命令 – 会记录下来 – 重点是过程，思路理解  
内存问题

JAVA应用常用配置： 6-16G区间  
1. JVM内存的占用（适用场景，启动后并发大，内存占用高）  
 -Xms -Xmx 最大最小定义一个区间。  
小优化点：启动时就向操作系统 占用内存  
-XX:+AlwaysPreTouch

2. 常见问题： 短时间内存用的多  
GC --- 引发的 Stop The World  
性能测试阶段应该被发现  
避免过多的GC  
System.gc() --- java代码触发。  
------ 实际问题：jar包依赖这么多，怎么去找到那一段分析那一段代码调用了System.gc()  
Btrace – 开源工具（命令行，集成JDK可视化工具 JvisualVM）  
  
3. OOM OutOfMemoryError不可预测  
 --- 留下证据 ---分析问题原因 ---   
把出问题的时候，内存的信息保留下来

*-XX:+HeapDumpOnOutOfMemoryError 生产环境标配*

jDK自带JvisualVM不太好用

网易案例：OOM案例  
因为前端一个查询条件，没有加时间限制

流程：浏览器 -🡪 后台系统 -🡪 订单服务中心 –》 数据库  
后台收到一个没有时间区间、传分页参数（后台没有默认值）

4. 堆外内存溢出 ---

生产环境标配堆外内存大小限制：  
-XX:MaxDirectMemorySize=128m  
   
避免出现服务器资源全部被耗尽的情况  
解决思路： 测试环境重新问题  
通过Btrace 跟踪堆外内存的申请

|  |
| --- |
| # 调用链  wget https://github.com/btraceio/btrace/releases/download/v1.3.11.3/btrace-bin-1.3.11.3.tgz  mkdir -p /usr/local/tools/btrace-bin-1.3.11.3  mv btrace-bin-1.3.11.3.tgz /usr/local/tools/btrace-bin-1.3.11.3/btrace-bin-1.3.11.3.tgz  cd /usr/local/tools/btrace-bin-1.3.11.3  tar -xvf btrace-bin-1.3.11.3.tgz  rm -rf btrace-bin-1.3.11.3.tgz |

命令行使用方式：  
/usr/local/tools/btrace-bin-1.3.11.3/bin/btrace -cp /usr/local/tools/btrace-bin-1.3.11.3/build/ 27086 /netease/omtest/app/TracingScript.java

Gperftools 谷歌提供性能优化的工具包—内存/CPU都可以

|  |
| --- |
| # 谷歌的内存申请器替代操作系统的，实现统计  mkdir /usr/local/tools  https://github.com/gperftools/gperftools  下载libunwind  wget http://ftp.twaren.net/Unix/NonGNU/libunwind/libunwind-1.1.tar.gz  tar -xvf libunwind-1.1.tar.gz  cd libunwind-1.1  ./configure --prefix=/usr/local/tools/libunwind/ CFLAGS=-U\_FORTRIFY\_SOURCE  make  make install  # 下载 https://github.com/gperftools/gperftools/releases  wget https://github.com/gperftools/gperftools/releases/download/gperftools-2.7/gperftools-2.7.tar.gz  tar -xvf gperftools-2.7.tar.gz  cd gperftools-2.7  ./configure --prefix=/usr/local/tools/gperftools LDFLAGS=-L/usr/local/tools/libunwind/lib CPPFLAGS=-L/usr/local/tools/libunwind/include  make  make install |

# 设定内存申请器

export LD\_PRELOAD=/usr/local/tools/gperftools/lib/libtcmalloc.so

# 指定内存分析结果存放路径

mkdir /tmp/gperftool-heap

export HEAPPROFILE=/tmp/gperftool-heap/heap

