### Dump Java堆栈信息

<https://www.jianshu.com/p/3f558809ab9e>

查看一个进程的内存占用每隔interval秒显示一次：

top -d ${interval} -p ${pid}

使用jmap查看各个对象占用内存情况，输出到live文件：

jmap -histo:live ${pid} > live

根据live文件中显示的占用内存最多的对象，分析其是由哪些类造成的。首先用jmap将堆内存dump下来到heap文件：

**jmap -dump:live,format=b,file=heap ${pid}**

使用jhat分析：**jhat heap**。jhat会启动一个7000端口的server，访问ip:7000可以查看堆内存情况。默认显示非平台的类（jdk的类不显示），在网页最下面可以查看平台的类，找到占用内存最大的那个类，点进去，最终找到是哪个类造成该类内存泄漏。

### 2. 找到对应的docker container

sudo docker ps | grep 'riihi-api'

**063c7811a6a5** docker-registry.sodacar.com:5000/sodacar/riihi-api:0.0.1-SNAPSHOT "/.r/r /opt/app/bi..." 18 hours ago Up 18 hours r-riihi-riihi-api-1-2fde341d

在host机器上

sudo docker cp **063c7811a6a5**:/opt/app/heap201802111027 .

sudo chmod 777 **heap201802111027**

### 3. 使用Jhat分析堆栈信息

<https://blog.gceasy.io/2015/08/28/jhat-heap-dump-analysis/>

可以查看，造成的原因是：

Redis类释放时，并不会去close jedisPool

<https://stackoverflow.com/questions/49553220/difference-in-committed-memory-and-rss-in-java-process>

<https://www.e-learn.cn/content/wangluowenzhang/185071>

## JVM NativeMemoryTracking 分析堆外内存泄露

<https://my.oschina.net/foxty/blog/1934968>

NMT必须先通过VM启动参数中打开，不过要注意的是，打开NMT会带来5%-10%的性能损耗。

-XX:NativeMemoryTracking=[off | summary | detail]

# off: 默认关闭

# summary: 只统计各个分类的内存使用情况.

# detail: Collect memory usage by individual call sites.

**jcmd查看NMT报告**

通过jcmd查看NMT报告以及查看对比情况。

jcmd <pid> VM.native\_memory [summary | detail | baseline | summary.diff | detail.diff | shutdown] [scale= KB | MB | GB]

# summary: 分类内存使用情况.

# detail: 详细内存使用情况，除了summary信息之外还包含了虚拟内存使用情况。

# baseline: 创建内存使用快照，方便和后面做对比

# summary.diff: 和上一次baseline的summary对比

# detail.diff: 和上一次baseline的detail对比

# shutdown: 关闭NMT

**VM退出时打印NMT**

可以通过下面VM参数在JVM退出时打印NMT报告。

-XX:+UnlockDiagnosticVMOptions -XX:+PrintNMTStatistics

可以在服务器中通过corn job来定期抓去NMT报告以及RSS和PMAP

COLLECTOR\_PID=`ps -ef|grep "ProcessName" | grep -v grep | awk '{print $2}'`

OUTDIR=/opt/chkmem

HOSTNAME=`hostname`

prstat -s rss 1 1 > ${OUTDIR}/${HOSTNAME}\_coll\_${COLLECTOR\_PID}\_prstat\_`date '+%Y%m%d\_%H%M%S'`.txt

/opt/jdk1.8.0\_40/bin/jcmd ${COLLECTOR\_PID} VM.native\_memory detail > ${OUTDIR}/${HOSTNAME}\_coll\_${COLLECTOR\_PID}\_nmd\_`date '+%Y%m%d\_%H%M%S'`.txt

pmap -x ${COLLECTOR\_PID} > ${OUTDIR}/${HOSTNAME}\_coll\_${COLLECTOR\_PID}\_pmap\_`date '+%Y%m%d\_%H%M%S'`.txt