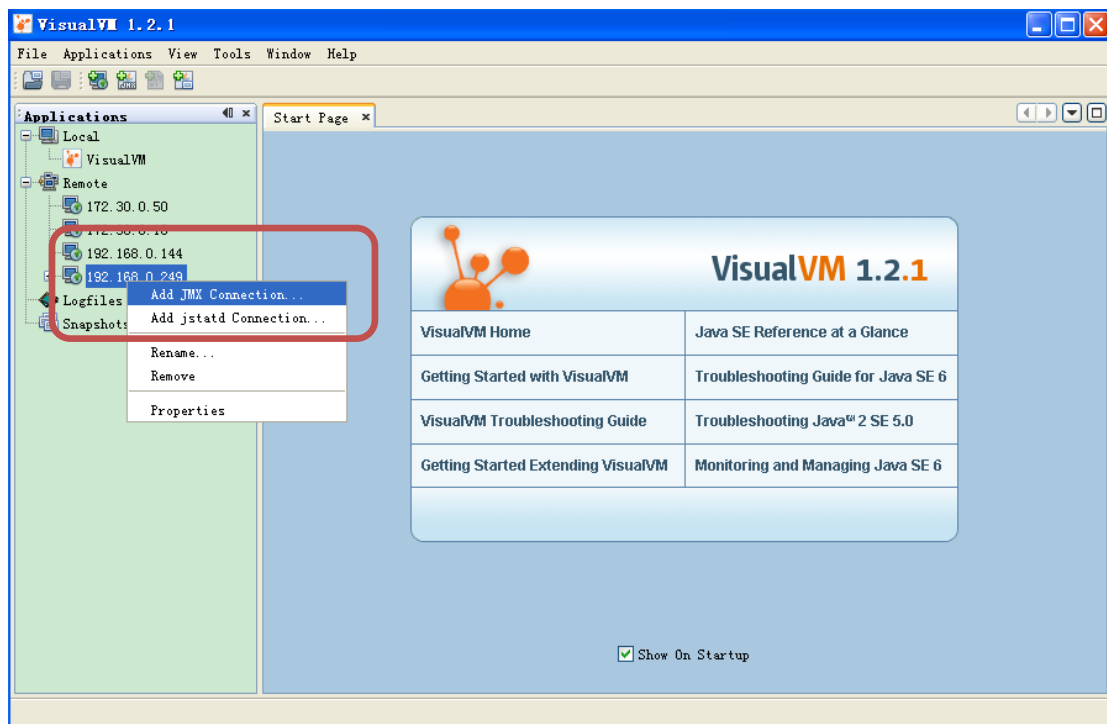
2. 在 Visual VM 添加远程连接的 Host。邮件点 Remote 后点击 Add Remote Host。



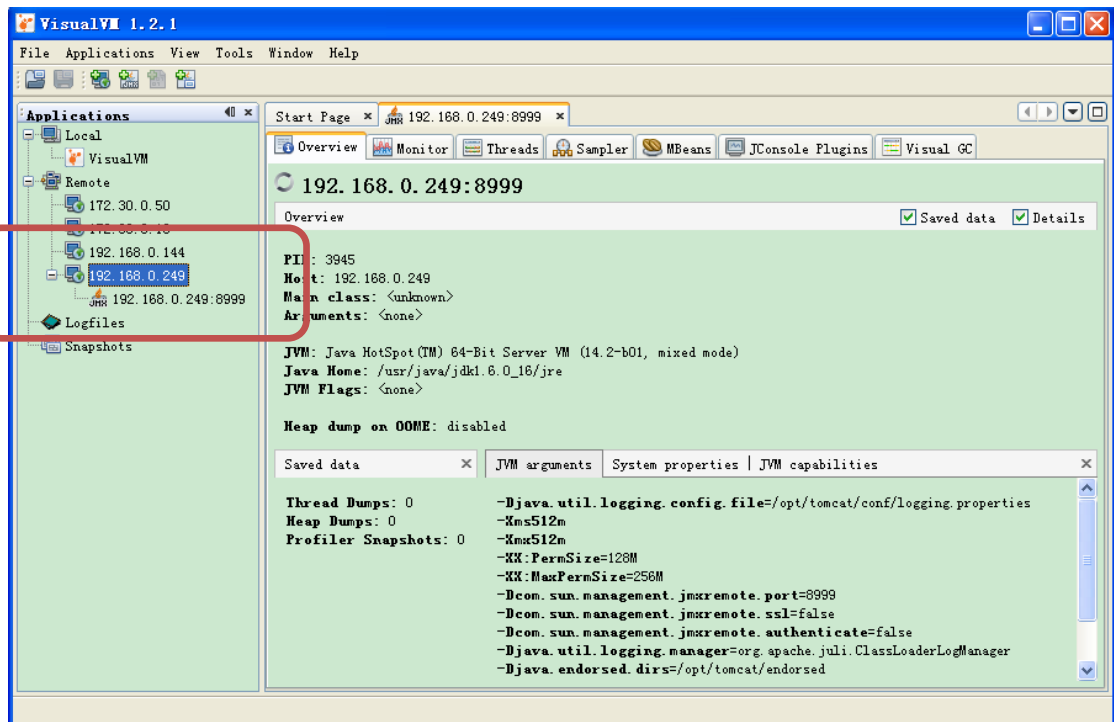
3. 添加 249 为需要监控的 Host



4. 使用 JMX 连接 249 的 8999 端口



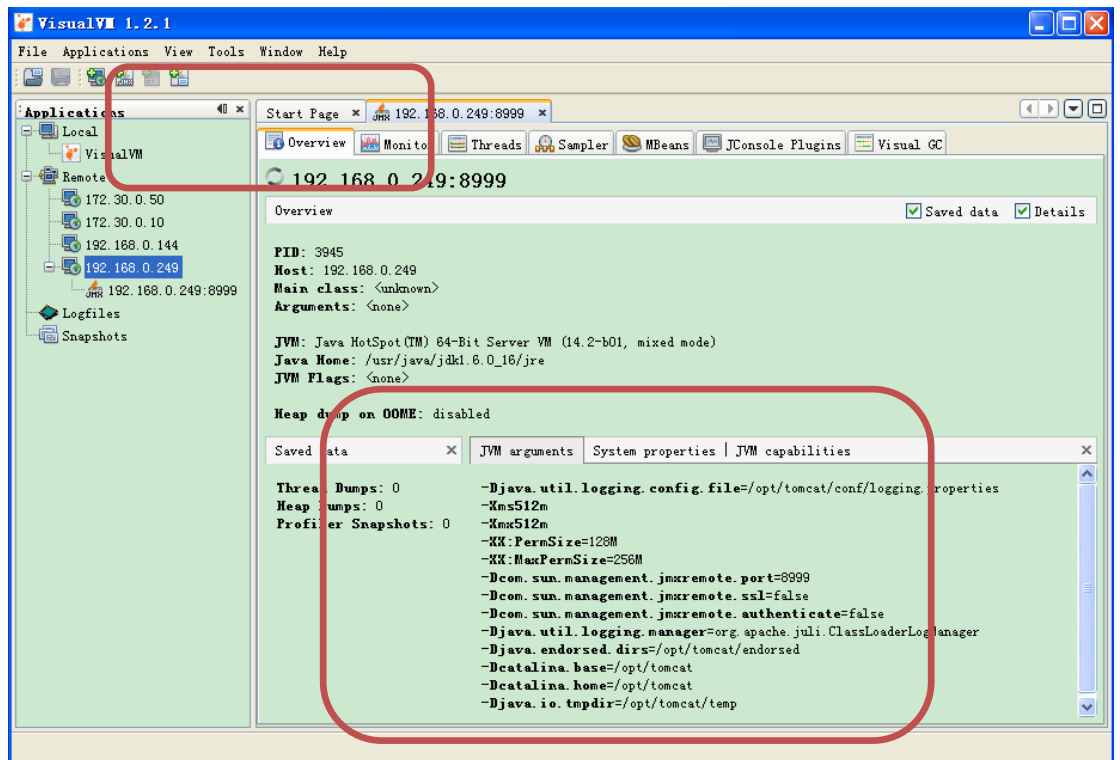
## 5. 连接 JMX 开始监控远程 Java 程序



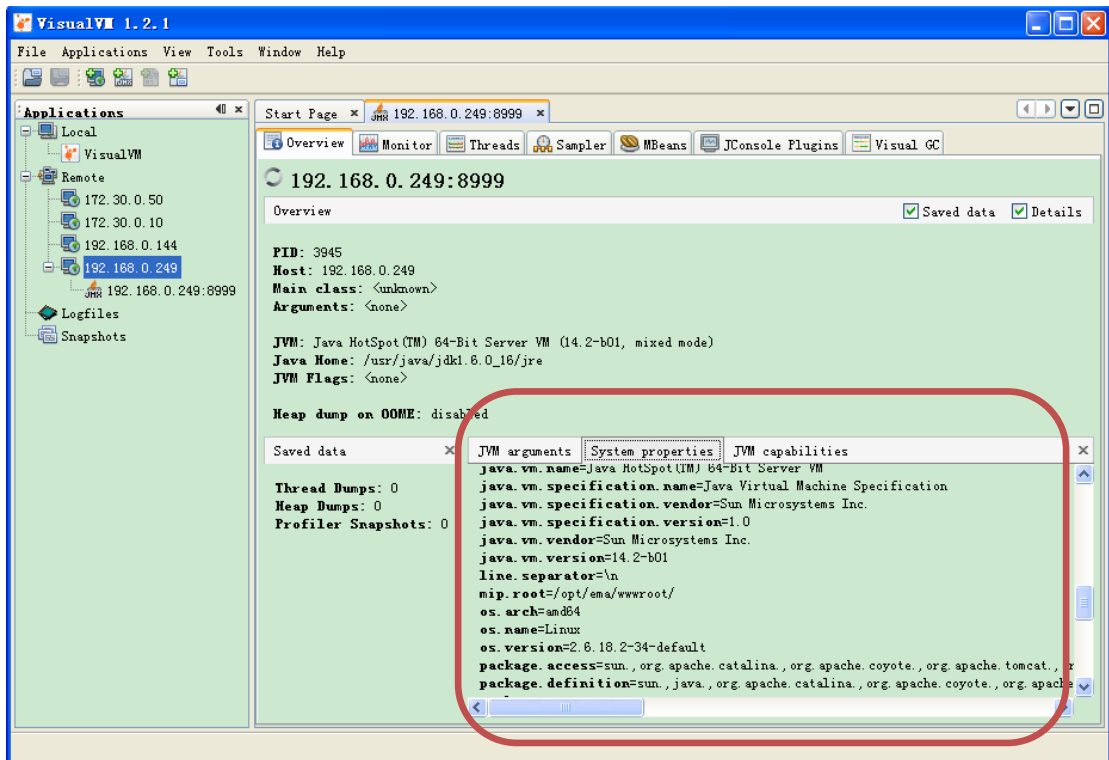
## 2.2 使用 Visual VM 查看 JVM 相关信息

Visual VM 可以将 JVM 的基本信息显示出来。基础信息包括：JVM 的启动参数、JRE 取得的系统参数（系统参数是可以在 Java 中使用 System 拿到的）、JVM 的能力参数。他们坐落在 Overview 试图内。

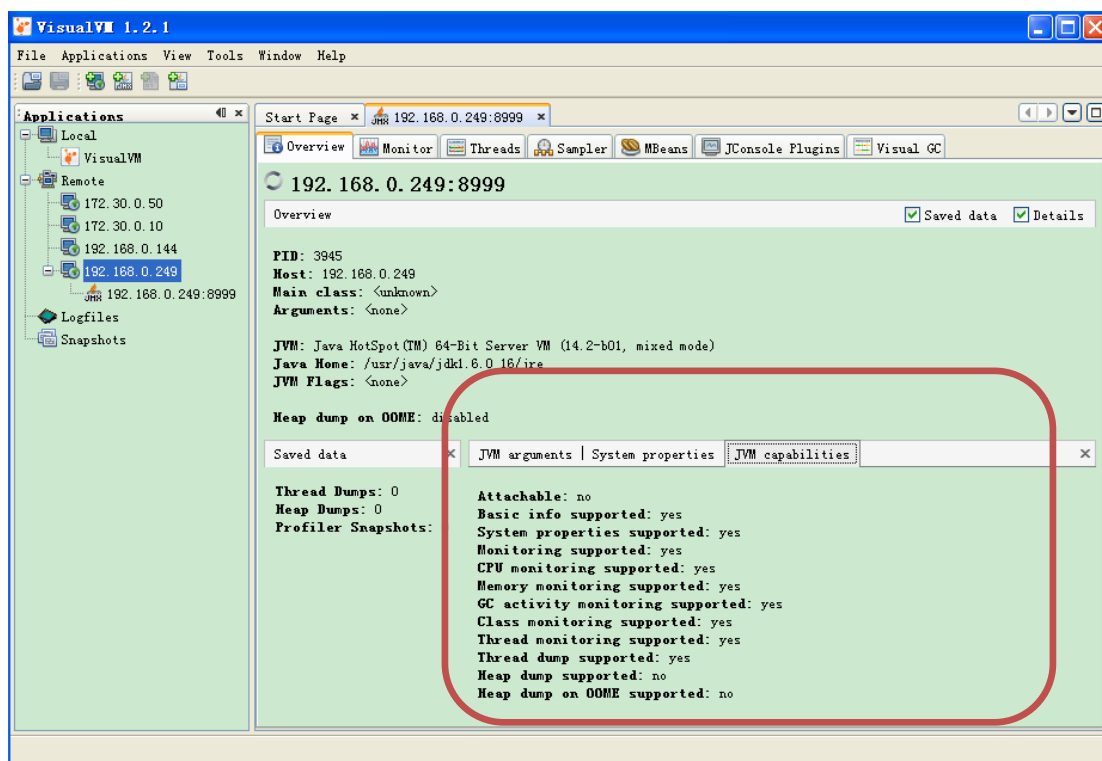
## 1. JVM 的启动参数



## 2. JRE 取得的系统参数

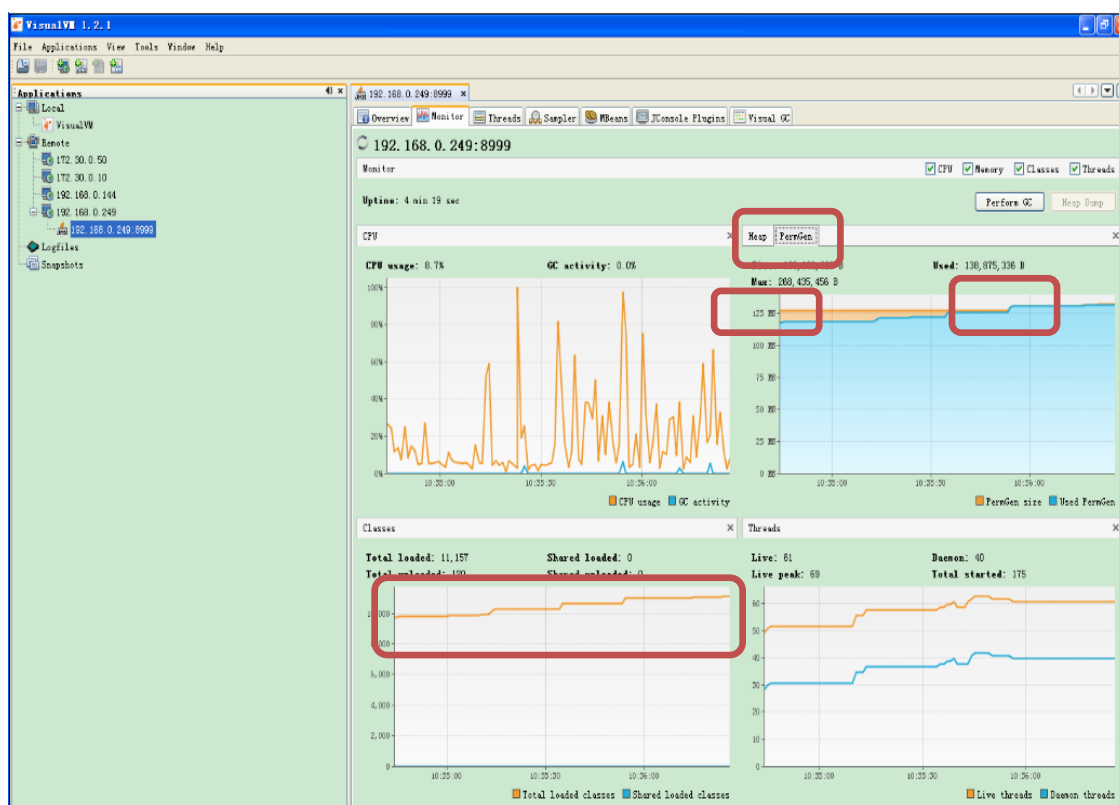


### 3. JVM 的能力参数

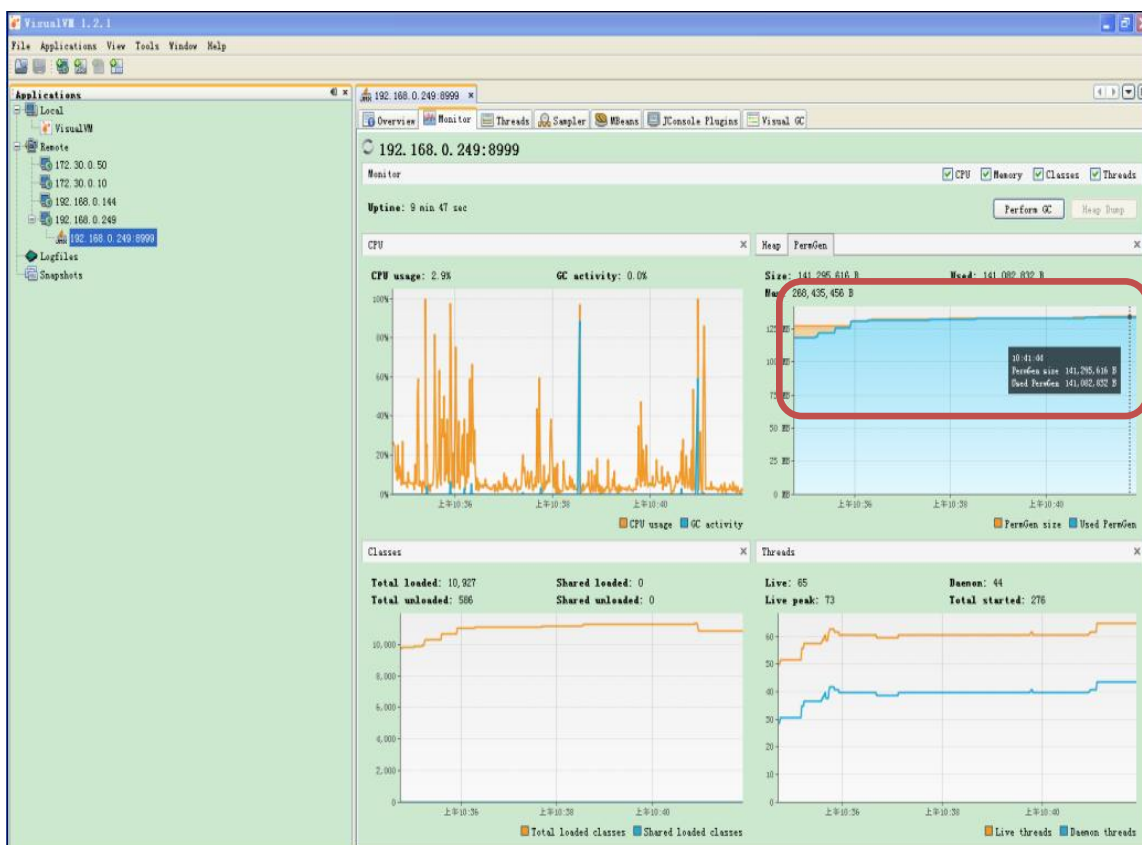


## 2.3 使用 Visual VM 解决内存溢出问题

最近公司的 Java 应用出现了 Perm 溢出的问题，我们可以用 Visual VM 来诊断下。将程序部署到 249 上之后通过 JMX 远程访问；之后观察 Monitor 视图



如上图所示，PermGen 在程序运行一段时间后超出了我们指定的 128MB，通过 Classes 视图看到，Java 在运行的同时加载了大量的类到内存中。PermGen 会存储 Jar 或者 Class 的描述信息；所以在 class 大量增加的同时 PermGen 超出了我们指定的范围。为了让应用稳定，我们需要探寻新的 PermGen 范围。检测时段时候后（如下图）发现 PermGen 在 145MB 左右稳定，那么我们就得到了稳定的新参数；这样 PermGen 内存溢出的问题得到解决。

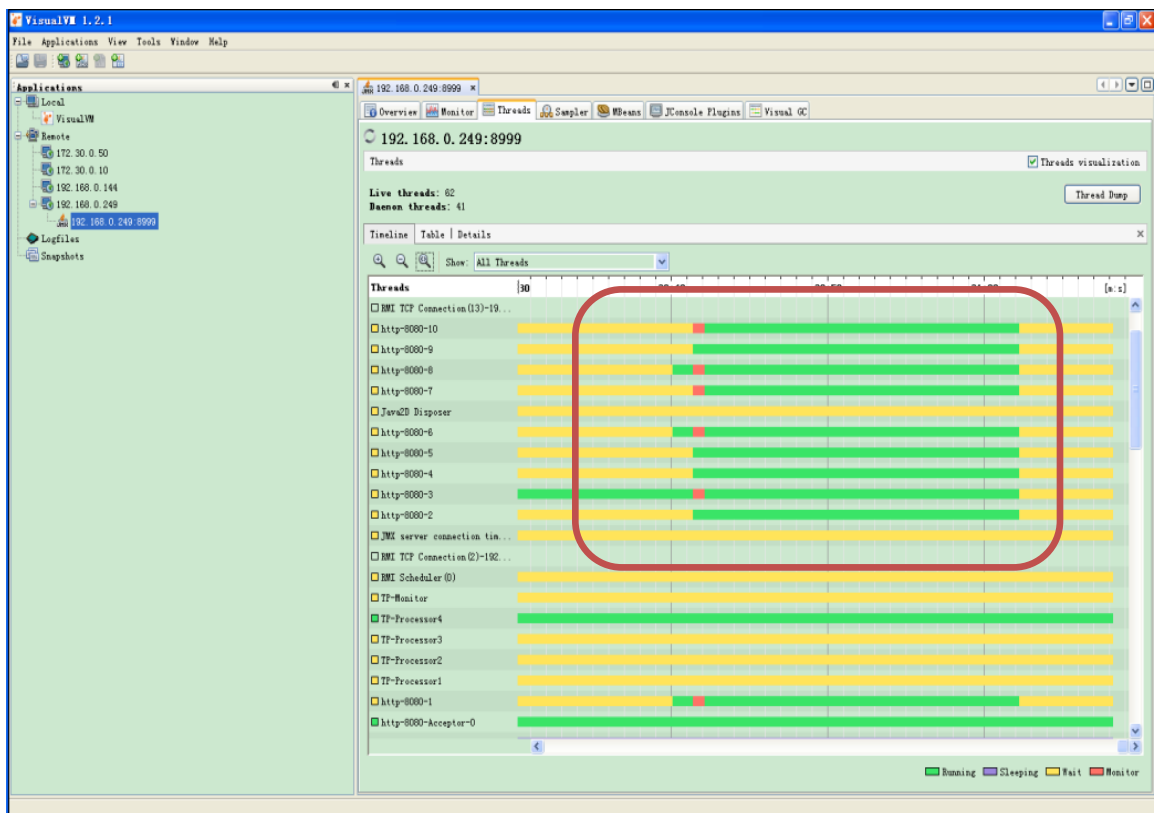


## 2.4 使用 Visual VM 查看 Tomcat 的线程状态

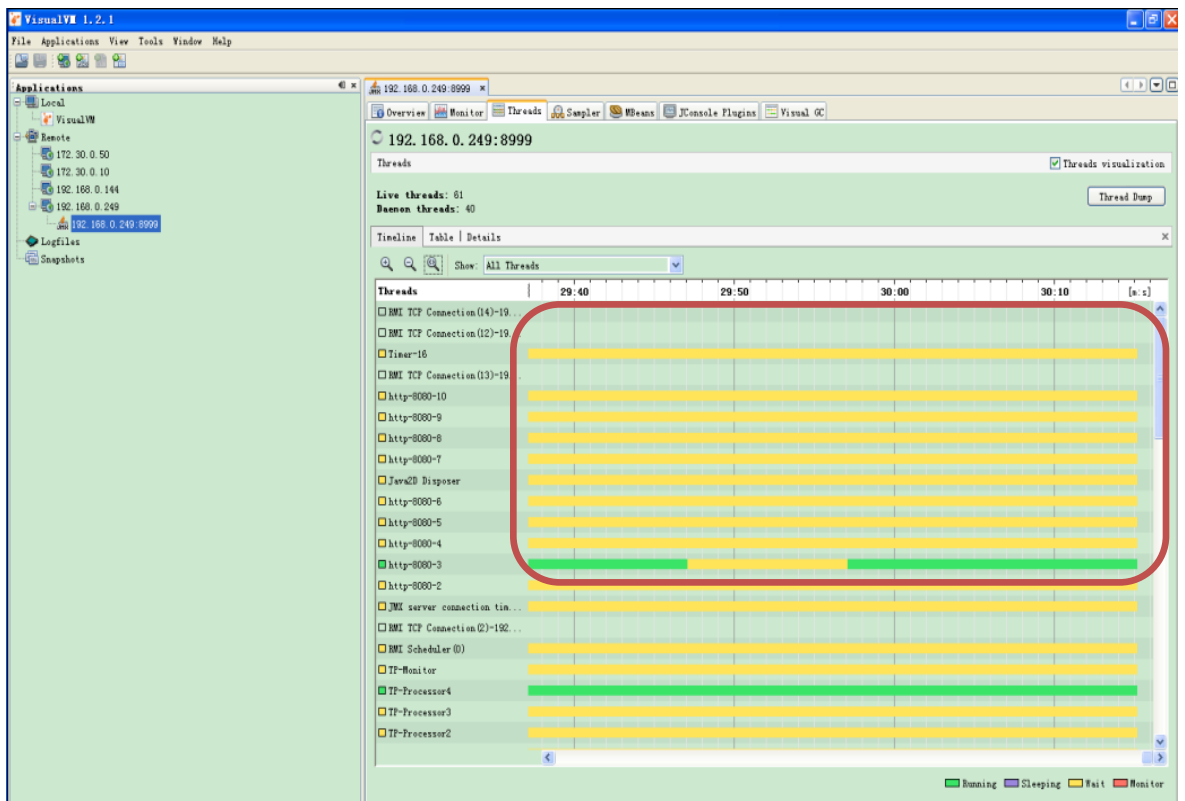
通过 Visual VM 可以检测到线程情况,这个东西在压力测试的时候特别好用。如果线程发生问题可以马上看出来,下面我们以一个正常运行的 tomcat 为例介绍下如何使用 Visual VM 观测 Thread



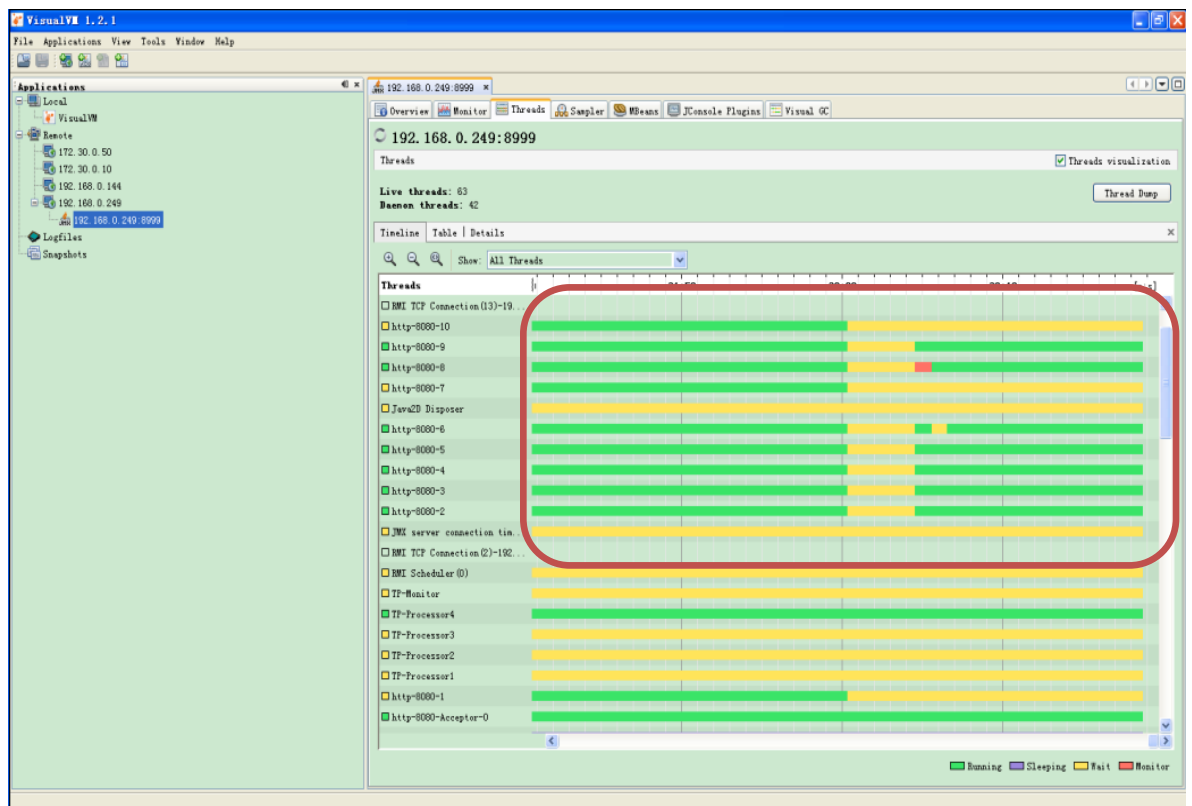
## 1. 一个用户访问 tomcat 线程



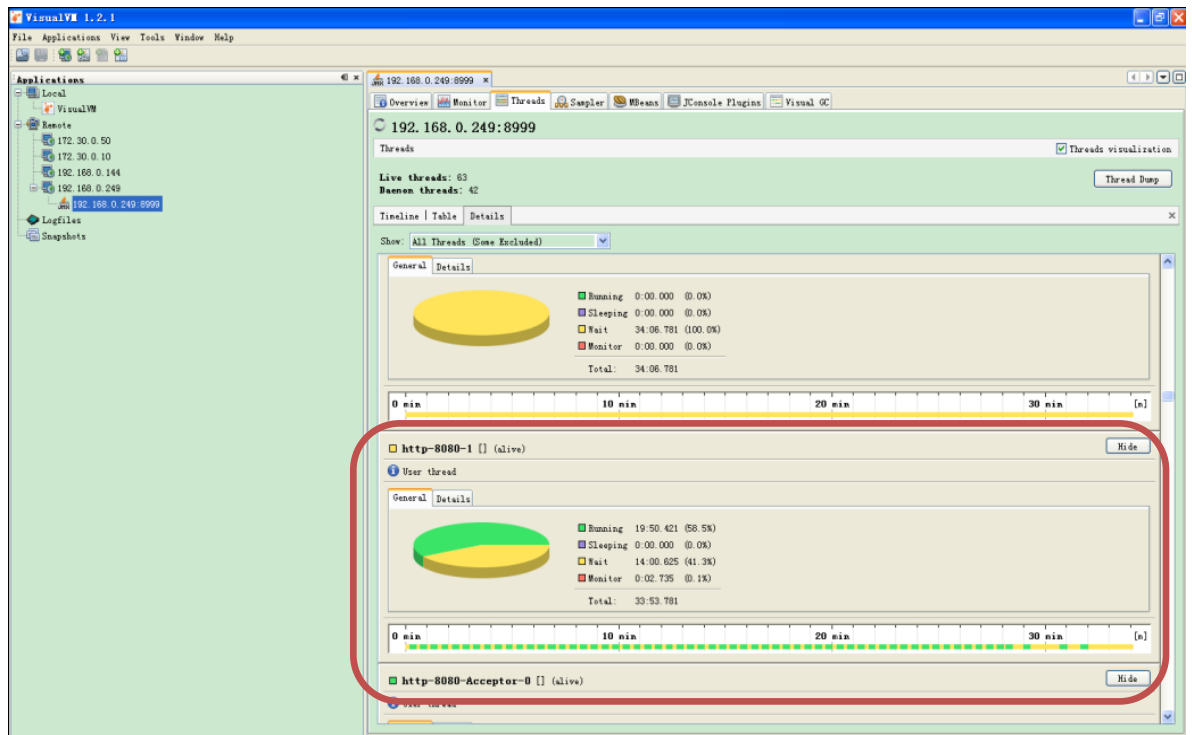
## 2. 用戶沉默了



### 3. 用户经常访问 tomcat



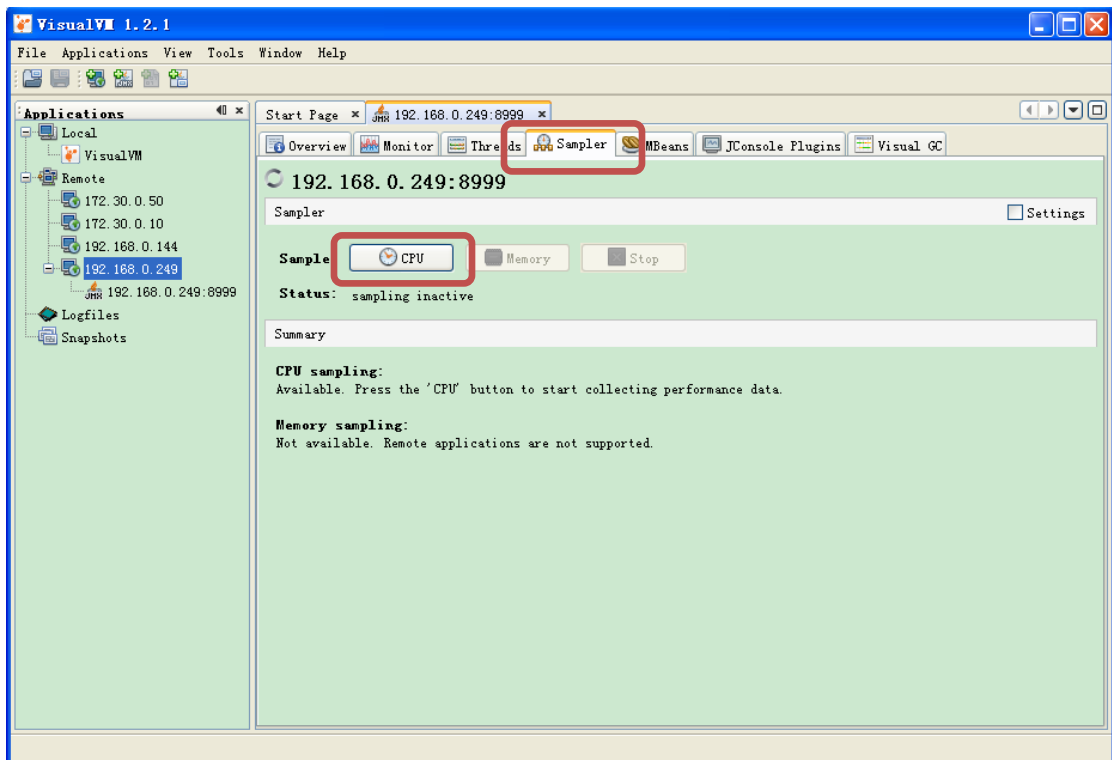
## 4. 访问情况统计



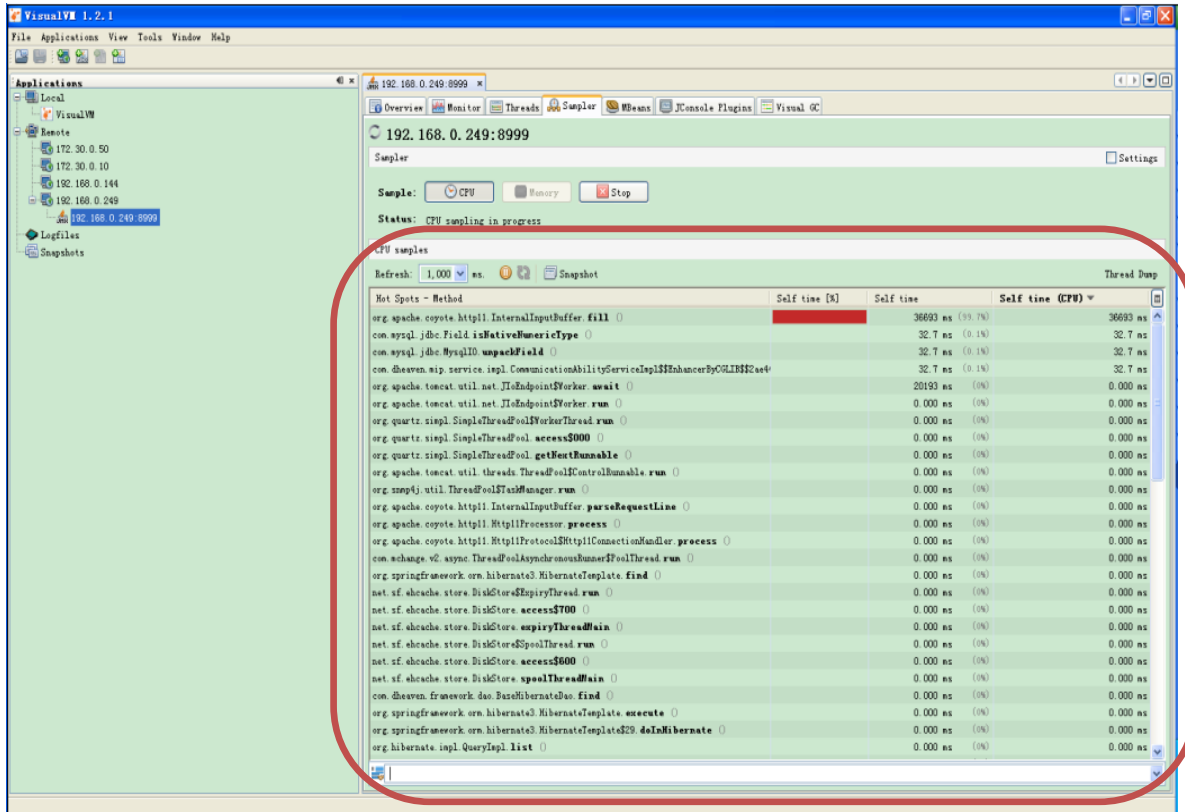
## 2.5 使用 Visual VM 查看 CPU 消耗情况

Visual VM 提供 CPU 监视功能，可以看到那些类消耗了 CPU。当应用耗费 CPU 过高时，可以通过此功能对应用进行追踪并调查原因。可以通过如下方式查看：

## 1. 如何进入 CPU 视图



## 2. 查看 CPU 视图



### 3. 使用搜索功能，查看 dheaven 相关的类

