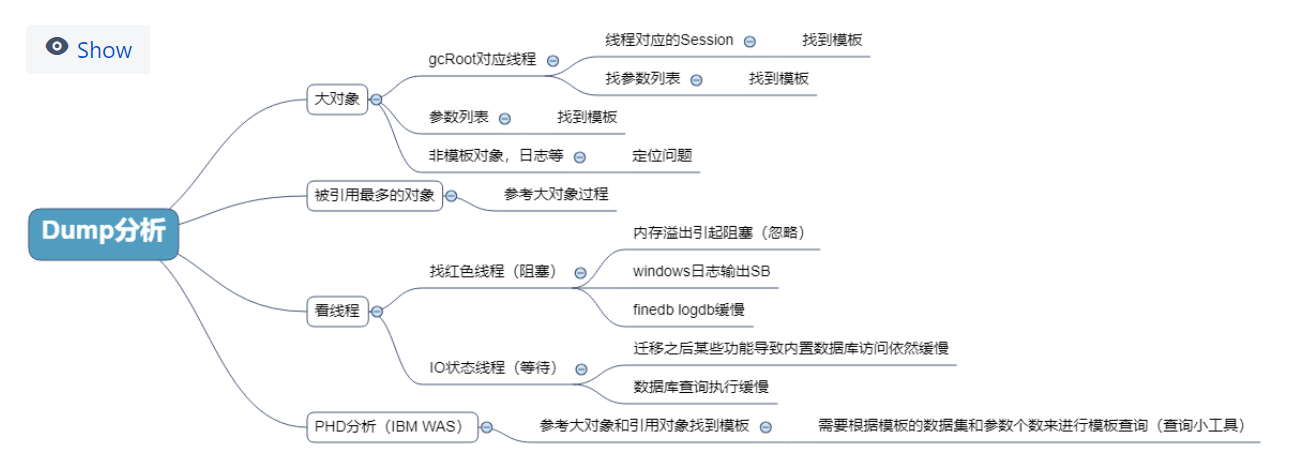
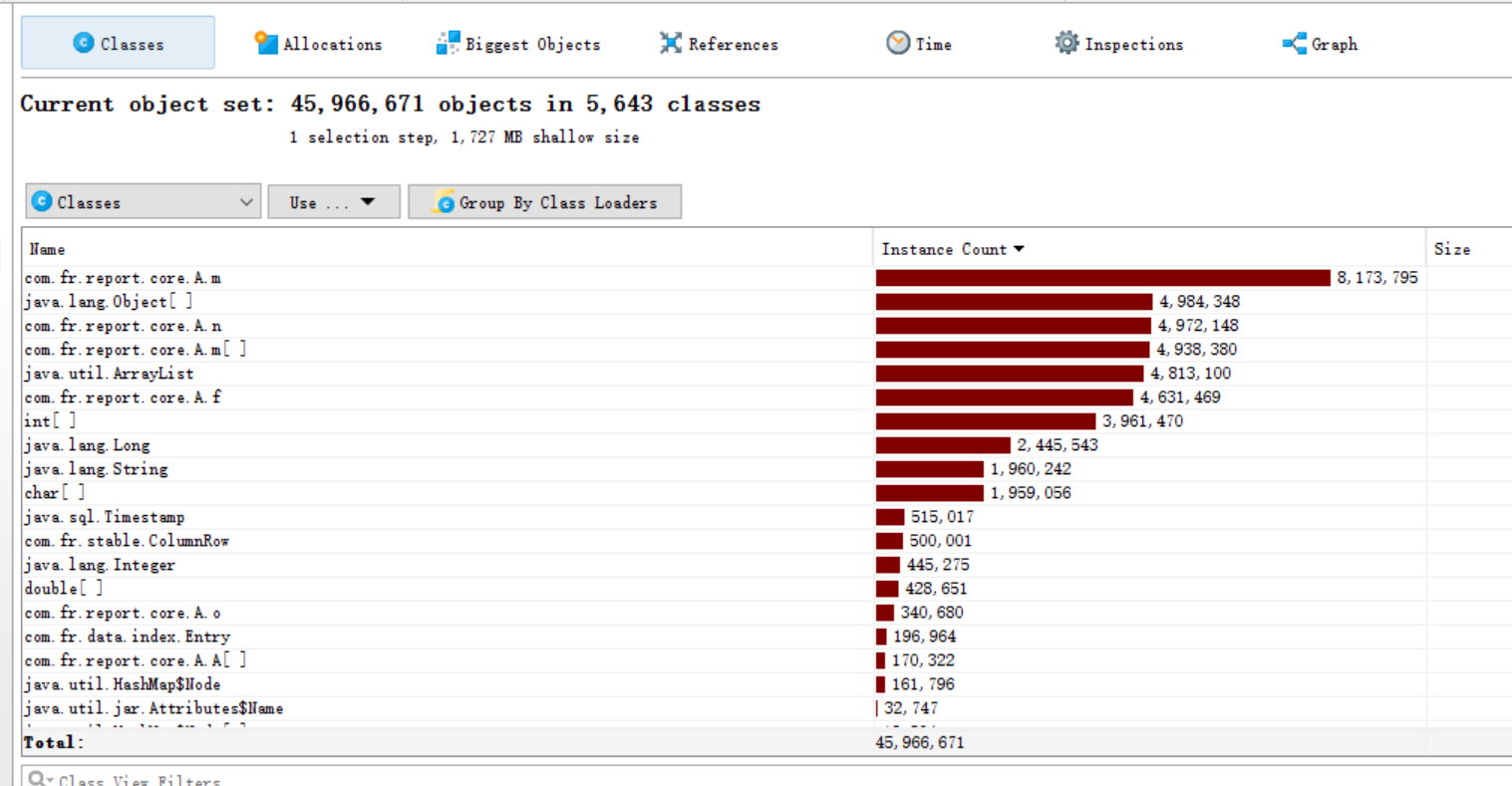
**Dump1分析**

什么是dump（宕机）呢？通俗来讲就是说电脑或手机down机吧,意思就是死机或是系统出现问题。广义宕机的意思是所有向服务器的请求都没有响应或者响应非常缓慢的情况都可以称为服务器宕机，这里有必要解释一下的是前端崩溃不是宕机。

宕机分析的目的就是要找到占用内存的模板，以下是一个宕机分析的通用的模板过程。



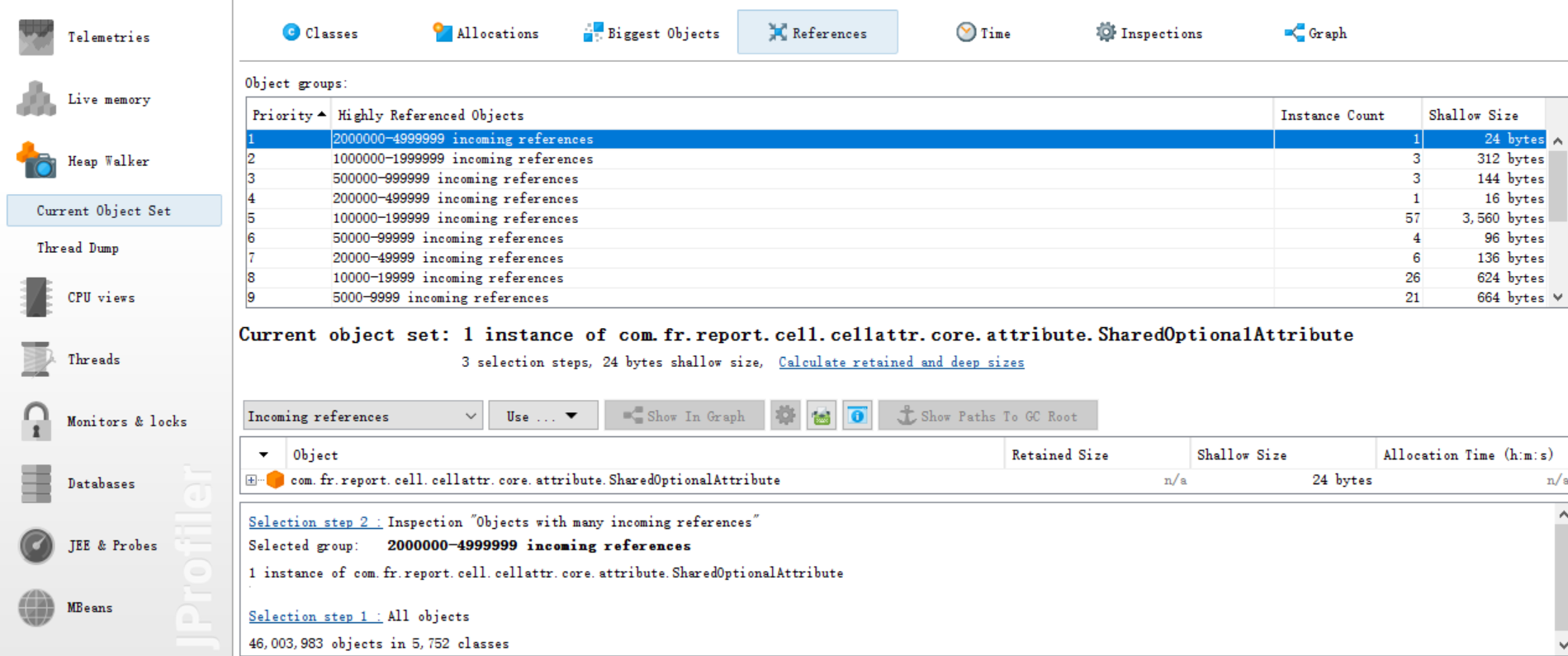
首先打开了一个dump1文件



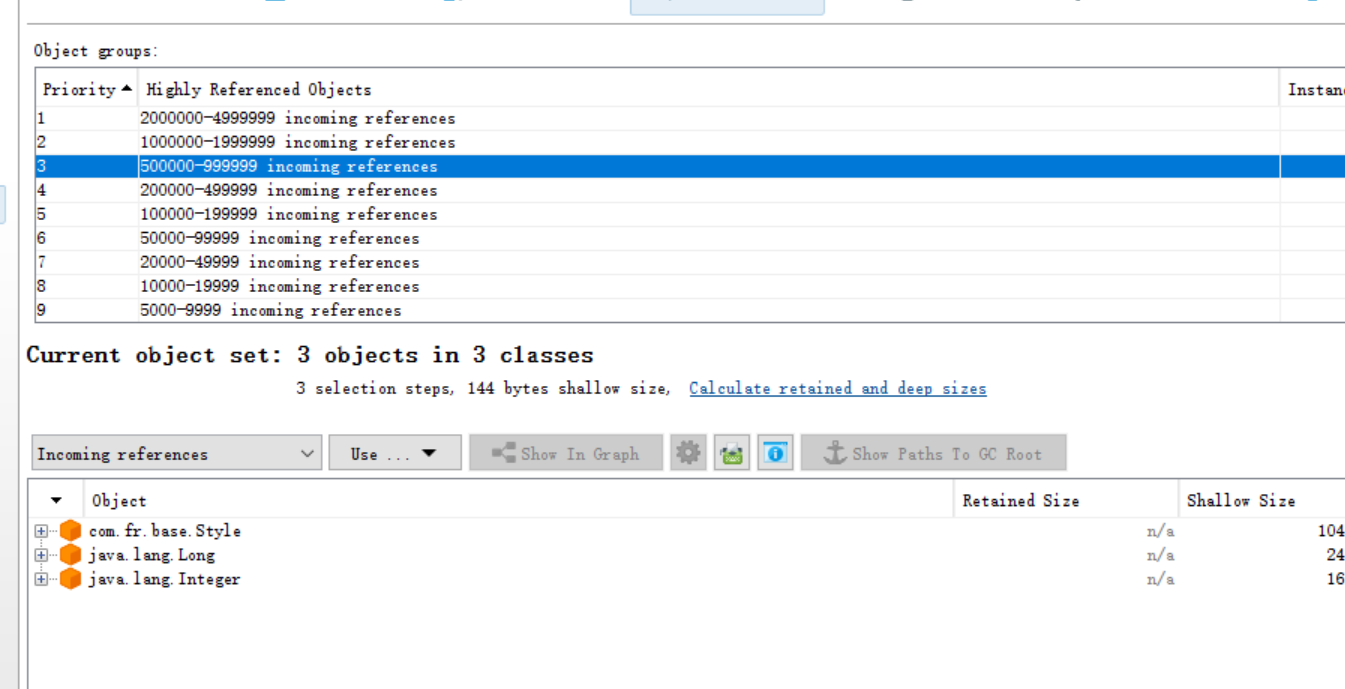
然而打开切换到Biggest Objects 看不到任何内容，这是一个BI模板。



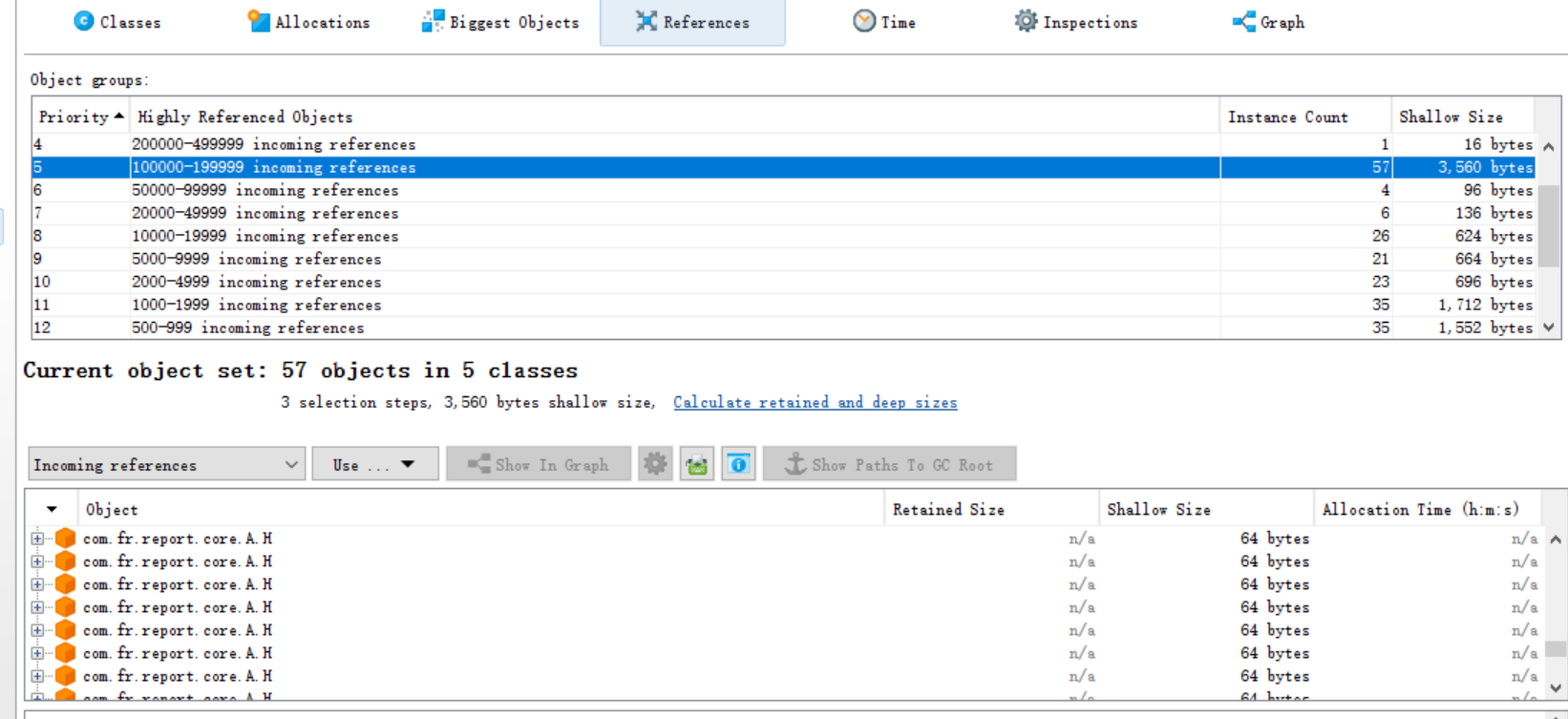
只能从引用出发解决问题：切换到Inspections ->reference & field analysis ->objects with many incoming references> show calculated results in a new Object set，得到下图

****

查看不同范围里的关键对象，一些百万级别的都是一些公共对象，没有什么问题



还有一个10W-20W之间，存在大量的关键对象com.fr.report.core.A.H，这才是需要的，如果这些对象出现在百万级别的区间，那么他就是有问题的，出现在1w-2w 属于正常现象，这是报表格子对象的引用。

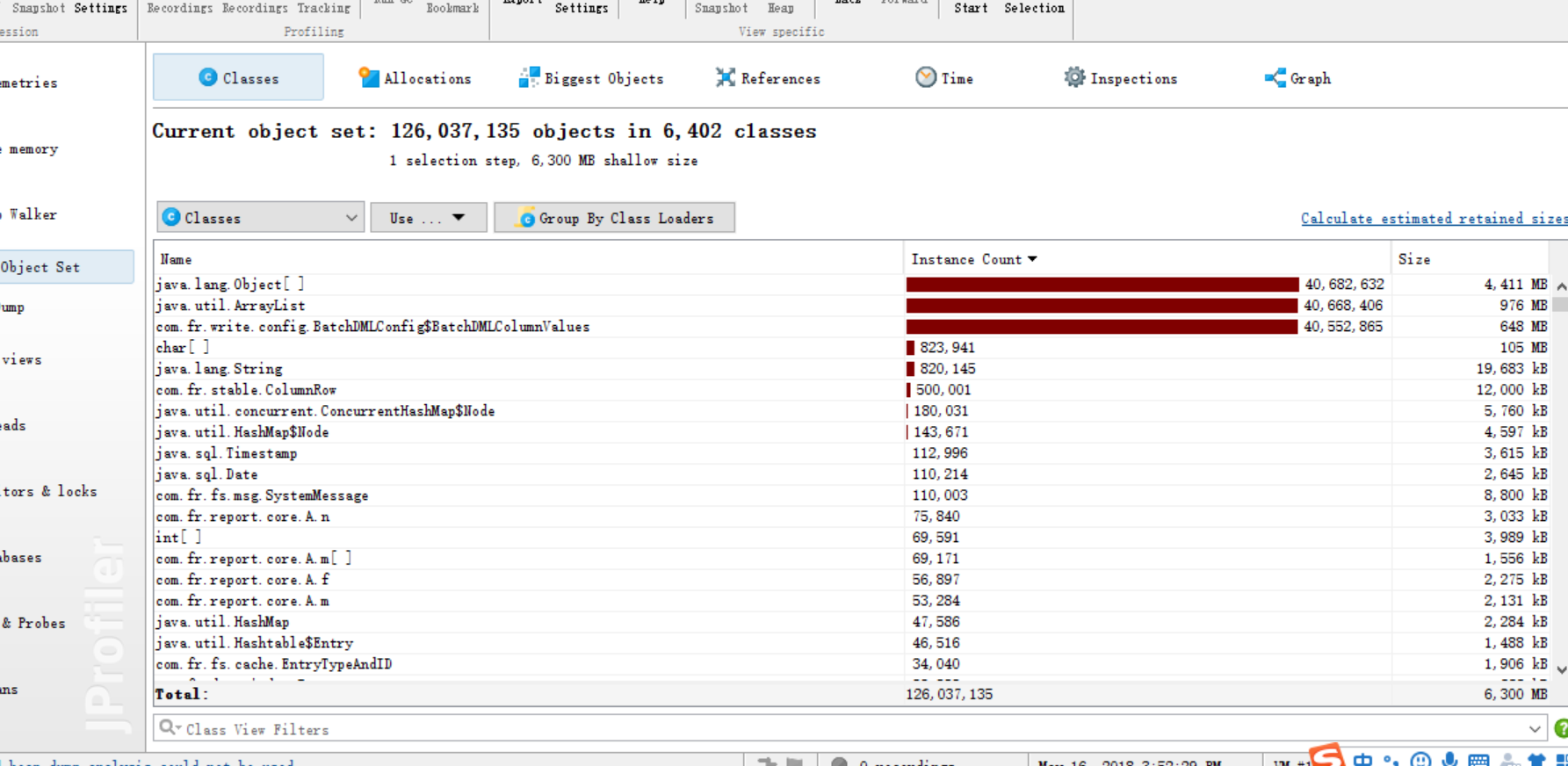


由上面说明并没有大的报表计算，上面日志查询的缓存也只是占用了较多的内存，并不是引起宕机问题的关键。

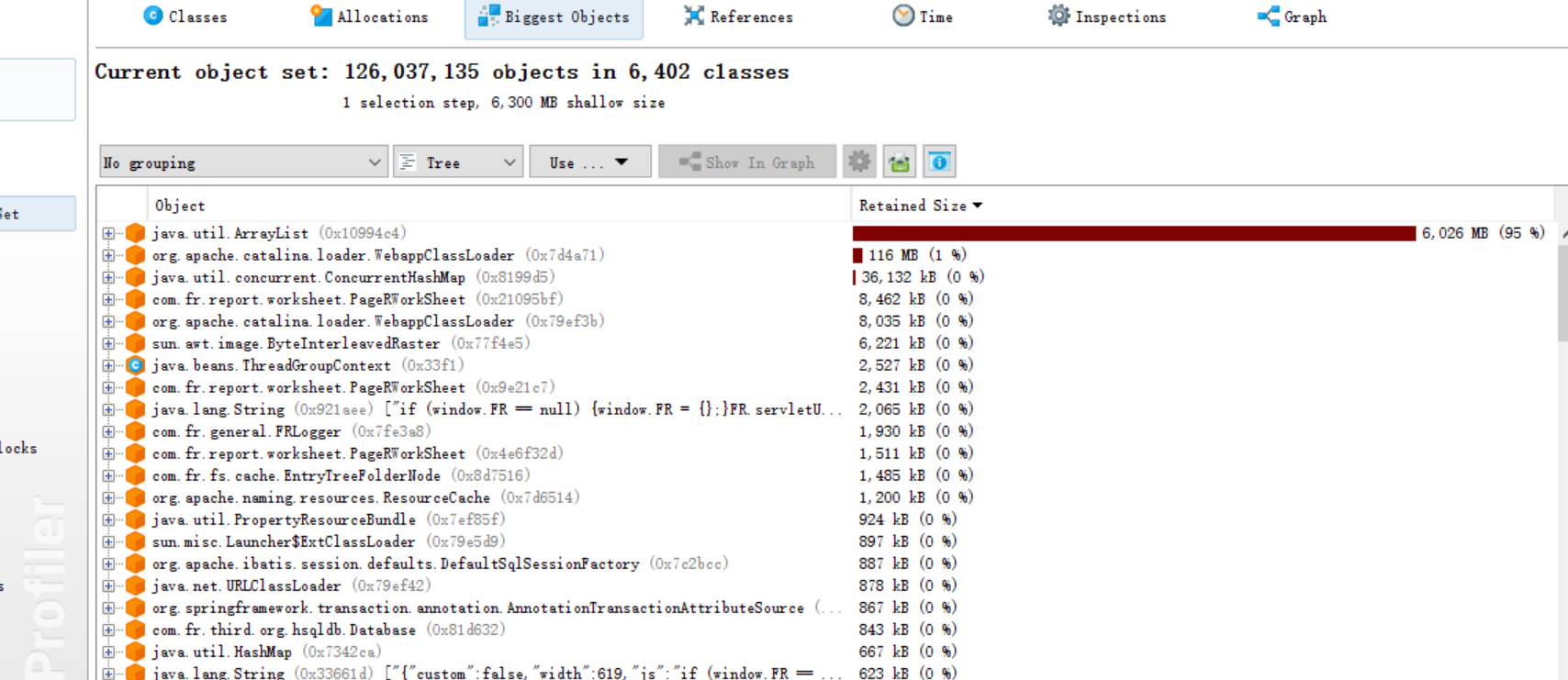
说明该模板没有什么问题。

**Dupm2的分析**

打开dupm2文件

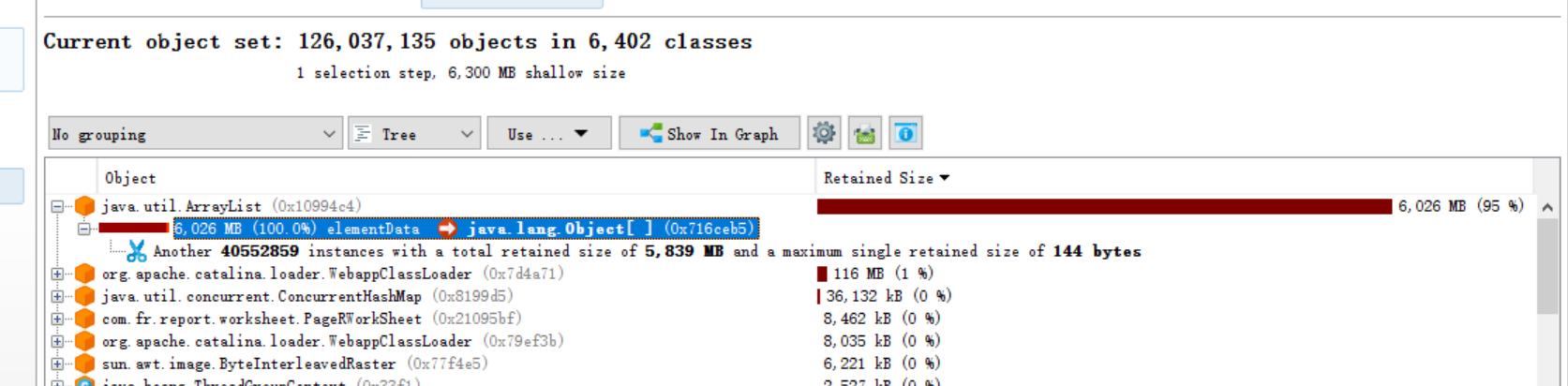


选择 Biggest Objects，明显看到一个很大的文件



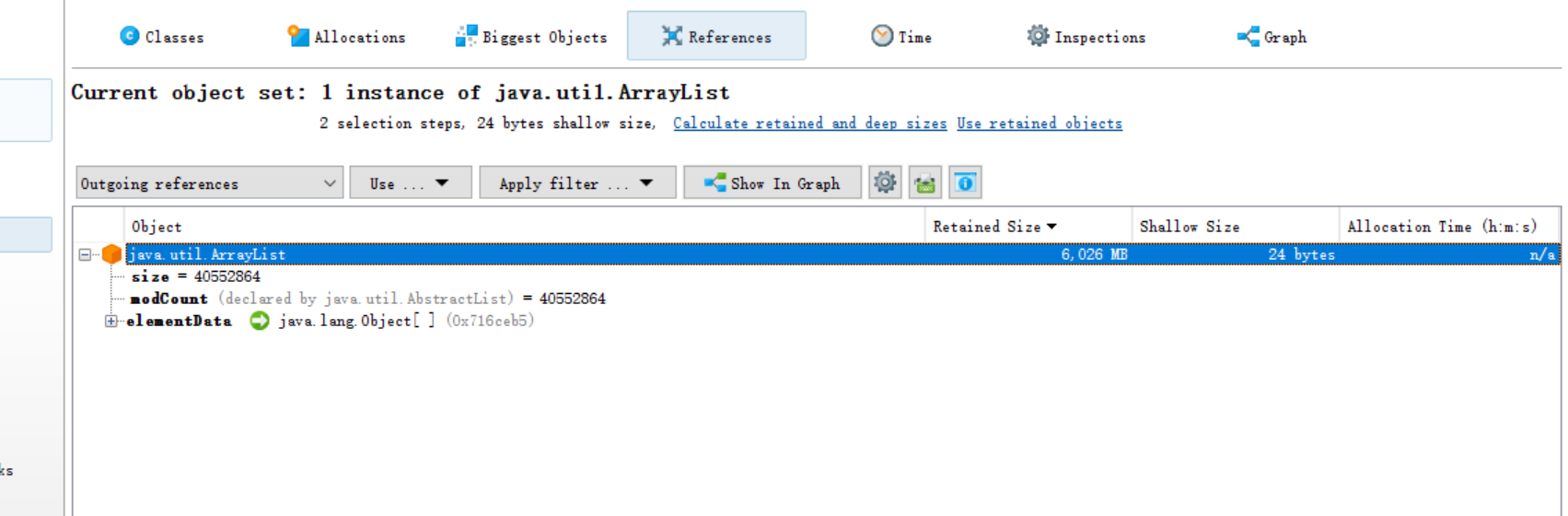
在标题那边显示总共6300MB内存远小于设置的12g 这个内存应该没慢

点开java.util.ArrayList



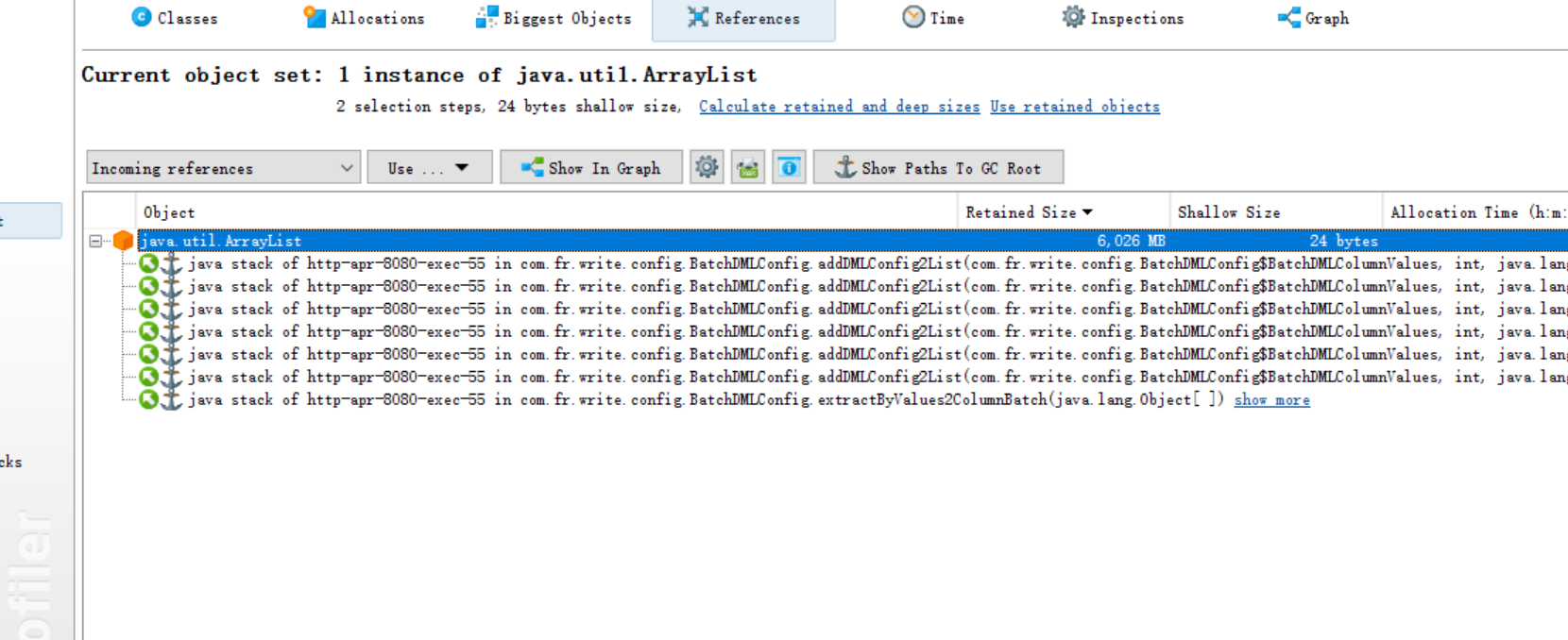
可以发现结果集有40000000+，只有唯一一个占用100%内6026MB

右击java.util.ArrayList，选择 Use Select Instances，将弹出界面点开看到



发现是一个40552864长度的数组...内部的这个对象也是填报的对象，所以可以肯定这是个填报的笛卡儿积问题.

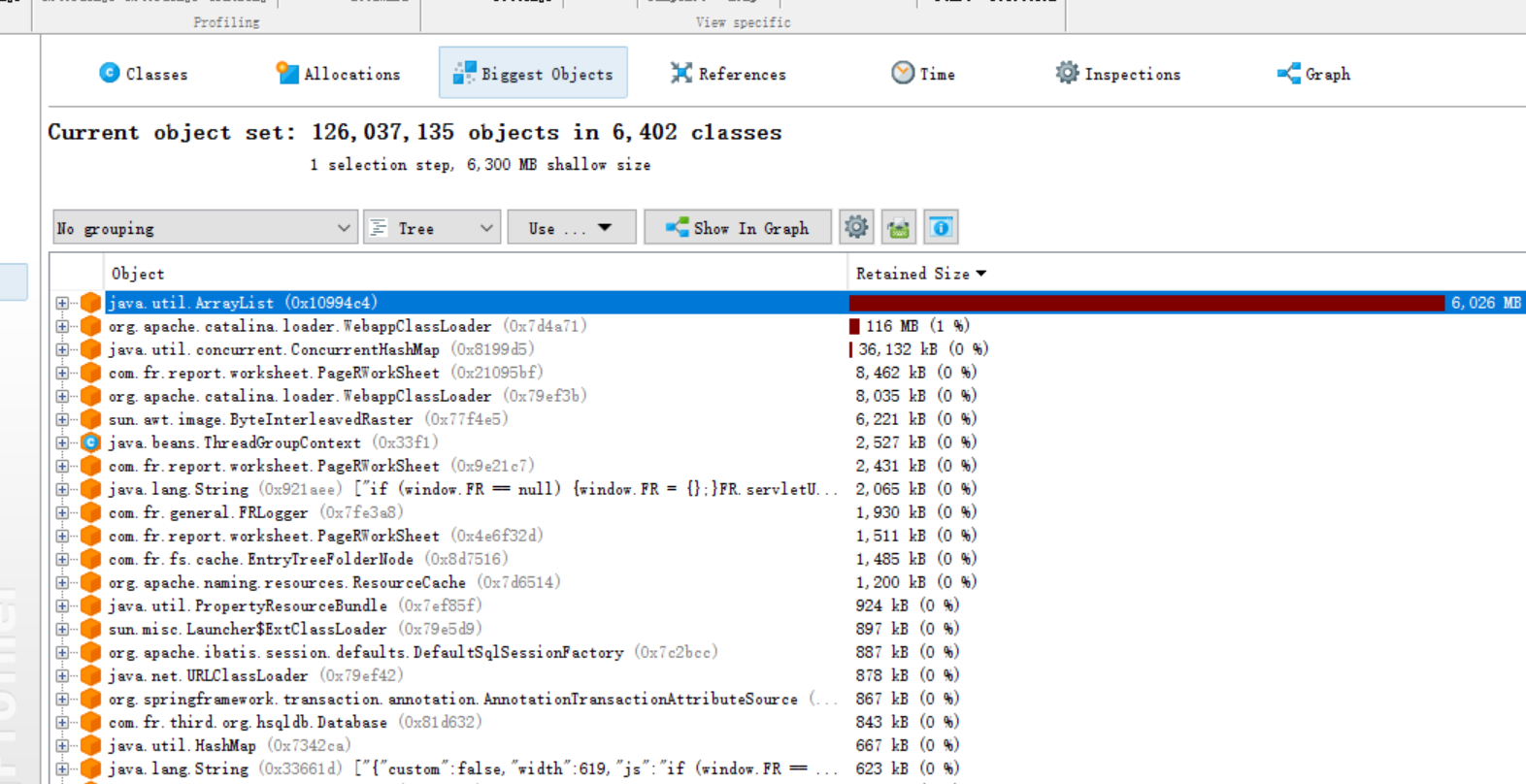
下面去找模板,切换到Incoming references 选中ArrayList，点击showPaths to GC root



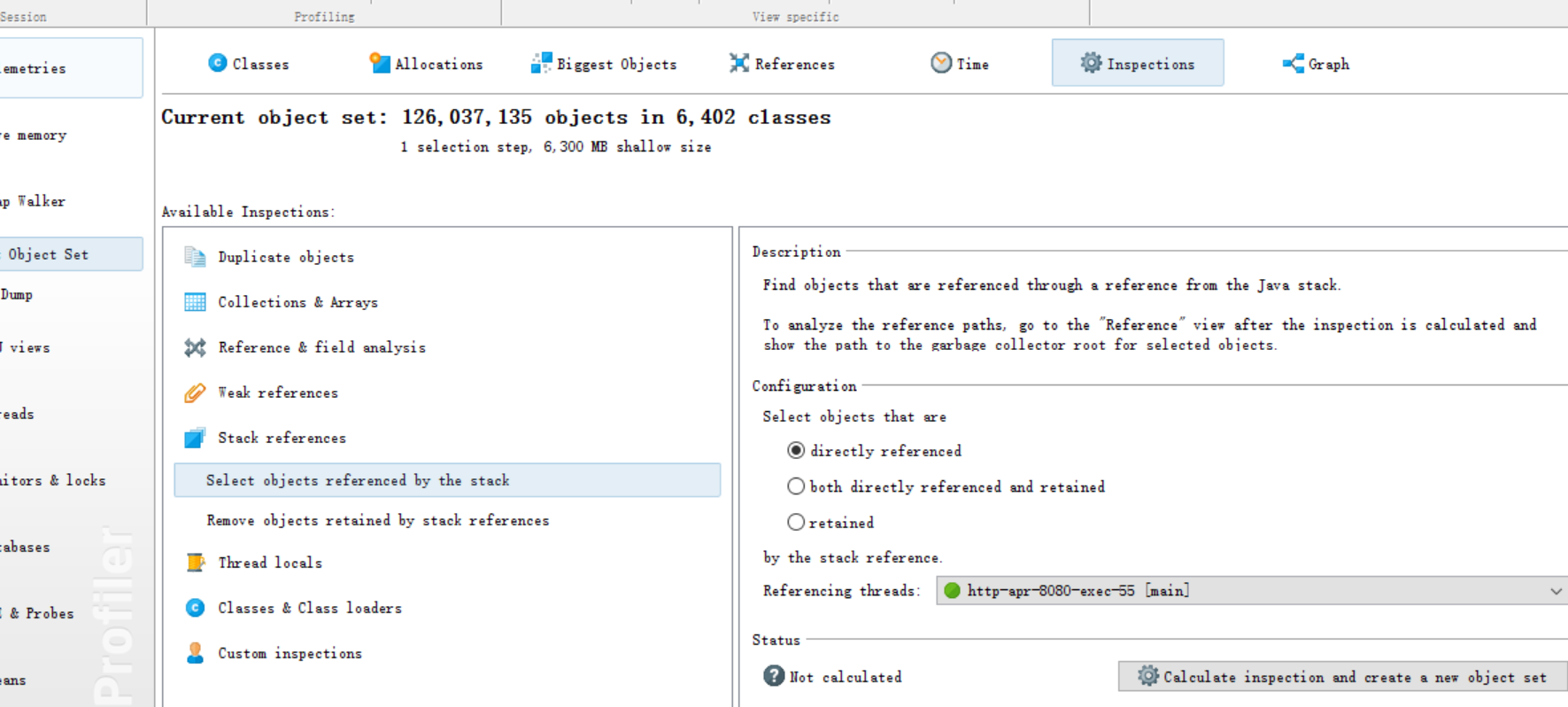
可以发现这是一个线程堆栈的引用栈，引用线程 http-apr-8080-exec-55。

点击Go To Start

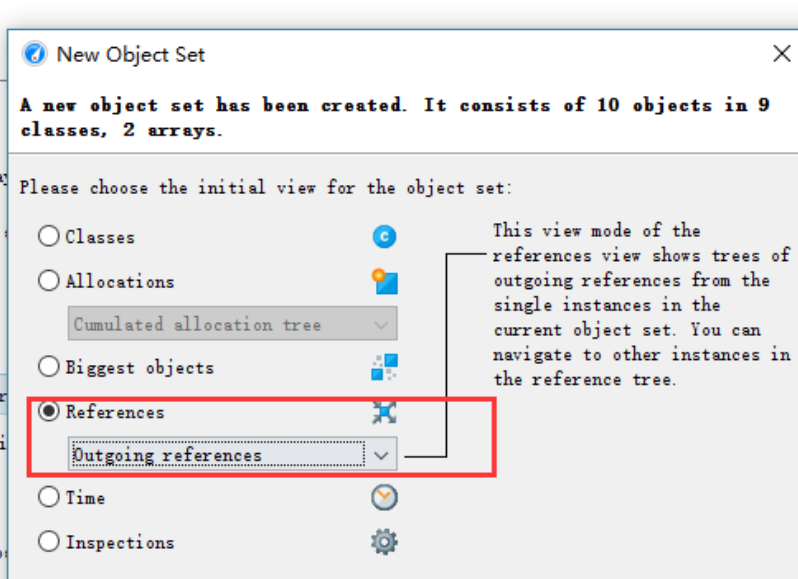
点击back按钮到最初的classes界面，这里为什么要返回 是因为jprofiler的每一步instance 操作相当于是做了一个过滤如果不返回下面所有的操作都是对这个对象。



点击 Inspections ->Stack references -> select objects referenced by the stack ->选择线程 55->calculate inspection and create a new object set

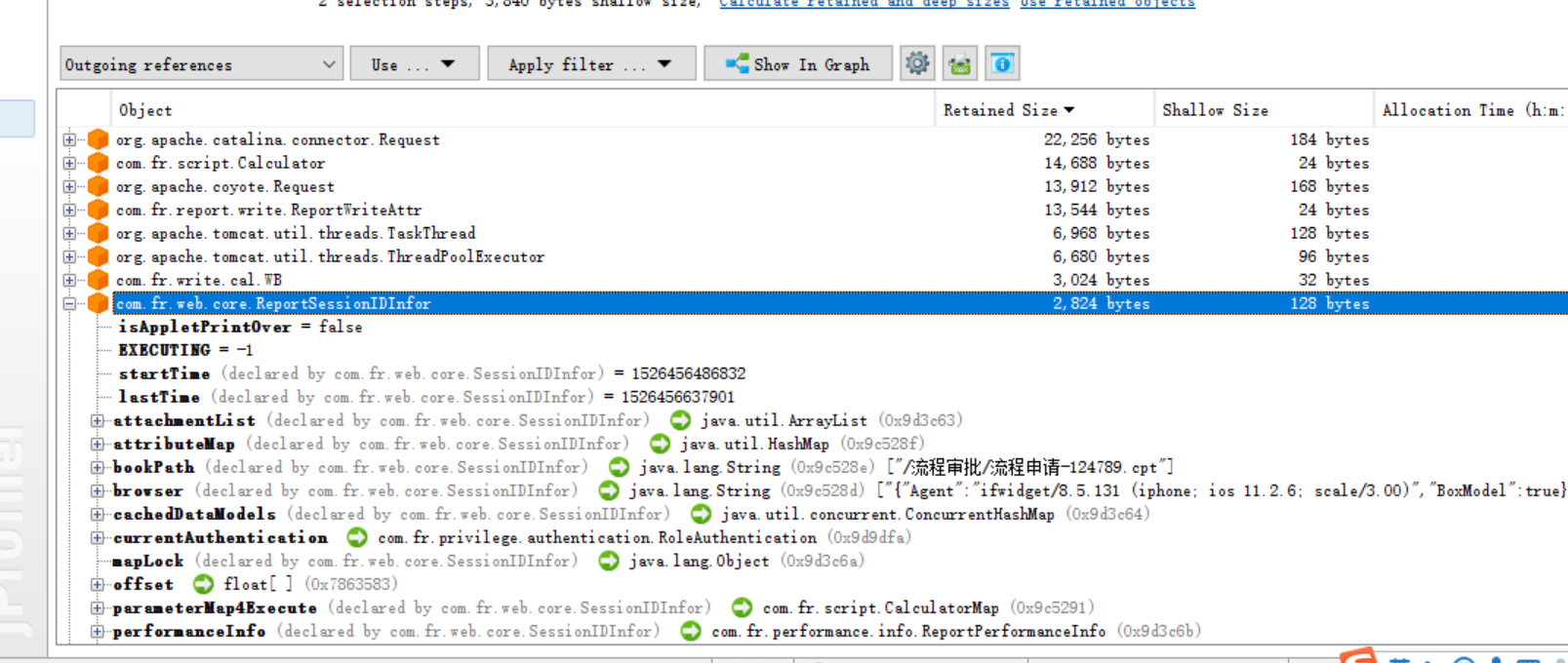


选择 outgoing references

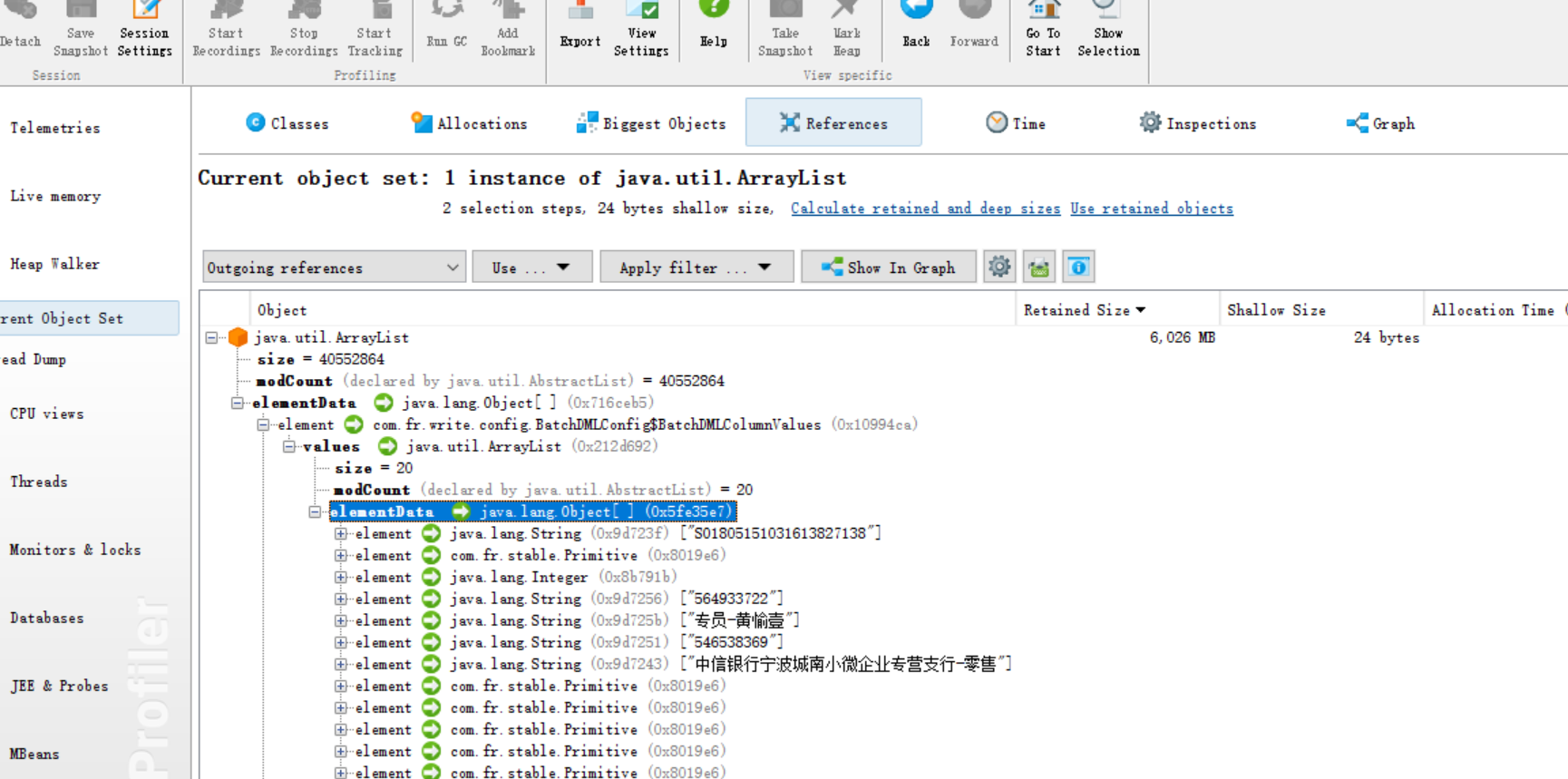


打开后, 搜索输入 ReportSessionIdInfo,

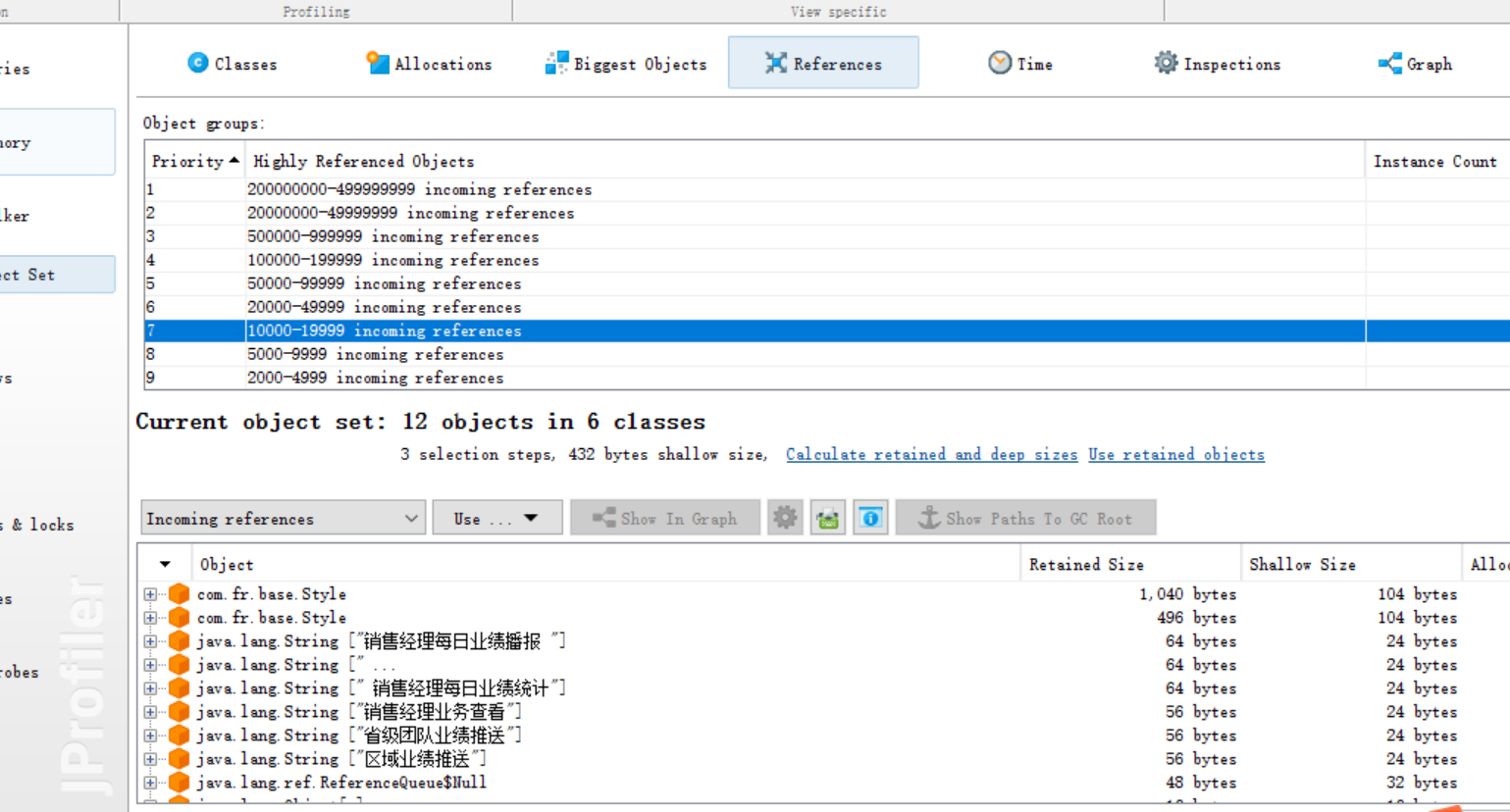
找到对应文件，展开后，找到问题模板 “/流程审批、流程申请.cpt”这个模板。

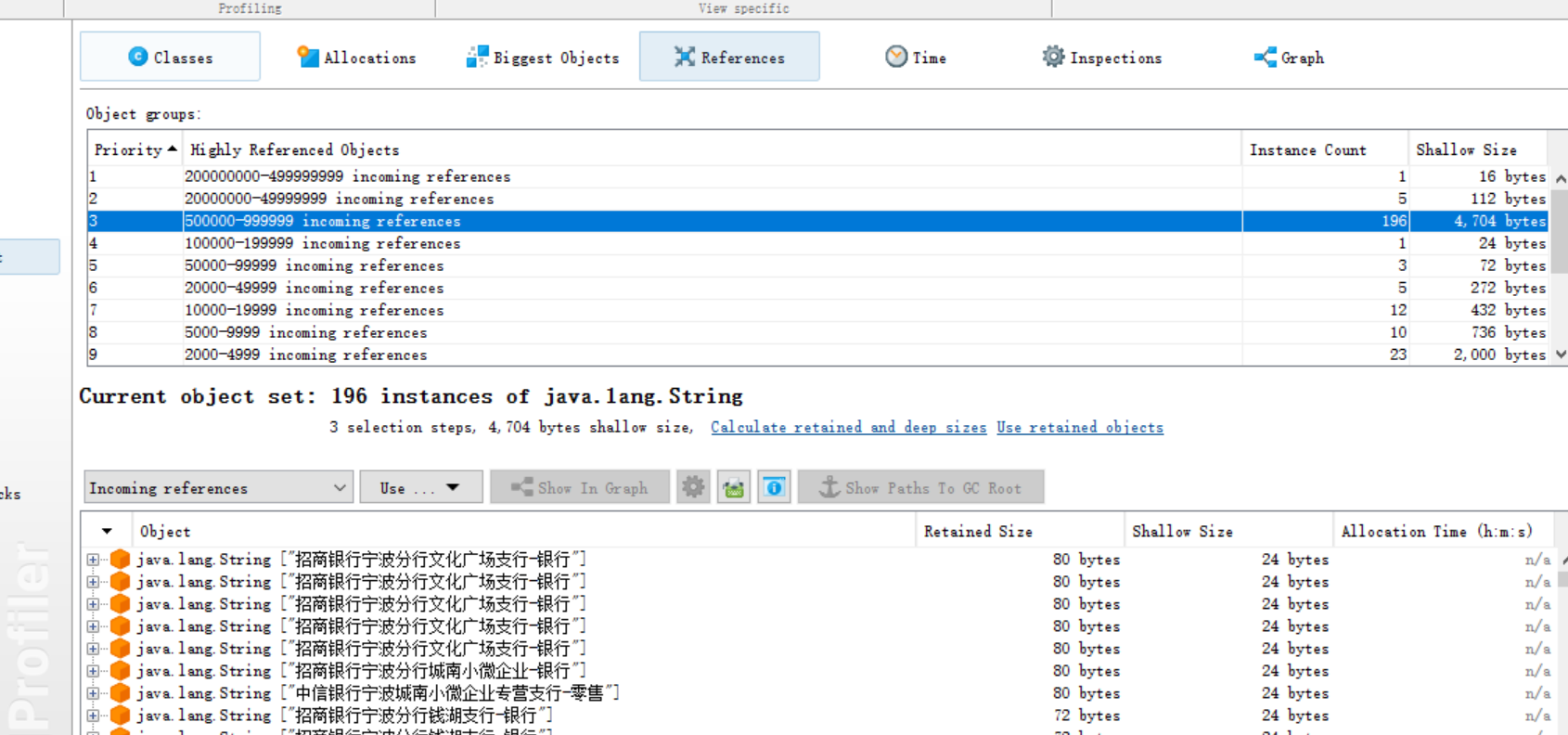


Go to start , 随便展开一下之前的填报List 可以得到笛卡儿积的一行数据：



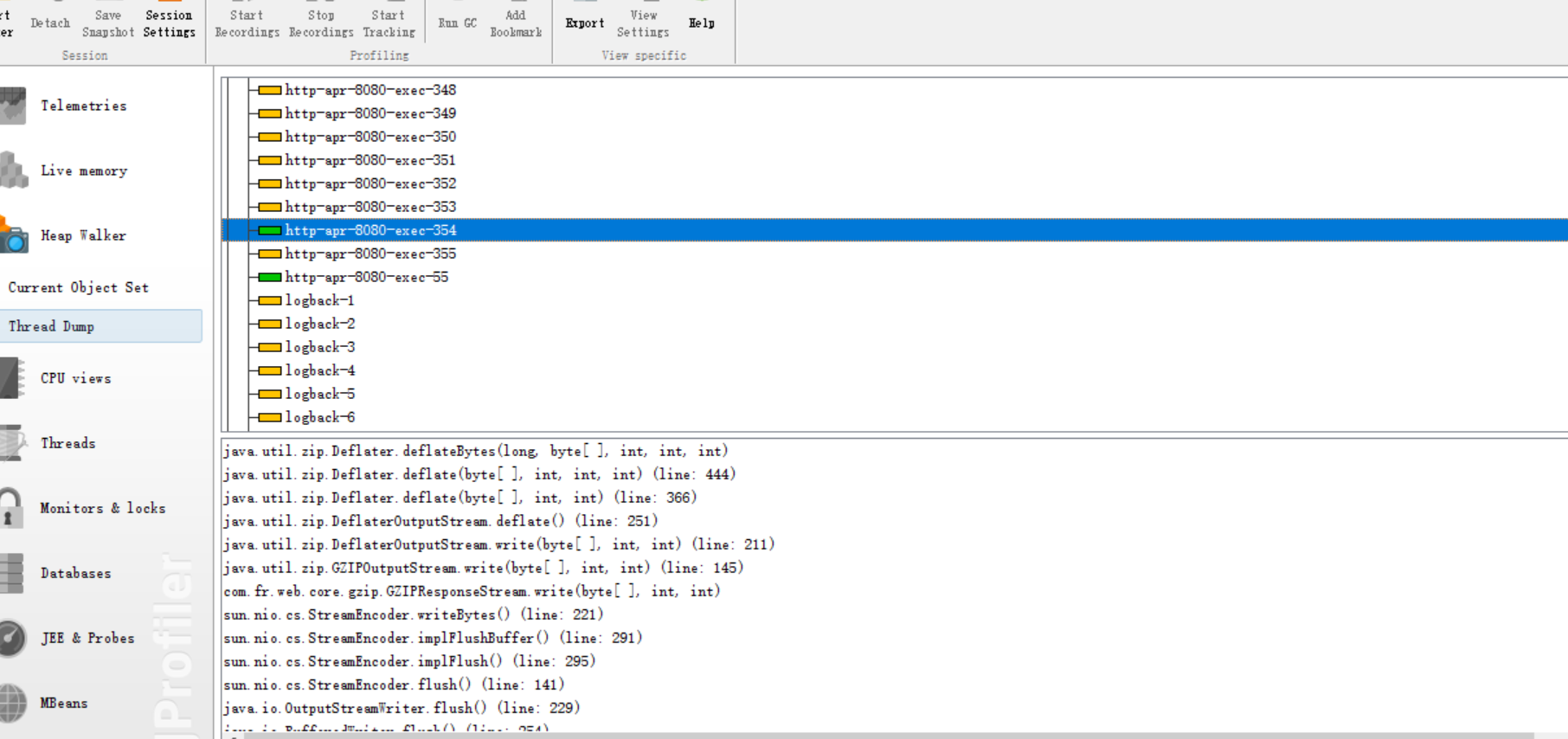
从引用出发解决问题：切换到Inspections ->reference & field analysis ->objects with many incoming references> show calculated results in a new Object set





可以看到数据都正常，说明并没有大的报表计算，上面日志查询的缓存也只是占用了较多的内存，并不是引起宕机问题的关键，

我们回到原点，继续检查线程



线程有上百个，在内存没有慢的情况下线程有如此多的等待数据库那么肯定数据查询缓慢了