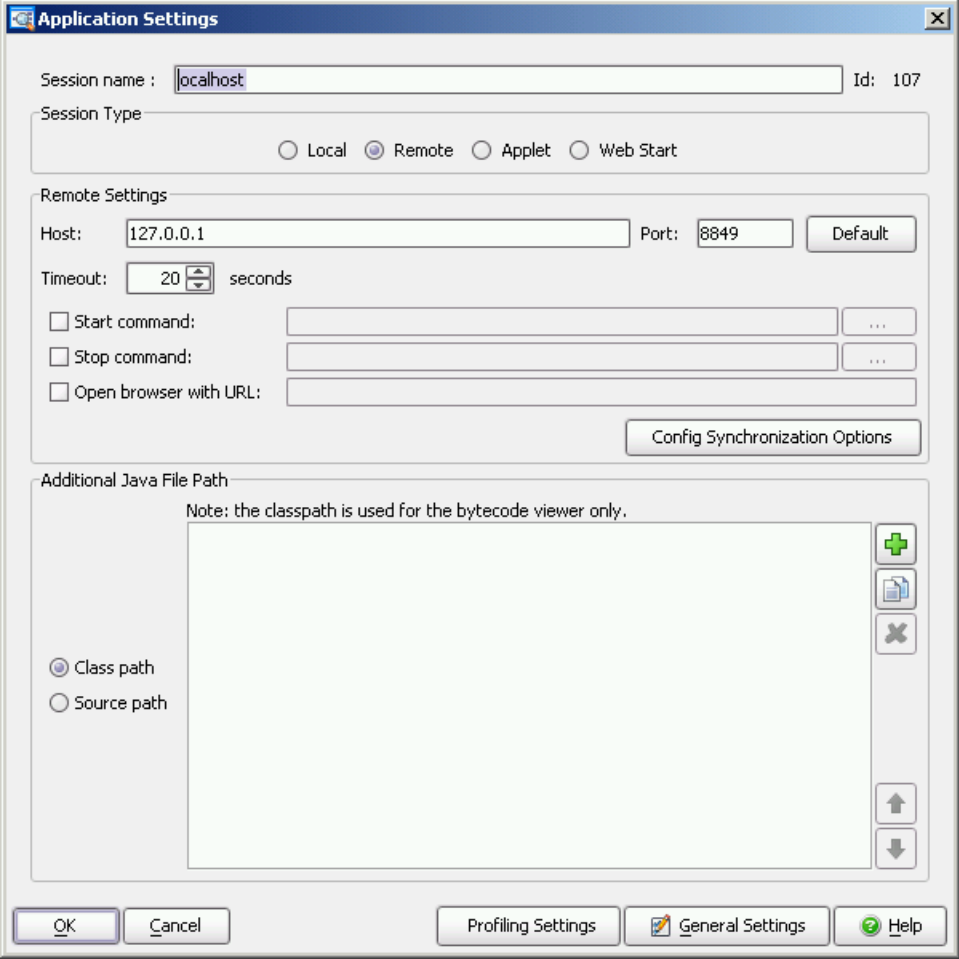


JProfiler

这个是一个服务器性能跟踪工具，可以分析服务器运行缓慢、包含内存泄漏等。通常不应该在并发测试中使用，因为开启此功能对服务器性能影响非常大。服务器开启JProfiler（通常建议在Tomcat上测试，因为配置会方便一些）：


- 1. 服务器上需要安装JProfiler（JProfiler服务器拷贝即可使用）
- 2. 修改服务器启动文件：
Tomcat: startup.sh/cmd
WebLogic: startWebLogic.sh/cmd
WebSphere: 进入控制台，应用程序服务器 > server1 - Java 和进程管理 - 进程定义 - Java 虚拟机
JBoss: run.sh/cmd
- 3. 在这些文件中（WebSphere例外）增加设置环境变量：
Windows: SET PATH=D:\Program Files\jprofiler4\bin\windows;%PATH%
Unix: export LD_LIBRARY_PATH=/usr/jprofiler4/bin/linux:\$LD_LIBRARY_PATH

- 在这些文件中（WebSphere在控制台中）找到Java参数设置的参数名称，通常为JAVA_OPTS、JAVA_OPTIONS等，添加以下设置：
Windows: SET JAVA_OPTS=%JAVA_OPTS% -agentlib:jprofilerti=port=8849 "-Xbootclasspath/a:D:\Program Files\jprofiler4\bin\agent.jar"
Unix: JAVA_OPTS=\${JAVA_OPTS} -agentlib:jprofilerti=port=8849 "-Xbootclasspath/a: /usr/jprofiler4/bin/agent.jar"
上述的路径请根据实际环境修改
- 4. 在启动服务器后，运行netstat -an|find "8849"或者netstat -an|grep 8849时应该可以看到该端口处于LISTENING状态。这时服务器是处于等待连接状态，还没有正式开始启动
 - 5. 在客户端启动JProfiler（注意必须与服务器使用的JProfiler为同一个版本，否则无法正常连接），点击菜单Session-New Session
 - 6. Session Type填写Remote（即使服务器是本地也是填写Remote），Host填写服务器地址，Port填写上面配置的端口号，然后点击确定

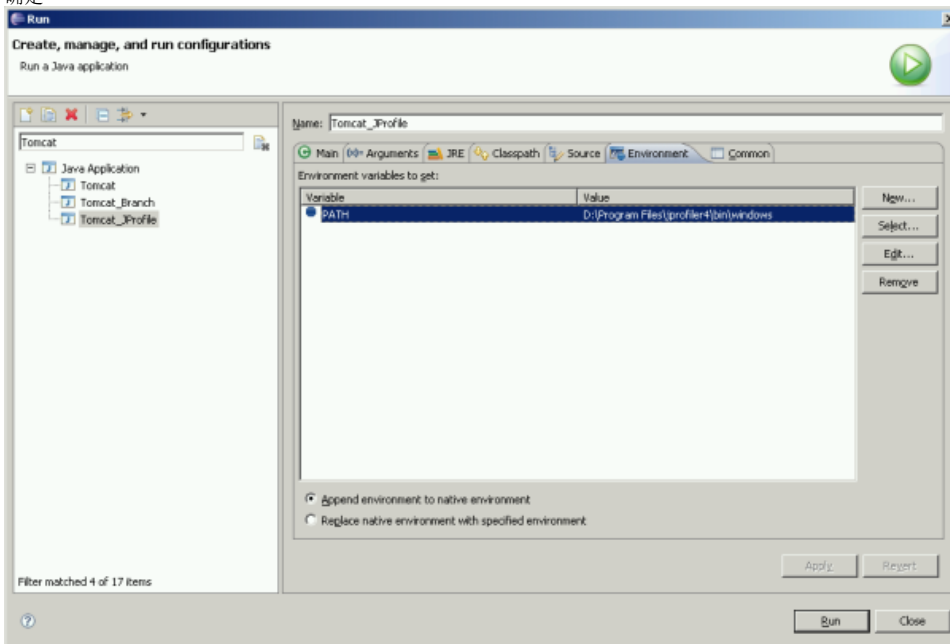


- 7. 在Profiling settings template中选择一个合适的模块（通常选择第一个即可），点击确定连接到服务器
- 8. 服务器此时才开始正式的启动工作

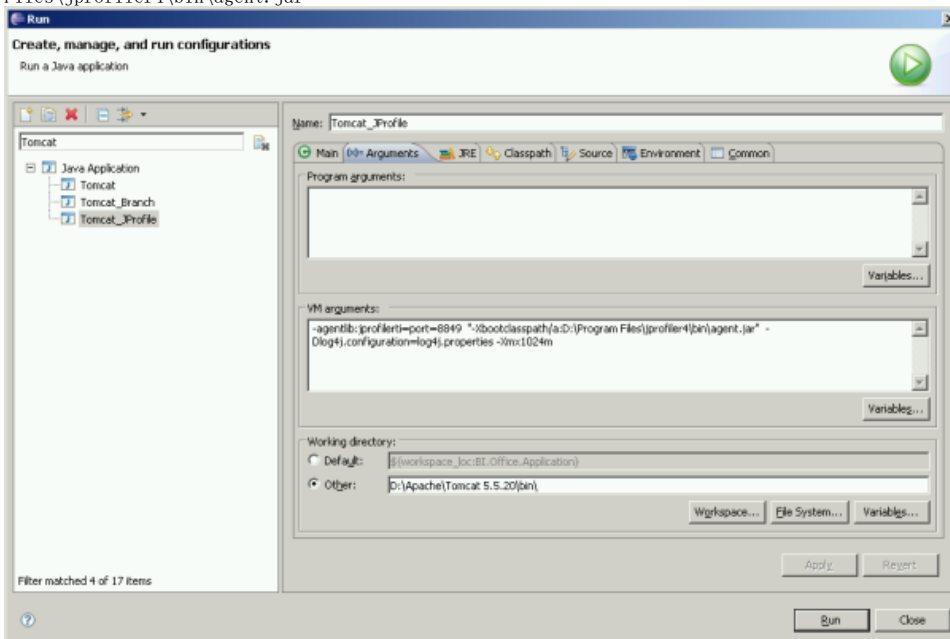
若在Eclipse中启用JProfiler，可以按以下步骤操作：

- 1. 点击Eclipse中Run按钮旁边的下拉菜单，点击Open Run Dialog
- 2. 在Java Application中找到之前运行的配置（如Tomcat），点击  将配置复制一份并修改名称为Tomcat_JProfiler

3. 切换到Environment标签页中点击New，在Name中输入PATH、Value中输入D:\Program Files\jprofiler4\bin\windows（请根据实际情况修改）并点击确定



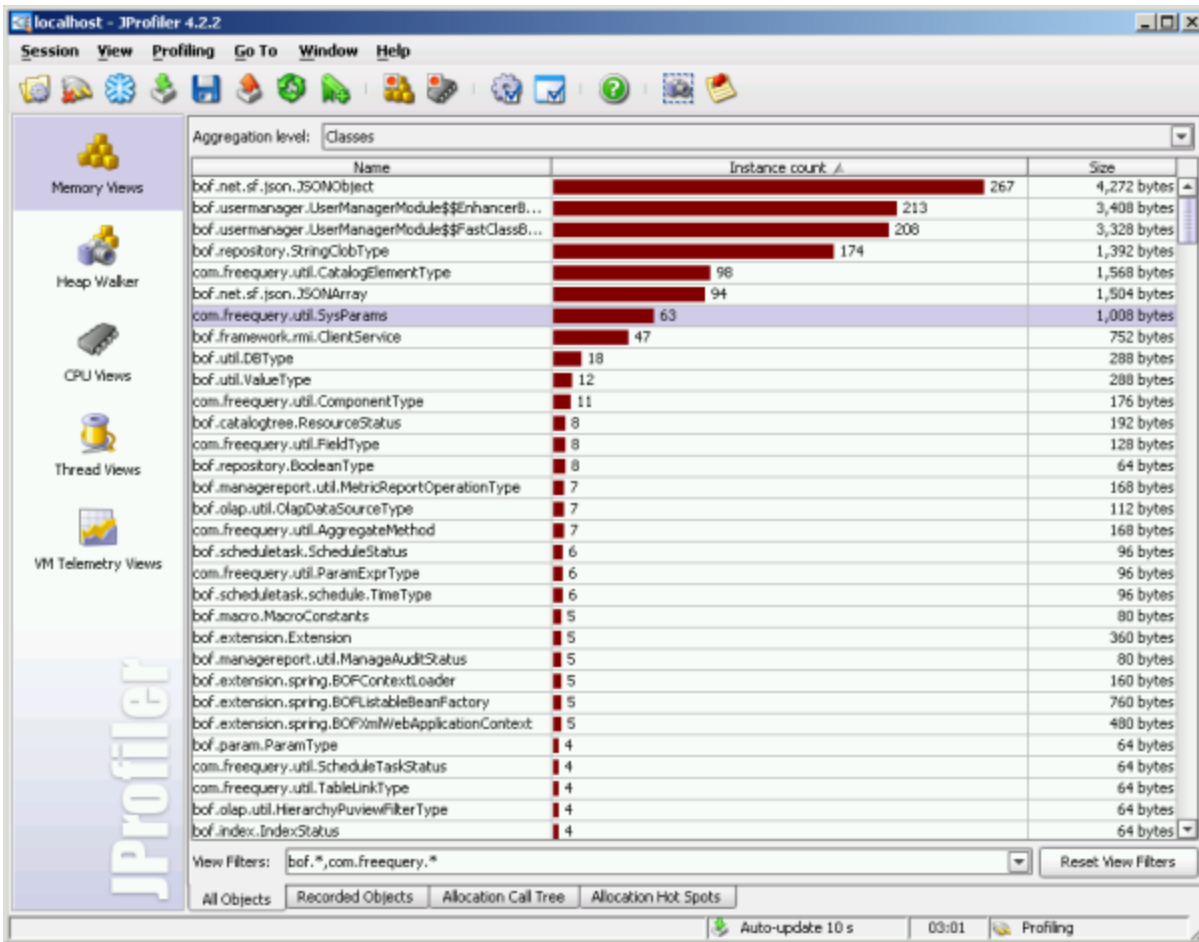
4. 切换到Arguments标签页中，在VM arguments中增加“-agentlib:jprofiler4=port=8849 -Xbootclasspath/ad:\Program Files\jprofiler4\bin\agent.jar”



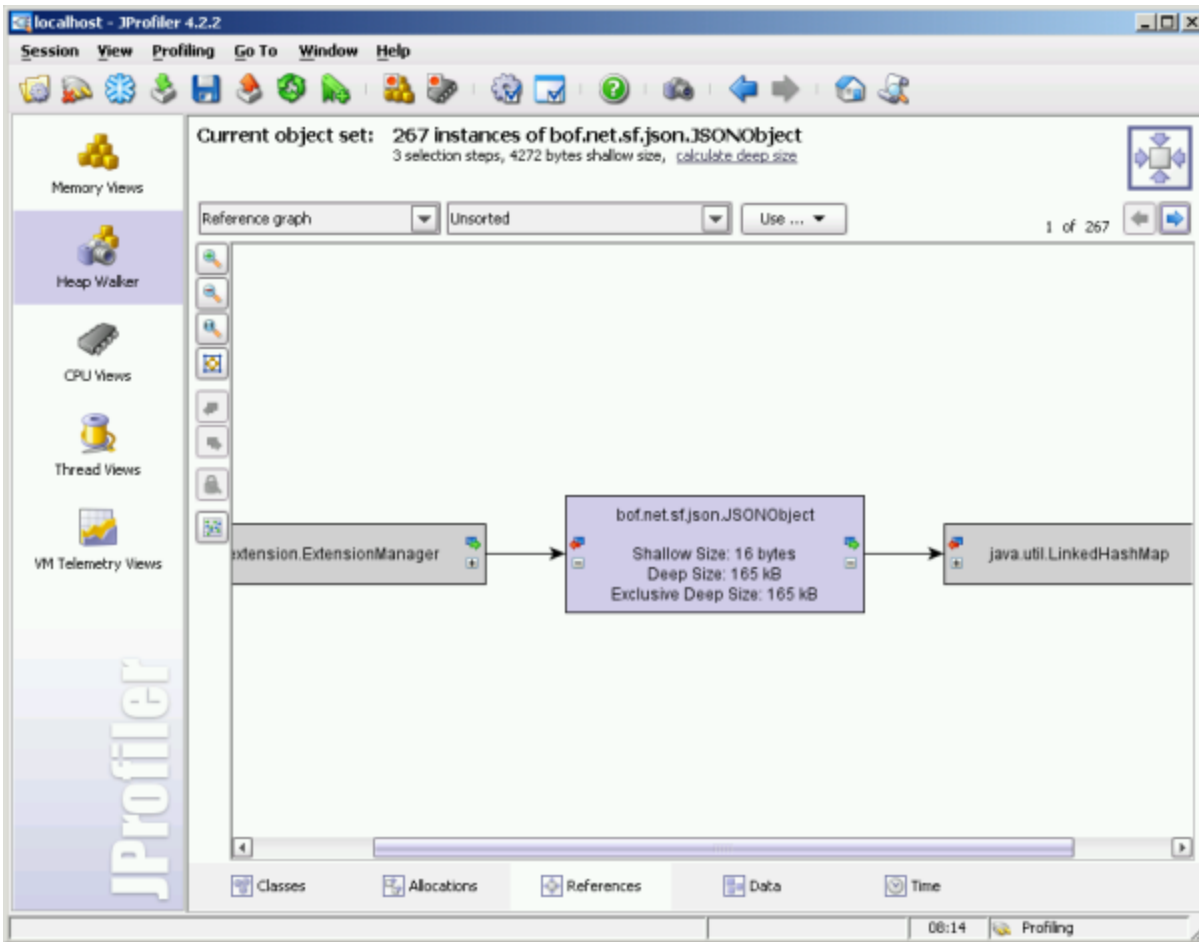
5. 点击“Run”就可以开始启动服务器，后续的操作参考上一部分的说明

Memory Views、Heap Walker

该页面列举当前JVM中使用的所有对象，通过在View Filters中输入“smarbtbi.,com.freequery.”等条件可以将过滤显示的类：



可以点击其中一个类右键“Take Heap Snapshot for Selection”这个类的堆镜像，然后双击这个类就可以显示这些实例是被哪个类初始化（Allocations）或者目前被谁引用（References）



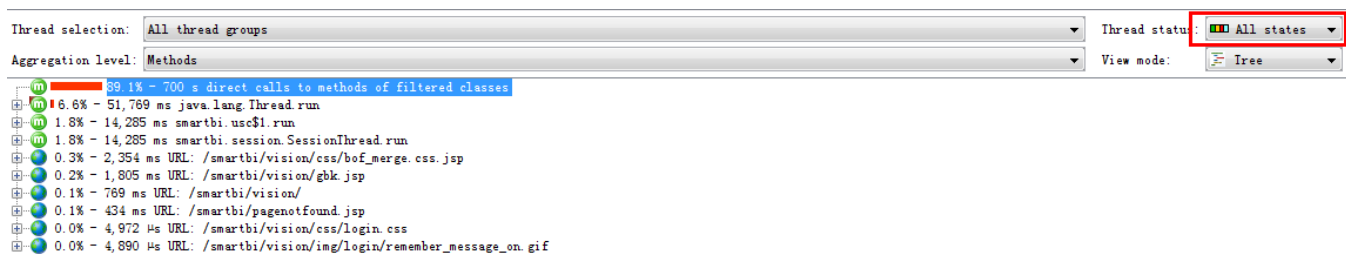
如上图显示的是bof.net.sf.json.JSONObject的实例引用情况，正在显示267个实例中的第1个。从中间的图形可以看出它被bof.extension.ExtensionManager引用。

通过查看各个实例的被引用情况可以发现代码是否存在没有正确释放内存的情况。

CPU Views



该视图是跟踪服务器运行性能，分析各个方法调用次数、运行花费的时间。切换到这个视图后，然后点击开始记录服务器的CPU运行情况，在操作完成后再点击一次即可停止录制。在使用中应该在操作Smartbi前记录在完成操作后马上停止，防止后续的操作影响当次记录的准确性。



JProfiler会以树的方式显示CPU的占用情况，每一个树节点都包含以下信息：

- 占用时间的百分比
- 占用的总时间
- 调用的次数，13 inv即表示调用了13次
- 类与方法的全名

在树节点上可以右键“Set As Root”设置该节点为根（右键再点击Show All可取消）。设置某一个节点为根后，CPU Views只会显示该节点及其下的子节点，并且其下子节点的占用时间百分比是以该节点的总时间计算的。