

# 如何进行CPU采样跟踪

## 问题

经常出现页面查询反应慢，最终还是可以查出数据，没出现卡死现象，因为打印线程信息不好分析问题原因，可采取CPU采样跟踪。

步骤1：进入CPU采样界面，路径是定制管理>系统运维>试工具集>CPU采样跟踪（如下）

工具名称	描述
<a href="#">JavaMelody</a>	JavaMelody
<a href="#">SVG地图编辑器</a>	用于绘制HTML5地图
<a href="#">列出已加载扩展包</a>	用于查看当前系统已加载的扩展包及每个扩展包的用途
<a href="#">列出会话信息</a>	用于查看系统会话信息，如总会话数、总内存数、已用内存数、空闲内存数以及用户访问信息，通过这！性一直增加）
<a href="#">列出对象池信息</a>	用于查看系统对象的使用情况，使用后是否被释放，辅助分析系统性能问题
<a href="#">列出系统中对象个数</a>	该页面会遍历Smartbi应用中所有的类及依赖类库，如果该类已经加载则遍历它的属性找出所有引用关系题。一般情况不需要使用此页面
<a href="#">列出WSDL地址信息</a>	用于查看通过WebService调用Smartbi 服务器端接口，以满足非J2EE平台的客户环境（如.Net平台）实现
<a href="#">列出字体信息</a>	列出字体信息
<a href="#">列出缓存同步信息</a>	配置集群环境时，利用该页面查看缓存同步信息，从而确认集群配置是否正确。
<a href="#">线程堆栈</a>	用于定时将JVM所有的线程堆栈打印到临时目录中，通过查看线程堆栈日志，可以看出线程们都在干什么：慢、假死等情况
<a href="#">JVM堆打印</a>	该页面显示目前JVM堆中所有对象个数及占用的内存数，用于性能问题的监控、诊断和分析
<a href="#">CPU监控</a>	用于查看、监控服务器CPU使用情况，是性能分析的辅助工具
<a href="#">CPU采样跟踪</a>	用于查看、监控服务器方法调用时间树结构
<a href="#">线程CPU监控</a>	线程CPU监控
<a href="#">内存监控</a>	以图形方式显示当前JVM所占用的内存，在该图形中Y轴的最大值表示服务器最大可以申请的内存（max色线表示当前已经使用的内存（totalMemory - freeMemory）
<a href="#">查找加载类路径</a>	通过输入类名，查看类所在路径。用于解决系统出现NoClassDefineFound之类的异常
<a href="#">分组广播测试</a>	用于测试网络是否能够支持分组广播，在集群中需要使用EhCache中的广播机制通知其它服务器，但是在

步骤2：进入CPU采样跟踪界面，点击开始按钮：



步骤3：访问smartbi，打开慢的报表，直到加载完成。返回CPU采样跟踪界面（cpusampling.jsp页面），下载文件并邮件发回（如截图）



