

无论是YGC或是FullGC，都会导致stop-the-world，即整个程序停止一些事务的处理，只有GC进程允许以进行垃圾回收，因此如果垃圾回收时间较长，部分web或socket程序，当终端连接的时候会报connetTimeOut或readTimeOut异常，

从JVM调优的角度来看，我们应该尽量避免发生YGC或FullGC，或者使得YGC和FullGC的时间足够的短。

所谓调优,就是找到适合自己程序的配置(jvm配置,比如,老年代大小,新生代大小,线程数,垃圾回收器等等)

原文链接：<https://blog.csdn.net/Javazhoumou/article/details/99298624>

使用工具：（java 自带分析工具）

1. **jmap**：生成虚拟机的内存转储快照（heapdump文件）,使用mat软件打开并分析

2. **JConsole**：JMX的可视化管理工具

这个工具相比较前面几个工具，使用率比较高，很重要。它是一个java GUI监视工具，可以以图表化的形式显示各种数据。并可通过远程连接监视远程的服务器VM。用java写的GUI程序，用来监控VM，并可监控远程的VM

3. **VisualVM**：多合一故障管理工具

这个工具也很牛bility。它同jconsole都是一个基于图形化界面的、可以查看本地及远程的JAVA GUI监控工具，Jvisualvm同jconsole的使用方式一样，直接在命令行打入jvisualvm即可启动，jvisualvm界面更美观一些

<https://baijiahao.baidu.com/s?id=1639024706303844305&wfr=spider&for=pc>

有时间就玩一下吧