



JVM参数使用手册

jvm 阅读约 14 分钟

JVM提供了大量的参数配置，可以通过配置这些参数对JVM进行调优、记录GC日志等等，本章只罗列我自己用到过和了解过的参数，根据功能划分。

声明:英文解释和使用方法多数来源于微信小程序**JVMPocket**，JVM交流QQ群号:462563010

内存相关

通过这些参数可以对JVM的内存分配做调整

Xms

英文解释: Initial heap size(in bytes)
中文释义: 堆区初始值
使用方法: -Xms2g 或 -XX:InitialHeapSize=2048m

Xmx

英文解释: Maximum heap size(in bytes)
中文释义: 堆区最大值
使用方法: -Xmx2g 或 -XX:MaxHeapSize=2048m

Xmn

英文解释: Maximum new generation size(in bytes)
中文释义: 新生代最大值
使用方法: -Xmn512m 或 -XX:MaxNewSize=512m

PermSize

英文解释: Initial size of permanent generation(in bytes)
中文释义: 永久代初始值
使用方法: -XX:PermSize=128m

MaxPermSize

英文解释: Maximum size of permanent generation(in bytes)
中文释义: 永久代最大值
使用方法: -XX:MaxPermSize=256m

Xss

英文解释: Thread Stack Size(in Kbytes)
中文释义: 线程栈最大值
使用方法: -Xss256k 或 -XX:ThreadStackSize=256k

GC策略相关

通过这些参数可以对JVM的GC性能进行调优

SurvivorRatio

英文解释: Rato of eden/survivor space size
中文释义: eden区和survivor的比值

使用方法：-XX:SurvivorRatio=6

使用经验：假如设为6，则表示每个survivor区跟eden区的比值为1:6,每个survivor区占新生代的八分之一

PretenureSizeThreshold

英文解释：Maximum size in bytes of objects allocated in DefNew generation;zero means no maximum

中文释义：可以在新生代直接分配的对象最大值，0表示没有最大值

使用方法：-XX:PretenureSizeThreshold=1000000

使用经验：设置该参数，可以使大于这个值的对象直接在老年代分配，避免在Eden区和Survivor区发生大量的内存复制，该参数只对Serial和ParNew收集器有效，Parallel Scavenge并不认识该参数

MaxTenuringThreshold

英文解释：Maximum value fo tenuring threshold

中文释义：年轻代最大年龄

使用方法：-XX:MaxTenuringThreshold=10

使用经验：每个对象在坚持过一次Minor GC之后，年龄就增加1，当超过这个参数值时就进入老年代，最大支持15

UseSerialGC

英文解释：Use the Serial garbage collector

中文释义：年轻代使用Serial垃圾收集器

使用方法：

开启 -XX:+UseSerialGC

关闭 -XX:-UseSerialGC

使用经验：不推荐使用，性能太差，老年代将会使用SerialOld垃圾收集器

UseParNewGC

英文解释：Use parallel threads in the new generation

中文释义：年轻代使用ParNew垃圾收集器

使用方法：

开启 -XX:+UseParNewGC

关闭 -XX:-UseParNewGC

ParallelGCThreads

英文解释：Number of parallel threads parallel gc will use

中文释义：并行执行gc的线程数

使用方法：-XX:ParallelGCThreads=16

UseParallelGC

英文解释：Use the Parallel Scavenge garbage collector

中文释义：年轻代使用Parallel Scavenge垃圾收集器

使用方法：

开启 -XX:+UseParallelGC

关闭 -XX:-UseParallelGC

使用经验：Linux下1.6,1.7,1.8默认开启，老年代将会使用SerialOld垃圾收集器

UseParallelOldGC

英文解释：Use the Parallel Old garbage collector

中文释义：年轻代使用Parallel Scavenge收集器

使用方法：

开启 -XX:+UseParallelOldGC

关闭 -XX:-UseParallelOldGC

使用经验：老年代将会使用Parallel Old收集器

UseConcMarkSweepGC

英文解释：Use Concurrent Mark-Sweep GC in the old generation

中文释义：老年代使用CMS收集器（如果出现"Concurrent Mode Failure"，会使用SerialOld收集器）

使用方法：

开启 `-XX:+UseConcMarkSweepGC`

关闭 `-XX:-UseConcMarkSweepGC`

使用经验：年轻代将会使用ParNew收集器

CMSInitiatingOccupancyFraction

英文解释：Percentage CMS generation occupancy to start a CMS collection cycle. A negative value means that `CMSTriggerRatio` is used

中文释义：触发执行CMS回收的当前年代区内存占用的百分比，负值表示使用`CMSTriggerRatio`设置的值

使用方法：`-XX:+CMSInitiatingOccupancyFraction=75`

UseCMSInitiatingOccupancyOnly

英文解释：Only use occupancy as a criterion for starting a CMS collection

中文释义：只根据占用情况作为开始执行CMS收集的标准

使用方法：

开启 `-XX:+UseCMSInitiatingOccupancyOnly`

关闭 `-XX:-UseCMSInitiatingOccupancyOnly`

UseCMSCompactAtFullCollection

英文解释：Use Mark-Sweep-Compact algorithm at full collections

中文释义：使用CMS执行Full GC时对内存进行压缩

使用方法：

开启 `-XX:+UseCMSCompactAtFullCollection`

关闭 `-XX:-UseCMSCompactAtFullCollection`

CMSFullGCsBeforeCompaction

英文解释：Number of CMS full collection done before compaction if > 0

中文释义：多少次FGC后进行内存压缩

使用方法：`-XX:CMSFullGCsBeforeCompaction=1`

CMSClassUnloadingEnabled

英文解释：Whether class unloading enabled when using CMS GC

中文释义：当使用CMS GC时是否启用类卸载功能

使用方法：

开启 `-XX:+CMSClassUnloadingEnabled`

关闭 `-XX:-CMSClassUnloadingEnabled`

CMSParallelRemarkEnabled

英文解释：Whether parallel remark enabled (only if ParNewGC)

中文释义：是否启用并行标记（仅限于ParNewGC）

使用方法：

开启 `-XX:+CMSParallelRemarkEnabled`

关闭 `-XX:-CMSParallelRemarkEnabled`

UseG1GC

英文解释：Use the Garbage-First garbage collector

中文释义：使用G1垃圾收集器

使用方法：

开启 `-XX:+UseG1GC`

关闭 `-XX:-UseG1GC`

MaxGCPauseMillis

英文解释：Adaptive size policy maximum GC pause time goal in millisecond, or (G1 Only) the maximum GC time per MMU time slice

中文释义：自适应大小策略的最大GC暂停时间目标（以毫秒为单位），或（仅G1）每个MMU时间片的最大GC时间

使用方法：`-XX:MaxGCPauseMillis=200`

DisableExplicitGC

英文解释: Ignore calls to System.gc()

中文释义: 禁用System.gc()触发FullGC

使用方法:

开启 -XX:+DisableExplicitGC

关闭 -XX:-DisableExplicitGC

PS:不建议开启, 如果开启了这个参数可能会导致对外内存无法及时回收造成对外内存溢出

GC日志相关

通过这些参数可以对JVM的GC日志输出进行配置, 方便分析

Xloggc

英文解释: GC log file

中文释义: GC日志文件路径

使用方法: -Xloggc:/data/gclog/gc.log

UseGCLogFileRotation

英文解释: Rotate gclog files(for long running applications). It requires -Xloggc:<filename>

中文释义: 滚动GC日志文件, 须配置Xloggc

使用方法:

开启 -XX:+UseGCLogFileRotation

关闭 -XX:-UseGCLogFileRotation

NumberOfGCLogFiles

英文解释: Number of gclog files in rotation(default:0,no rotation)

中文释义: 滚动GC日志文件数, 默认0, 不滚动

使用方法: -XX:NumberOfGCLogFiles=4

GCLogFileSize

英文解释: GC log file size,requires UseGCLogFileRotation. Set to 0 to only trigger rotation via jcmd

中文释义: GC文件滚动大小, 需配置UseGCLogFileRotation, 设置为0表示仅通过jcmd命令触发

使用方法: -XX:GCLogFileSize=100k

PrintGCDetails

英文解释: Print more details at garbage collection

中文释义: GC时打印更多详细信息

使用方法:

开启 -XX:+PrintGCDetails

关闭 -XX:-PrintGCDetails

可以通过jinfo -flag [+|-]PrintGCDetails <pid> 或 jinfo -flag PrintGCDetails=<value> <pid> 来动态开启或设置值

PrintGCDateStamps

英文解释: Print date stamps at garbage collection

中文释义: GC时打印时间戳信息

使用方法:

开启 -XX:+PrintGCDateStamps

关闭 -XX:-PrintGCDateStamps

可以通过jinfo -flag [+|-]PrintGCDateStamps <pid> 或 jinfo -flag PrintGCDateStamps=<value> <pid> 来动态开启或设置值

PrintTenuringDistribution

英文解释: Print tenuring age information

中文释义: 打印存活实例年龄信息

使用方法：

开启 `-XX:+PrintTenuringDistribution`

关闭 `-XX:-PrintTenuringDistribution`

PrintGCApplicationStoppedTime

英文解释：Print the time of application has been stopped

中文释义：打印应用暂停时间

使用方法：

开启 `-XX:+PrintGCApplicationStoppedTime`

关闭 `-XX:-PrintGCApplicationStoppedTime`

PrintHeapAtGC

英文解释：Print heap layout before and after each GC

中文释义：GC前后打印堆区使用信息

使用方法：

开启 `-XX:+PrintHeapAtGC`

关闭 `-XX:-PrintHeapAtGC`

异常相关

通过这些参数可以在JVM异常情况下执行某些操作，以保留现场做分析用

HeapDumpOnOutOfMemoryError

英文解释：Dump heap to file when java.lang.OutOfMemoryError is thrown

中文释义：抛出内存溢出错误时导出堆信息到指定文件

使用方法：

开启 `-XX:+HeapDumpOnOutOfMemoryError`

关闭 `-XX:-HeapDumpOnOutOfMemoryError`

可以通过`jinfo -flag [+|-]HeapDumpOnOutOfMemoryError <pid>` 或 `jinfo -flag HeapDumpOnOutOfMemoryError=<value> <pid>` 来动态开启或设置值

HeapDumpPath

英文解释：When HeapDumpOnOutOfMemoryError is on, the path(filename or directory) of the dump file(defaults to java_pid<pid>.hprof in the working directory)

中文释义：当HeapDumpOnOutOfMemoryError开启的时候，dump文件的保存路径，默认为工作目录下的java_pid<pid>.hprof文件

使用方法： `-XX:HeapDumpPath=/data/dump/jvm.dump`

其他

server

英文解释：server mode

中文释义：服务端模式

使用方法： `-server`

TieredCompilation

英文解释：Enable tiered compilation

中文释义：启用多层编译

使用方法：

开启 `-XX:+TieredCompilation`

关闭 `-XX:-TieredCompilation`

阅读 5.6k •更新于 2019-12-01



赞 7



收藏 46



赞赏



分享

本作品系 原创 ， 采用《署名-非商业性使用-禁止演绎 4.0 国际》许可协议



MartinDai

1.4k



关注作者

1 条评论

得票 • 时间



撰写评论 ...

提交评论



SeanDragon： 好东西，感谢分享。

回复 • 2017-08-10

推荐阅读

OpenJDK9 Hotspot：Zero 解释器 - BytecodeInterpreter

本文简要介绍hotspotzero解释器中BytecodeInterpreter类解释执行字节码过程，用到的相关的概念可以参考之前关于zero解释器相...
xingpingz • 阅读 31

JVM的编译策略

jvm系列垃圾回收基础JVM的编译策略GC的三大基础算法GC的三大高级算法GC策略的评价指标JVM信息查看GC通用日志解读jvm的...
codecraft • 阅读 382

JVM内存模型解析

JVM内存模型可以分为两个部分，如下图所示，堆和方法区是所有线程共有的，而虚拟机栈，本地方法栈和程序计数器则是线程私...
ziwenxie • 阅读 30

JVM系列-01-JVM内存模型

TOC本篇文章是本人阅读《深入理解JVM》和《java虚拟机规范》时的笔记。JVM是HotSpot，jdk1.7。大神绕路，不喜勿喷。每个...
诸葛流云 • 阅读 22

jvm参数陷阱

本文主要记录一些jvm参数的使用陷阱。-XX:MaxTenuringThreshold只对串行回收器和ParNew有效，对ParallGC无效。存活次数在...
codecraft • 阅读 13

分析JVM GC及内存情况的方法

当JVM响应变慢或者停滞的时候，我们往往需要对GC和其内存情况是进行分析，下面列举一些常用的分析方法和工具：详细解释可...
chanjarster • 阅读 16

译文-Hotspot VM中Oop Maps的含义？

原文出处：WhatdoesOopMapsmeansinHotspotVMexactlyOopMaps是记录Java栈中引用对象的数据结构。它的主要作用是发现Jav...
子瞻 • 阅读 3

jvm参数备忘

本文主要记录一些jvm参数。-XX:+PrintCommandLineFlags这个参数的作用是显示出VM初始化完毕后所有跟最初的默认值不同的...
codecraft • 阅读 17

Martin的专栏

用户专栏

工作经验小结

15 人关注 24 篇文章

- 关注专栏
- 专栏主页

产品

- 热门问答
- 热门专栏
- 热门课程
- 最新活动
- 技术圈
- 酷工作
- 移动客户端

课程

- Java 开发课程
- PHP 开发课程
- Python 开发课程
- 前端开发课程
- 移动开发课程

资源

- 每周精选
- 用户排行榜
- 徽章
- 帮助中心
- 声望与权限
- 社区服务中心

合作

- 关于我们
- 广告投放
- 职位发布
- 讲师招募
- 联系我们
- 合作伙伴

关注

- 产品技术日志
- 社区运营日志
- 市场运营日志
- 团队日志
- 社区访谈

条款

- 服务条款
- 隐私政策
- 
- 下载 App