一、

-Xint, -Xcomp, -Xmixed

-Xint 强制全部使用字节码解释运行，-Xcomp全部编译成机器码执行，-Xmixed是混合模式，jdk默认的启动参数。

二、

1.jvm参数分类

Jvm参数分为三类。第一类包括了标准参数，java –help即可显示；第二类是X参数，可以用java –X来检索；第三类是XX参数，所有的XX都以“-XX:”开始，随后语法取决于参数类型：

1）对于布尔类型的参数，用“+”或者“-”，-XX:+<name>用于激活<name>选项，-XX:-<name>用于注销<name>选项。

2）对于非布尔类型的参数，需要赋值，-XX:<name>=<value>

-XX:+PrintCompilation 可以简单输出一些字节码转化成本地代码的编译过程。

-XX:+CITime 输出编译时间。

-XX:+CITime和-XX:+PrintCompilation一起使用可以对JIT编译器有更好的理解，如果使用-Xint，那么编译时间显示是0，因为是字节码解释运行，不需要编译。

2.jvm参数调优

-XX:+PrintFlagsFinal -XX:+PrintFlagsInitial 可以输出所有XX参数和值。

-XX:+PrintCommandLineFlags 打印出所有用户或者jvm默认设置过的XX参数和值。

-Xms和-Xmx（等同于-XX:InitialHeapSize and -XX:MaxHeapSize），默认和最大堆内存大小。

-XX:+HeapDumpOnOutOfMemoryError 让JVM在发生内存溢出时候自动生成堆内存快照，默认快照的存放路径在JVM启动目录下java\_pid<pid>.hprof文件里（<pid>指JVM进程的进程号），也可以通过设置-XX:HeapDumpPath=<path>改变默认堆内存快照生成路径。

-XX:OnOutOfMemoryError 利用该命令，可以在发生内存溢出时，执行一些清理工作等。

|  |
| --- |
| $ java -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp/heapdump.hprof -XX:OnOutOfMemoryError ="sh ~/cleanup.sh" MyApp  该例子表示当发生内存溢出时，堆内存快照写到/tmp/heapdump.hprof路径下， |

-XX:PermSize and -XX:MaxPermSize 设置永久代的初始大小和最大值。注意这里设置的永久代不会被包括在使用参数-Xmx（-XX:MaxHeapSize）设置的堆内存大小中。

-XX:InitialCodeCacheSize and -XX:ReservedCodeCacheSize，用来存储已编译方法生成的本地代码，如果被占满，那么JIT编译器将被停用。

-XX:NewSize，-XX:MaxNewSize 设置新生代的大小

-XX:NewRatio，设置老年代和新生代比例，优点是新生代会随着整个堆大小动态扩展。

|  |
| --- |
| $ java -XX:NewSize=32m -XX:MaxNewSize=512m -XX:NewRatio=3 MyApp  JVM会尝试为新生代分配1/4的堆空间，但是不会小于32M，也不会大于512M |

-XX:SurvivorRatio，伊甸园区和幸存区大小比例。比如-XX:SurvivorRatio=10，表示一点远去是幸存区to的比例是10，所以eden：survivor1：survivor2 = 10：1：1。

-XX:+PrintTenuringDistribution，执行JVM在每次新生代GC时，输出幸存区中对象的年龄分布。

三、吞吐量收集器

评估一个垃圾收集算法的两个标准：

1.吞吐量越高算法越好

2.暂停时间越短算法越好