



链滴

JVM 命令行参数解析

作者: [james](#)

原文链接: <https://ld246.com/article/1513694373406>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1. java 命令行参数

先看 java 命令行的参数


```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">solr@2f1fe8cc9f09:/opt/solr/server/solr-webapp/webapp$ java
</span></span><span class="highlight-line"><span class="highlight-cl">Usage: java [-opti
ns] class [args...]
</span></span><span class="highlight-line"><span class="highlight-cl">           (to execute
a class)
</span></span><span class="highlight-line"><span class="highlight-cl"> or java [-option
] -jar jarfile [args...]
</span></span><span class="highlight-line"><span class="highlight-cl">           (to execute
a jar file)
</span></span><span class="highlight-line"><span class="highlight-cl">where options inc
lude:
</span></span><span class="highlight-line"><span class="highlight-cl"> -d32    use a 3
-bit data model if available
</span></span><span class="highlight-line"><span class="highlight-cl"> -d64    use a 6
-bit data model if available
</span></span><span class="highlight-line"><span class="highlight-cl"> -server  to se
lect the "server" VM
</span></span><span class="highlight-line"><span class="highlight-cl"> -zero    to sele
ct the "zero" VM
</span></span><span class="highlight-line"><span class="highlight-cl"> -dcevm   to se
lect the "dcevm" VM
</span></span><span class="highlight-line"><span class="highlight-cl">           The de
fault VM is server,
</span></span><span class="highlight-line"><span class="highlight-cl">           becaus
you are running on a server-class machine.
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> -cp &lt;class se
rch path of directories and zip/jar files&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> -classpath &lt;c
lass search path of directories and zip/jar files&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">           A : sep
rated list of directories, JAR archives,
</span></span><span class="highlight-line"><span class="highlight-cl">           and ZIP
archives to search for class files.
</span></span><span class="highlight-line"><span class="highlight-cl"> -D&lt;name&gt;
=&lt;value&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">           set a s
tem property
</span></span><span class="highlight-line"><span class="highlight-cl"> -verbose:[class|
c|jni]
</span></span><span class="highlight-line"><span class="highlight-cl">           enable
verbose output
</span></span><span class="highlight-line"><span class="highlight-cl"> -version  prin
t product version and exit
</span></span><span class="highlight-line"><span class="highlight-cl"> -version:&lt;val
e&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">           Warni
g: this feature is deprecated and will be removed
</span></span></code>
```

 in a fu
ure release.
 require
the specified version to run
 -showversion p
rint product version and continue
 -jre-restrict-sea
ch | -no-jre-restrict-search
 Warni
g: this feature is deprecated and will be removed
 in a fu
ure release.
 include
exclude user private JREs in the version search
 -? -help print
this help message
 -X print
elp on non-standard options
 -ea[:<packag
name>...]:<classname>]
 -enableassertio
s[:<packagename>...]:<classname>]
 enable
assertions with specified granularity
 -da[:<packag
name>...]:<classname>]
 -disableasserti
ns[:<packagename>...]:<classname>]
 disable
assertions with specified granularity
 -esa | -enablesy
temassertions
 enable
ystem assertions
 -dsa | -disables
stemassertions
 disable
ystem assertions
 -agentlib:<li
name>[=<options>]
 load na
ive agent library <libname>, e.g. -agentlib:hprof
 see als
, -agentlib:jwp=help and -agentlib:hprof=help
 -agentpath:<
athname>[=<options>]
 load na
ive agent library by full pathname
 -javaagent:<j
rpath>[=<options>]
 load Ja
a programming language agent, see java.lang.instrument
 -splash:<ima
epath>

```
</span></span><span class="highlight-line"><span class="highlight-cl"> show s
lash screen with specified image
</span></span><span class="highlight-line"><span class="highlight-cl">See http://www.or
cle.com/technetwork/java/javase/documentation/index.html for more details.
</span></span><span class="highlight-line"><span class="highlight-cl">solr@2f1fe8cc9f09
/opt/solr/server/solr-webapp/webapp$ java -version
</span></span><span class="highlight-line"><span class="highlight-cl">openjdk version "1
8.0_141"
</span></span><span class="highlight-line"><span class="highlight-cl">OpenJDK Runtime
Environment (build 1.8.0_141-8u141-b15-1~deb9u1-b15)
</span></span><span class="highlight-line"><span class="highlight-cl">OpenJDK 64-Bit S
rver VM (build 25.141-b15, mixed mode)
</span></span><span class="highlight-line"><span class="highlight-cl">solr@2f1fe8cc9f09
/opt/solr/server/solr-webapp/webapp$
</span></span></code></pre>
```


看一个实际运行的 java 命令:


```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">solr@2f1fe8cc9f09:/opt/solr/server/solr-webapp/webapp$ ps aux | grep java
</span></span><span class="highlight-line"><span class="highlight-cl">solr      1  0.2 13
3 3060500 272904 ?    Ssl  Oct08  44:03 /docker-java-home/jre/bin/java -server -Xms512m -
mx512m -XX:NewRatio=3 -XX:SurvivorRatio=4 -XX:TargetSurvivorRatio=90 -XX:MaxTenuring
hreshold=8 -XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:ConcGCThreads=4 -XX:Par
allelGCThreads=4 -XX:+CMSScavengeBeforeRemark -XX:PretenureSizeThreshold=64m -XX:+U
eCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=50 -XX:CMSMaxAbortab
ePreambleTime=6000 -XX:+CMSParallelRemarkEnabled -XX:+ParallelRefProcEnabled -XX:-Omi
StackTracelnFastThrow -verbose:gc -XX:+PrintHeapAtGC -XX:+PrintGCDetails -XX:+PrintGCda
eStamps -XX:+PrintGCTimeStamps -XX:+PrintTenuringDistribution -XX:+PrintGCApplicationS
oppedTime -Xloggc:/opt/solr/server/logs/solr_gc.log -XX:+UseGCLogFileRotation -XX:Numbe
OfGCLogFiles=9 -XX:GCLogFileSize=20M -Dsolr.log.dir=/opt/solr/server/logs -Djetty.port=89
3 -DSTOP.PORT=7983 -DSTOP.KEY=solrrocks -Duser.timezone=UTC -Djetty.home=/opt/solr/
erver -Dsolr.solr.home=/opt/solr/server/solr -Dsolr.install.dir=/opt/solr -Dsun.net.inetaddr.ttl
60 -Dsun.net.inetaddr.negative.ttl=60 -Xss256k -jar start.jar --module=http
</span></span></code></pre>
```


-Xms512m -Xmx512m

用于设置 java 堆的最小值和最大值，将最大-Xmx 和最小-Xms 设置为一样可以避免堆自动扩展

参考: <a href="https://link.ld246.com/forward?goto=http%3A%2F%2Fwww.cnblogs.co
%2Flangtianya%2Fp%2F3898657.html" target="_blank" rel="nofollow ugc">JVM 调优总结 -Xs
s -Xmx -Xmn -Xss 等

参考: <a href="https://link.ld246.com/forward?goto=http%3A%2F%2Fwww.importnew.c
m%2F13954.html" target="_blank" rel="nofollow ugc">成为 Java GC 专家 (5) —Java 性能调
原则

-XX:NewRatio=3

用来指定新生代和整个堆的大小比例，或者直接用-XX:NewSize 来指定所需的新生代空间。
果设置了 NewRatio，那么整个堆空间的 1/(NewRatio + 1)就是新生代空间的大小。

使用 CMS 垃圾回收时，需要设置一个充足的新生代空间。然而，当新生代空间的大小超过一个
定的水平，程序的响应能力会被降低。

```
</ul>
</li>
<li>-XX:SurvivorRatio=4
<ul>
<li>Eden 区与 Survivor 区的大小比值。设置为 4,则两个 Survivor 区与一个 Eden 区的比值为 2:4,
个 Survivor 区占整个新生代的 1/6。 </li>
<li>参考: <a href="https://link.ld246.com/forward?goto=http%3A%2F%2Fwww.cnblogs.co
%2Ffredcreeen%2Farchive%2F2011%2F05%2F04%2F2037057.html" target="_blank" rel="nofoll
w ugc">JVM 系列三:JVM 参数设置、分析</a> </li>
</ul>
</li>
<li>-XX:TargetSurvivorRatio=90
<ul>
<li>参考: <a href="https://link.ld246.com/forward?goto=http%3A%2F%2Fwww.cnblogs.co
%2Fangtianya%2Fp%2F3898760.html" target="_blank" rel="nofollow ugc">JVM -XX: 参数介
</a> </li>
<li>参考: <a href="https://link.ld246.com/forward?goto=http%3A%2F%2Fblog.csdn.net%2F
ero_007%2Farticle%2Fdetails%2F52797684" target="_blank" rel="nofollow ugc">MaxTenuri
gThreshold 和 TargetSurvivorRatio 参数说明 </a> </li>
<li>设定 survivor 区的目标使用率。默认 50, 即 survivor 区对象目标使用率为 50%, 最高 90%。 <
li>
<li>设置 survivor 区的目标使用率, 当使用率达到时重新调整 TenuringThreshold 值, 让对象尽早
去 old 区。 </li>
</ul>
</li>
<li>-XX:MaxTenuringThreshold=8
<ul>
<li>设置对象在新生代中最大的存活次数,最大值 15,并行回收机制默认为 15,CMS 默认为 4。每经过
次 YGC, 年龄加 1, 当 survivor 区的对象年龄达到 TenuringThreshold 时, 表示该对象是长存活对
, 就会直接晋升到老年代。 </li>
<li>如果设置为 0 的话,则年轻代对象不经过 Survivor 区,直接进入老年代。对于老年代比较多的应用,
以提高效率。如果将此值设置为一个较大值,则年轻代对象会在 Survivor 区进行多次复制,这样可以增加
象再年轻代的存活时间,增加在年轻代即被回收的概率。该参数只有在串行 GC 时才有效。 </li>
</ul>
</li>
<li>-XX:+UseConcMarkSweepGC
<ul>
<li>使用 CMS 内存收集算法 </li>
<li>启用 CMS 低停顿垃圾收集器,减少 FGC 的暂停时间 </li>
</ul>
</li>
<li>-XX:+UseParNewGC
<ul>
<li>设置新生代为并行收集。可与 CMS 收集同时使用 JDK5.0 以上,JVM 会根据系统配置自行设置,
以无需再设置此值 </li>
</ul>
</li>
<li>-XX:ConcGCThreads=4
<ul>
<li>参考: <a href="https://link.ld246.com/forward?goto=http%3A%2F%2Ffive.com%2Fusef
l-jvm-flags-part-7-cms-collector%2F" target="_blank" rel="nofollow ugc">JVM 实用参数 (七
CMS 收集器</a> </li>
<li>标志-XX: ConcGCThreads=(早期 JVM 版本也叫-XX:ParallelCMSThreads)定义并发 CMS 过
运行时的线程数。 </li>
```


-XX:ParallelGCThreads=4

并行收集器的线程数。
此值最好配置与处理器数目相等，同样适用于 CMS

-XX:+CMSScavengeBeforeRemark

参考：JVM GC 算法 CM 详解(转)
在 CMS GC 前启动一次 ygc，目的在于减少 old gen 对 ygc gen 的引用，降低 remark 时的开
-----一般 CMS 的 GC 耗时 80% 都在 remark 阶段
开启-XX:+CMSScavengeBeforeRemark 选项，强制 remark 之前开始一次 minor gc，减少 re
ark 的暂停时间，但是在 remark 之后也将立即开始又一次 minor gc。

-XX:PretenureSizeThreshold=64m

对象超过多大是直接在旧生代分配
单位字节 新生代采用 Parallel Scavenge GC 时无效另一种直接在旧生代分配的情况是大的数组
象,且数组中无外部引用对象.

-XX:+UseCMSInitiatingOccupancyOnly

命令 JVM 不基于运行时收集的数据来启动 CMS 垃圾收集周期，禁止 hostspot 自行触发 CMS
C。
只有当我们充足的理由(比如测试)并且对应用程序产生的对象的生命周期有深刻的认知时，才应
使用该标志。

-XX:CMSInitiatingOccupancyFraction=50

设定 CMS 在对内存占用率达到 50% 的时候开始 GC(因为 CMS 会有浮动垃圾,所以一般都较早
动 GC);
这两个设置一般配合使用,一般用于『降低 CMS GC 频率或者增加频率、减少 GC 时长』的需求;

-XX:CMSInitiatingOccupancyFraction=70 是指设定 CMS 在对内存占用率达到 70% 的时候开
GC(因为 CMS 会有浮动垃圾,所以一般都较早启动 GC);
-XX:+UseCMSInitiatingOccupancyOnly 只是用设定的回收阈值(上面指定的 70%),如果不指定,J
M 仅在第一次使用设定值,后续则自动调整.

-XX:CMSMaxAbortablePrecleanTime=6000

- 参考：<a href="https://link.ld246.com/forward?goto=http%3A%2F%2Fblog.csdn.net%2F1xb
champagne%2Farticle%2Fdetails%2F18357947" target="_blank" rel="nofollow ugc">Tenured
区并发垃圾回收器 CMS 介绍
-XX:+CMSParallelRemarkEnabled

- 降低标记停顿

-

-

- -XX:+ParallelRefProcEnabled

-

- 参考: 一步优化 JVM 五: 优化延迟或者响应时间(3)

- 这个选项可以用 HotSpot VM 的任何一种垃圾回收器上, 他会是用多个的引用处理线程, 而不单个线程。这个选项不会启用多线程运行方法的 finalizer。他会使用很多线程去发现需要排队通知的 finalizable 对象。

-

-

- -XX:-OmitStackTraceInFastThrow

-

- 参考: 强要求 JVM 始终抛出含堆栈的异常

- 参考: java 中的 except on stack 有时候不输出的原因

-

-

- -verbose:gc

-

- 表示输出虚拟机中 GC 的详细情况.

- 参考: JVM 启动参数之 -verbose:gc

- 参考: vm 参数-verbose:gc 和-XX:+PrintGC 有区别?

-

-

- -XX:+PrintHeapAtGC

-

- 打印 GC 前后的详细堆栈信息

-

-

- -XX:+PrintGCDetails

-

- 需要在生产环境或者压测环境中测量这些参数下系统的表现, 这时候需要打开 GC 日志查看具体信息, 因此加上参数: -verbose:gc -XX:+PrintGCTimeStamps -XX:+PrintGCDetails -Xloggc:/home/test/logs/gc.log

-

-

- -XX:+PrintGCDateStamps

- -XX:+PrintGCTimeStamps

- -XX:+PrintTenuringDistribution

-

- 查看每次 minor GC 后新的存活周期的阈值

-

-

- -XX:+PrintGCApplicationStoppedTime

- 打印垃圾回收期间程序暂停的时间.可与上面混合使用

-
- -Xloggc:/opt/solr/server/logs/solr_gc.log

-
- 把相关日志信息记录到文件以便分析.与上面几个配合使用

-
- -XX:+UseGCLogFileRotation

-
- 启用 GC 日志文件的自动转储

-
- -XX:NumberOfGCLogFiles=9

-
- GC 日志文件的循环数目

-
- -XX:GCLogFileSize=20M

-
- 控制 GC 日志文件的大小
- 参考: -xx:+usegclogfilerotation

-
- Built-in support for GC log rotation has been added to the HotSpot JVM. It is described in the RFE 6941923 and is available in: Java 6 Update 34 Java 7 Update 2 (but there is no referen e to it in these release notes).
- There are three new JVM flags that can be used to enable and configure it:
- -XX:+UseGCLogFileRotation
must be used with -Xloggc;
-XX:NumberOfGCLogFiles=
must be >=1, default is one;
-XX:GCLogFileSize=M (or K)
default will be set to 512K.

-
-
-
- -Xss256k

-
- 每个线程的堆栈大小
- JDK5.0 以后每个线程堆栈大小为 1M,以前每个线程堆栈大小为 256K.更具应用的线程所需内存小进行 调整.在相同物理内存下,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有限制的,不能无限生成,经验值在 3000~5000 左右。

-
-
-

-
- <h3 id="1-2-javap反编译工具">1.2 javap 反编译工具</h3>

-
- 将 class 文件反编译为字节码
- 使用-XX:+PrintAssembly 参数来输出反汇编

<h3 id="1-3-java命令行参数使用">1.3 java 命令行参数使用</h3>

-XX:+PrintGCDetails, 在发生垃圾收集行为的时候打印内存回收日志。

