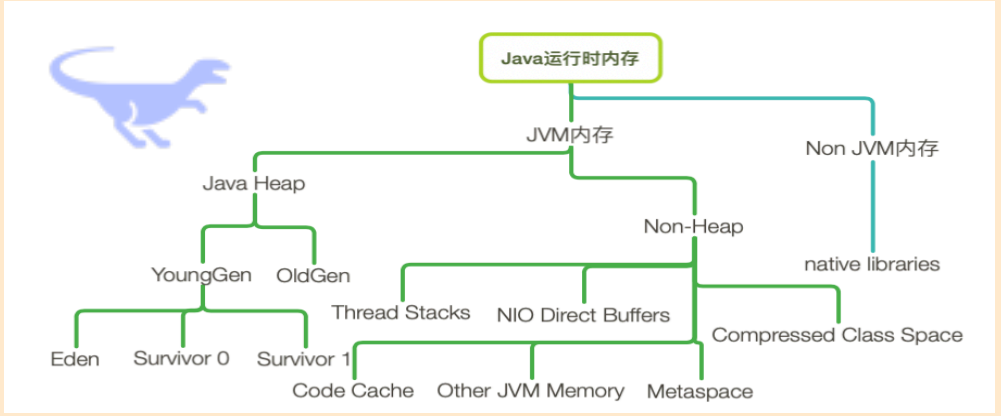




HotSpot JVM可用GC类型		
年轻代回收算法	老年代回收算法	GC启用参数
Serial(DefNew))	Serial Mark Sweep Compact(PSOldGen)	-XX:+UseSerialGC
Parallel scavenge(PSYoungGen)	Serial Mark Sweep Compact(PSOldGen)	-XX:+UseParallelGC
Parallel scavenge(PSYoungGen))	Parallel Mark Sweep Compact(ParOldGen)	-XX:+UseParallelOldGC(随-XX:+UseParallelGC开启)
G1(Garbage First)		-XX:+UseG1GC(默认)
ZGC(Z garbage collector)		-XX:+UnlockExperimentalVMOptions -XX:+UseZGC



Application Class Data Sharing配置	
-XX:+UnlockCommercialFeatures	解锁商业功能
-Xshare:mode	设置CDS的mode(auto,on,off)
-XX:DumpLoadedClassList	dump加载的class list
-XX:SharedClassListFile	需要共享的class list
-XX:SharedArchiveFile	需要创建的shared archive文件

VM通用参数	
VM参数	说明
-Xrs	减少JVM对操作系统信号的捕获，用户自行处理接收到的信号
-XX:-CompactStrings -XX:+UseLargePages -XX:+UseHugeTLBFS	取消JDK9中的字符串压缩功能 开启largepage
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=path	OutOfMemoryError异常时，dump heap信息

JIT调优	
VM参数	说明
-XX:ReservedCodeCacheSize=size -XX:InitialCodeCacheSize=size	JIT最大的code cache size 初始化的codeCacheSize
-XX:AllocateInstancePrefetchLines=lines -XX:AllocatePrefetchDistance=size -XX:AllocatePrefetchInstr=instruction	设置要在实例分配指针之前预取的行数。默认值为1 设置对象分配的预取距离的大小（以字节为单位） 设置prefetch指令，值为0-3
-XX:AllocatePrefetchLines=1	使用JIT编译代码生成的预取指令在最后一个对象分配后加载的缓存行数。
-XX:AllocatePrefetchStyle=1	生成预取指令的代码样式：0,没有预取指令生成；1,在每次分配之后执行预取指令；2,当预取指令执行的时候，使用TLAB分配水印指针
-XX:AllocatePrefetchStepSize=size -XX:+BackgroundCompilation -XX:CICompilerCount=threads -XX:CompileCommand=command,method[,option] -XX:CompileCommandFile=filename -XX:CompileOnly=methods -XX:CompileThreshold=invocations -XX:CompileThresholdScaling=scale	预取指令的step size 开启后台编译 设置编译线程的个数 指定具体方法的编译行为 从文件指定具体方法的编译行为 只有指定的方法才会被编译 设置在编译之前需要被解释执行的次数 CompileThreshold的比例，小于1表示提前编译，大于1表示延后编译
-XX:+UseRTMLocking -XX:RTMAbortRatio=abort_ratio	开启RTM Locking RTM transactions中断的比例，超出该比例代码会被反优化

JIT调优	
VM参数	说明
-XX:+DoEscapeAnalysis -XX:+Inline -XX:InlineSmallCode=size -XX:MaxInlineSize=size -XX:MaxNodeLimit=nodes -XX:NonNMethodCodeHeapSize=size -XX:NonProfiledCodeHeapSize=size -XX:ProfiledCodeHeapSize=size -XX:MaxTrivialSize=size -XX:+OptimizeStringConcat -XX:+SegmentedCodeCache -XX:-TieredCompilation -XX:+UseCondCardMark -XX:+UseCountedLoopSafepoints -XX:+LogCompilation -XX:+PrintAssembly -XX:+PrintCompilation -XX:+PrintInlining -XX:+UseRTMDeopt -XX:RTMRetryCount=count	开启逃逸分析 开启inline方法 设置需要inline方法的最大大小 设置最大inline大小 单方法编译的最大节点个数 code segment中非方法code的大小 code segment中nonprofiled methods的大小 code segment中profiled methods的大小 普通方法内联的最大大小 String连接优化 code cache分区 取消分层编译 更新card table之前检查card是否已经被标记过 将safe point保持在计数循环中。 输出编译日志 打印汇编代码 输出编译的方法信息 输出inlining信息 开启RTM代码的反优化 RTM locking code重试的次数

Thread配置	
VM参数	说明
-XX:TLABSize=size -XX:+UseTLAB -XX:+ResizeTLAB -XX:MinTLABSize=64k	TLAB的初始大小 开启TLAB 允许JVM对TLAB进行调整 最小TLAB大小

G1调优参数	
VM参数	说明
-XX:G1HeapRegionSize=32m -XX:G1ReservePercent=10 -XX:G1MixedGCCountTarget=8 -XX:G1MixedGCLiveThresholdPercent=percent -XX:G1HeapWastePercent=10 -XX:MaxGCPauseMillis=500 -XX:G1MaxNewSizePercent=percent -XX:G1NewSizePercent=percent	heap区域的大小 设置预留空闲内存百分比，以降低内存溢出的风险 一个混合收集周期中包含多少次混合收集 old region被包含在一个混合收集周期的阈值 设置浪费的堆内存百分比，当可回收百分比小于浪费百分比时，JVM就不会启动混合垃圾收 GC的最大暂停时间 young gen占用heap的最大比例 young gen占用heap的最小比例

内存大小调整	
VM参数	说明
-XX:ReservedCodeCacheSize=512m -XX:MaxDirectMemorySize=2g -XX:MaxRAM=size -XX:InitialRAMPercentage -XX:MaxRAMPercentage -XX:MinRAMPercentage	codeCache的最大大小 NIO direct buffer的最大值 RAM相关参数

GC日志详情	
VM参数	说明
-verbose:gc -Xlog:gc -Xlog:gc* -Xlog:task*=debug -Xlog:gc+heap=trace -Xlog:age*=level -Xlog:ref*=debug -Xlog:ergo*=level -XX:+PrintPromotionFailure -Xlog:safepoint -Xlog:gc+region=trace	打印基本的GC信息 打印详细的GC信息 输出GC work thread task的timestamps GC的heap信息 young gen的age信息 STW阶段打印reference processing 输出自适应的分代大小 输出promotion失败的信息 是应用程序在不停止的情况下工作的时间,即两个连续安全点之间的时间 输出G1 region分配和回收信息

GC日志输出	
VM参数	说明
-Xlog:gc:garbage-collection.log -XX:ErrorFile=filename -XX:LogFile=path	将GC日志输出到文件 GC错误日志重定向 JVM日志重定向

其他GC日志信息	
-XX:+PrintTLAB -XX:+PrintPLAB -XX:+PrintOldPLAB -Xlog:stringdedup*=debug -XX:+PrintHeapAtSIGBREAK -XX:+PrintClassHistogramAfterFullGC -XX:+PrintClassHistogramBeforeFullGC -XX:+PrintClassHistogram	输出TLAB信息 输出PLAB信息 输出old space的PLAB信息 输出String去重的信息 接收到signal的时候的heap信息 full GC后的class直方图信息 full GC之前的class直方图信息 在收到Ctl+B或Ctl+C信号时输出class的直方图信息

通用GC参数	
VM参数	说明
-Xnoclassgc -XX:ActiveProcessorCount=x -XX:+AggressiveHeap -XX:+AlwaysPreTouch -XX:InitiatingHeapOccupancyPercent=percent -XX:MaxGCPauseMillis=time -XX:MaxHeapFreeRatio=percent -XX:MinHeapFreeRatio=percent -XX:TargetSurvivorRatio=percent -XX:+ScavengeBeforeFullGC -XX:+UseStringDeduplication -XX:StringDeduplicationAgeThreshold=threshold -XX:+DisableExplicitGC	禁用classes的GC 重写VM使用的CPU核数 开启java heap优化 在main方法执行之前将所有的page都加载到heap中 触发GC的heap使用比例 最大的GC暂停时间 GC之后最大的Heap释放比例 GC之后最小的Heap释放比例 young GC之后Survivor的目标比例 在fullGC之前运行youngGC 开启字符串去重 字符串去重的最小Age数 禁止显式调用System.gc()

GC并发线程控制	
VM参数	说明
-XX:ParallelGCThreads=threads -XX:+ParallelRefProcEnabled -XX:+UseGCOverheadLimit -XX:ConcGCThreads = n	设置STW的垃圾收集线程数 开启并发reference processing OutOfMemoryError之前JVM在GC上使用的时间比例 设置并行标记线程的数量

Young space tenuring	
-XX:InitialTenuringThreshold=8 -XX:MaxTenuringThreshold=15 -XX:PretenureSizeThreshold=2m -XX:+AlwaysTenure -XX:+NeverTenure	设置保有年龄阈值,就是一个对象多少age之后会被升级到old space 最大的保有年龄阈值 超出该阈值，对象将会被直接分配到old space 将young space中的survivor对象直接提升到old space 除非survivor space不能容纳该对象了，否则不会提升到old space

FlightRecorder	
-XX:FlightRecorderOptions	
globalbuffersize=size maxchunksize=size memorysize=size numglobalbuffers old-object-queue-size= repository=path retransform={true false} samplethreads={true false} stackdepth=depth threadbuffersize=size	全局buffer的大小,全局buffer可以有多个 一个data chunk记录的最大值 总的内存大小=globalbuffersize * numglobalbuffers 全局buffer的个数 保存的最大objects的个数 磁盘上面的存储地址 event class是否需要使用JVM TI做转换 是否支持取样 栈的深度 指定每个线程的本地buffer大小

FlightRecorder	
-XX:StartFlightRecording	
delay=time disk={true false} dumponexit={true false} duration=time filename=path maxage=time maxsize=size path-to-gc-roots={true false} settings=path	java应用程序开启之后多久开始记录 记录的时候是否写到磁盘 JVM shutdown的时候是否dumped正在运行的record 记录持续时间 recording停止时候记录写入的文件 磁盘数据保留时间 磁盘数据的最大值 是否收集GC root的信息 event的配置文件，默认使用的是JRE_HOME/lib/jfr/default.jfc