jvm运行时的数据区域分为

1。线程私有的：

程序计数器

虚拟机栈：(-Xss **-Xss128k**：设置每个线程的堆栈大小 减小这个值能生成更多的线程 但是操作系统对一个进程内的线程数还是有限制的，不能无限生成，经验值在3000~5000左右。)

栈帧（当前栈顶栈帧才有效）

局部变量表（参数、方法内部局部变量，0th slot代表this，其次参数，然后局部变量）

操作数栈

动态链接

方法出口

正常完成出口

异常完成出口

本地方法栈

2.线程共享的：**JVM内存大小=年轻代大小 + 年老代大小 + 持久代大小**

堆（-Xmx **-Xmx3550m**：设置JVM最大可用内存为3550M默认(MaxHeapFreeRatio参数可以调整)空余堆内存大于70%时，JVM会减少堆直到 -Xms的最小限制 -Xms 设置JVM初始内存为3550m。默认(MinHeapFreeRatio参数可以调整)空余堆内存小于40%时，JVM就会增大堆直到-Xmx的最大限制）

年轻代(Minor GC)(-Xmn(**-Xmn2g**：设置年轻代大小为2G 增大年轻代后,将会减小年老代大小 推荐配置为整个堆的3/8) -XX:NewRatio-老年: 年轻) 若为4 则老年占4/5 年轻 1/5 Xms=Xmx并且设置了Xmn的情况下，该参数不需要进行设置。

e(-XX:SurvivorRatio e:s0) 若为4 则2个s占 2/6， e占4/6默认比例为8：1

s0

s1

老年代(Major GC)

方法区（永久代）(-XX:MaxPermSize -XX:MinPermSize)

运行时常量池

静态变量

常量

即时编译后的代码

直接内存

args:

-verbose:class

-XX:+PrintGCDetails

-XX:+PrintGCTimeStamps

-XX:+PrintHeapAtGC

-XX:+TraceClassLoading

-XX:+TraceClassUnLoading

GC

回收

算法

引用计数法

根搜索算法

区域

方法区

废弃常量

无用类

堆

年轻代

复制

老年代

标记-整理

收集器

年轻代

Serial（单线程）

ParNew（多线程版Serial）

Parallel Scanvenge（控制吞吐量）(GCTimeRatio,MaxGCPauseMillis)

老年代

CMS(Concurrent Mark Sweep)（最短停顿，Concurrent Mode Failure->Full GC,HandPromotionFailure->Full GC）(CMSInitiatingOccupancyFraction,UseCMSCompactAtFullCollection,CMSFullGCsBeforeCompaction)

初始标记（STW，快）

并发标记（很慢）

重新标记（STW，较慢）

并发清除（慢）

Serial Old（单线程）

Parallel Old（老年版Parallel Scanvenge）

G1（Garbage First,jdk1.7，标记-整理，无空间碎片，精准控制停顿）

指定GC

-XX:UseSerialGC Serial+Serial (Old client default)

-XX:UseParNewGC ParNew+Serial Old

-XX:UseConMarkSweepGC ParNew+CMS+SerialOld

-XX:UseParallelGC Parallel Scanvenge+Serial Old(Server Default)

-XX:UseParallelOldGC Parallel Scanvenge+Parallel Old

通用

ParallelGCThreads

SurvivorRatio

PretenureSizeThreshold

MaxTenuringThreshold

UseAdaptiveSizePolicy

HandlePromotionFailure

类加载

时机

main

父类先初始化

new getstatic putstatic invokestatic

反射

过程

加载

类全名->class字节流（类加载器）

字节流->方法区运行时存储结构

堆中产生Class对象（as方法区入口）

验证

字节流符合虚拟机要求

准备

赋予变量初始值

解析

常量池符号引用->直接引用

时机：

anewarray，checkcast，getfield，getstatic，instanceof，invokeinterface，invokespecial，invokestatic，invokevirtual，multianewarray，new，putfield，putstatic

类别：

类或接口

加载类

字段

解析字段所属类

查找字段引用（本身，接口及父接口，父类，异常）

类方法

解析方法所属类

查找方法引用（本身，父类，接口及父接口，异常）

接口方法

解析方法所属类

查找方法引用（本身，父接口，异常）

初始化

<clinit>执行过程，父类的先执行（接口的不是，父接口变量被使用才会先初始化父接口），<clinit>不是必须生成的，执行过程线程安全的。

使用

卸载

类加载器

Bootstrap lib -Xbootclasspath=

Extension lib\ext -Djava.ext.dirs=

Application

应用：

热部署-OSGi

虚拟机执行

解释执行

编译执行

混合多级

方法调用

解析（编译器能确定调用版本，符号引用->直接引用）（能被invokestatic invokespecial调用的方法）（非虚方法，包含final修饰方法）（静态过程）

静态方法

私有方法

实例构造器

父类方法

分派

静态分派（重载时根据静态类型确定版本）

java静态多分派

动态分派

java动态单分派

方法区（cache）（分派调用的稳定优化手段）

建立虚方法表

建立接口方法表

查看gc 每秒刷新

jps 看进程id

jinfo 看进程属性 vm参数之类

jstat -gcutil 6392 1000