

线程堆栈

问题

系统出现无响应的状态，但此时其他用户登录还是可以访问我们系统的，若是这样可打印线程堆栈信息。

步骤1：进入线程堆栈界面，路径是定制管理>系统运维>调试工具集：



找到调试工具集，进入界面后，选择“线程堆栈”：

| 工具名称 | 描述 |
|----------------------------|---|
| JavaMelody | JavaMelody |
| SVG地图编辑器 | 用于绘制HTML5地图 |
| 列出已加载扩展包 | 用于查看当前系统已加载的扩展包及每个扩展包的用途 |
| 列出会话信息 | 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，通过这些信息确定是否存在内存泄漏（即资源没有正常释放，导致会话属性一直增加） |
| 列出对象池信息 | 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 |
| 列出系统中对象个数 | 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。该页面主要作用查找是否有一些static或者list等没有释放的问题。一般情况不需要使用此页面 |
| 列出WSDL地址信息 | 用于查看通过WebService调用Smartbi服务器接口，以满足非J2EE平台的客户环境（如.Net平台）实现与Smartbi系统的集成要求 |
| 列出字体信息 | 列出字体信息 |
| 列出缓存同步信息 | 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。 |
| 线程堆栈 | 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，哪些资源存在争用，线程有无死锁，方便定位服务器运行缓慢、假死等情况 |
| JVM堆打印 | 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 |
| CPU采样跟踪 | 用于查看、监控服务器方法调用时间树结构 |
| 线程CPU监控 | 线程CPU监控 |
| 查找加载类路径 | 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 |
| 分组广播测试 | 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，但是在某些网络环境中分组广播可能被禁止，该页面是提供测试使用的 |
| 录制诊断数据 | 用于录制诊断数据 |
| 使用诊断数据 | 使用录制诊断数据生成的文件 |
| 重建索引 | 重新创建所有索引信息。 |
| 系统检查 | 用于检查知识库信息的完整性与正确性 |
| 运行监控 | 用于监控服务器目前运行情况 |
| GC监控 | 用于监控服务器GC运行情况 |
| Log4配置 | 用于获取、修改Log4j的级别配置 |

步骤2：打印线程堆栈，先开始打印，然后再次返回打开有问题的报表（或者进行出现无响应的操作），打印几分钟（3、4分钟差不多）就结束打印，然后把堆栈信息发回。

打印间隔: 10 秒 停止打印

线程堆栈日志目录: D:\Smartbi\tomcat_60_smartbi_16000\temp\ThreadDumpList\141614608017252761.dir

当前线程堆栈

2016-11-22 10:49:40

Full thread dump Java HotSpot(TM) 64-Bit Server VM (24.51-b03 mixed mode):

```

"Thread Dump Timer" daemon prio=6 tid=0x00000000ce16000 nid=0x24d0 in Object.wait() [0x00000001c79f000]
  java.lang.Thread.State: TIMED_WAITING (on object monitor)
    at java.lang.Object.wait(Native Method)
    at java.util.TimerThread.mainLoop(Timer.java:552)
    - locked <0x00000007f331fa00> (a java.util.TaskQueue)
    at java.util.TimerThread.run(Timer.java:505)

```

Locked ownable synchronizers:
- None

```

"MySQL Statement Cancellation Timer" daemon prio=6 tid=0x00000000ce16800 nid=0x1988 in Object.wait() [0x00000001d30f000]
  java.lang.Thread.State: WAITING (on object monitor)
    at java.lang.Object.wait(Native Method)
    at java.lang.Object.wait(Object.java:503)
    at java.util.TimerThread.mainLoop(Timer.java:526)
    - locked <0x000000079c00bf18> (a java.util.TaskQueue)
    at java.util.TimerThread.run(Timer.java:505)

```

Locked ownable synchronizers:
- None

```

"MySQL Statement Cancellation Timer" daemon prio=6 tid=0x00000000ce68000 nid=0xa44 in Object.wait() [0x00000001d02f000]
  java.lang.Thread.State: WAITING (on object monitor)
    at java.lang.Object.wait(Native Method)
    at java.lang.Object.wait(Object.java:503)
    at java.util.TimerThread.mainLoop(Timer.java:526)
    - locked <0x000000079b589ce8> (a java.util.TaskQueue)
    at java.util.TimerThread.run(Timer.java:505)

```

Locked ownable synchronizers:
- None

打印线程堆栈，先开始打印，然后再次返回打开有问题的报表（或者进行出现无响应的操作），打印几分钟（3、4分钟差不多）就结束打印，然后把堆栈信息发回