## -XX:MaxPermSize = <N>

持久代大小通过 -XX:MaxPermSize = <N> 进行设置。

## 串行收集器

 用单线程处理所有垃圾回收工作，因为无需多线程交互，所有效率比较高。但是，也无法使用多处理器的优势，所以此收集器适合单处理器机器。当然，此收集器也可以用在小数据量（100M左右）情况下的多处理器机器上。

可以使用 -XX:+UseSerialGC打开

## 并行收集器

        对**年轻代**进行并行垃圾回收，因此可以减少垃圾回收时间。一般在多线程多处理器机器上使用。使用 -XX:+UseParallelGC 打开

默认**老年代**会是单线程回收。使用 -XX:+UseParallelOldGC打开对**老年代**的并行回收。

   使用 -XX:ParallelGCThreads = <N> 设置并行垃圾回收的线程数。此值可以设置与机器处理器数量相等

## -XX:NewRatio

表示年轻代（年轻代包括：Eden和两个Surivor）与年老代（年老代不包括持久代）的比值。

如果我们设置-XX:NewRatio=4 那么表示年轻代：年老代=1:4 也就是说：新生代是年老代的四分之一。

## -XX:SurvivorRatio

表示年轻代中,Survivor与Eden的比值。

如果我们设置：-XX：SurivivorRatio=3,那么Survivor:Eden=1:3 也就是说：Survivor是Eden的三分之一。

         那么在整个年轻代中，Surivivor占多少比例呢？答案是：占五分之二。

        因为年轻代中有两个Survivor和Eden

       Survivor:Eden=1:3，所以,Surivivor占年轻代的2/5,Eden占3/5。

 Survivor=500 \* 2/5=200m                       Eden=500 \* 3/5=300m

## 查看jvm中对象的大小

命令：jmap -histo:live pid  
描述：显示堆中对象的统计信息

其中包括每个Java类、对象数量、内存大小(单位：字节)、完全限定的类名。打印的虚拟机内部的类名称将会带有一个’\*’前缀。如果指定了live子选项，则只计算活动的对象。

