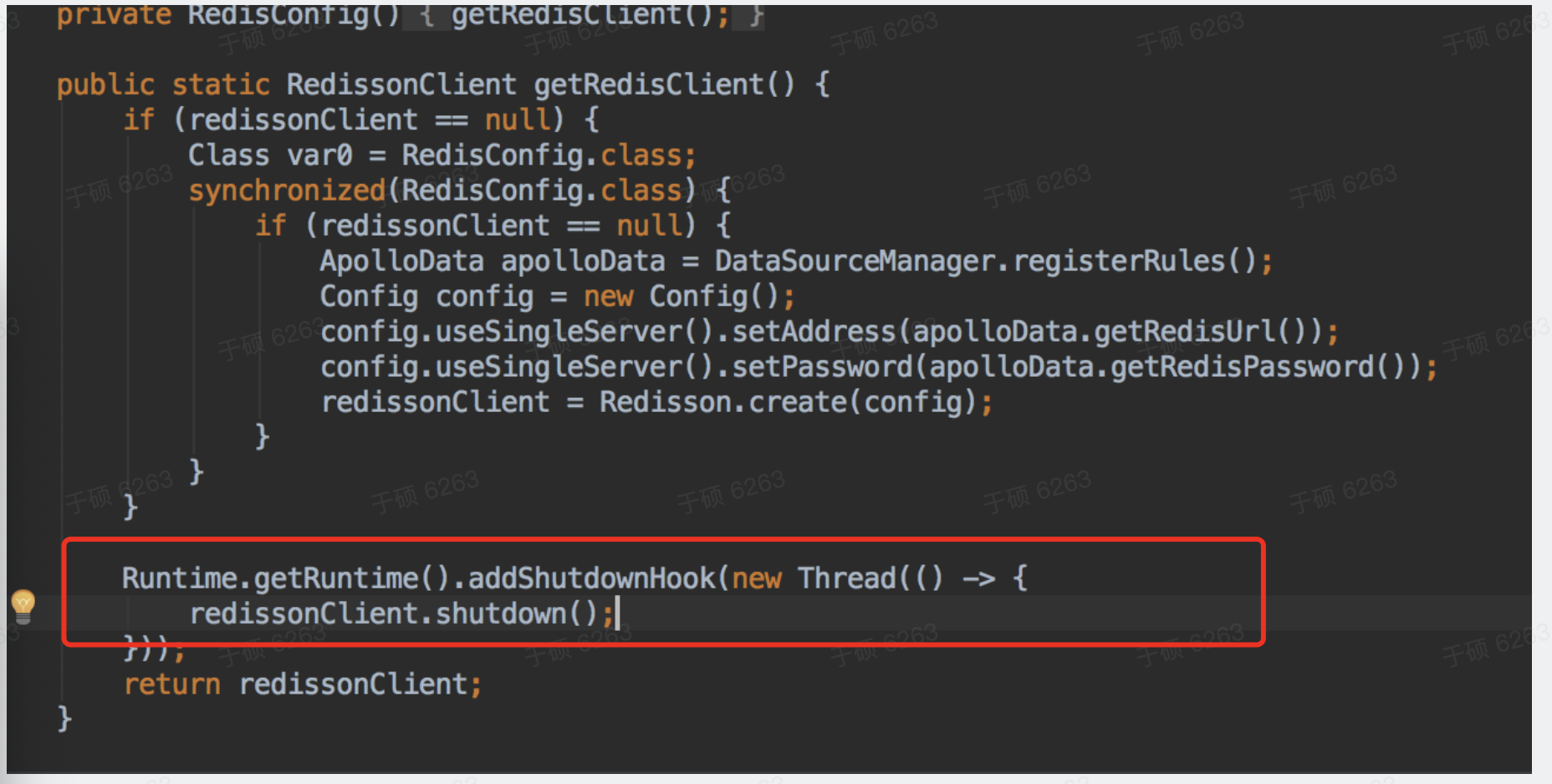
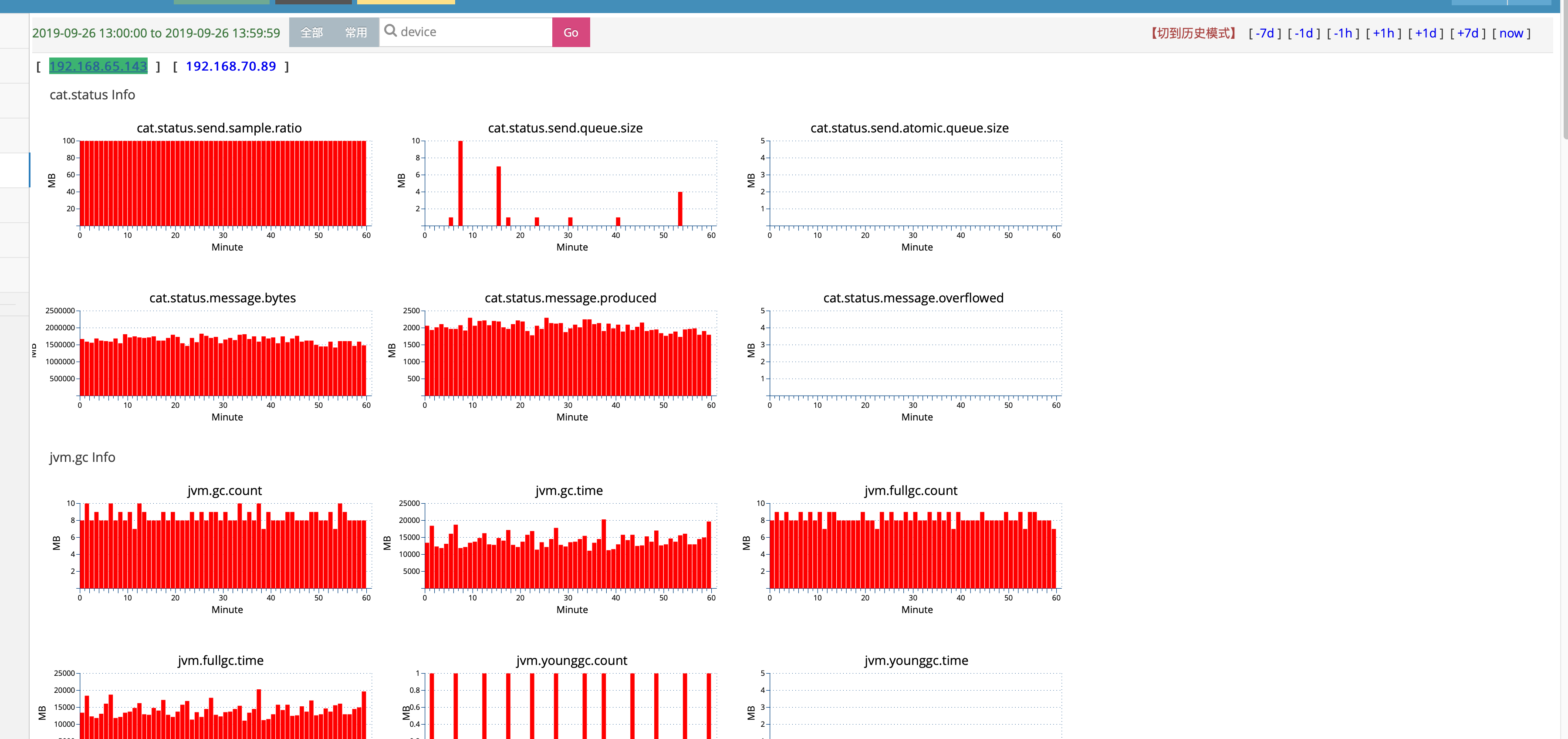
之前因为代码的问题引发了device项目出现了频繁fullgc的现象，当时由于不太会处理这种情况，所以在同事的帮助下，才查到问题的原因。

后面自己对这次的排查重新回顾了一遍并记录一下排查过程

起因：

图中的代码应该写到单例模式判断的条件内，但是写在了外面，导致每次获取client的时候都会新起一个shutdownHook线程并且无法回收，导致内存占用越来越大。

当程序正常退出,系统调用 System.exit方法或虚拟机被关闭时才会执行添加的shutdownHook线程。其中shutdownHook是一个已初始化但并不有启动的线程，当jvm关闭的时候，会执行系统中已经设置的所有通过方法addShutdownHook添加的钩子，当系统执行完这些钩子后，jvm才会关闭。所以可通过这些钩子在jvm关闭的时候进行内存清理、资源回收等工作

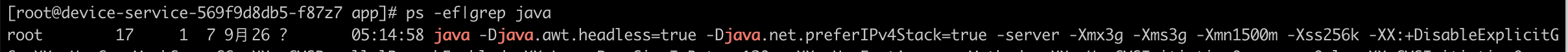


排查过程：

1.生成堆内存dump文件：

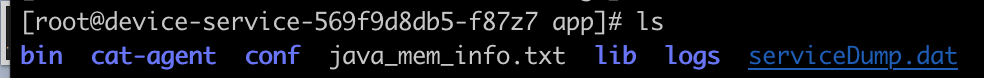
首先获取java进程对应的pid

ps -ef|grep java



打印出dump文件

jmap -dump:format=b,file=文件名 [pid]     如： jmap -dump:format=b,file=serviceDump 17



2.下载serviceDump文件到本地

由于源文件较大推荐压缩后再下载

mac推荐大家使用rzsz 具体文档可以参考这里 <https://www.jianshu.com/p/1166eb16413a>

3.利用jprof进行分析

从网络上下载jprof并安装，修改serviceDump.dat文件后缀名为.hprof，之后用jprof打开

可以看到堆内存的使用情况，shutdownHook占用了大量空间，后续对代码进行修改上线后解决问题。

