## 【JVM】1、java虚拟机参数-X 与 -XX的区别

Options that begin with -X are non-standard (not guaranteed to be supported on all VM implementations), and are subject to change without notice in subsequent releases of the JDK.

以 -X 开头的是非标准选项（不能保证被所有的 JVM 实现都支持），如果在后续版本的 JDK 中有变更恕不另行通知。

Options that are specified with -XX are not stable and are not recommended for casual use. These options are subject to change without notice.

指定 -XX 的选项是不稳定、不建议随便使用的。这些选项在今后变更恕不另行通知。

## Java 自带工具

Jvm虚拟机带来了诸多好处，同时也屏蔽了开发人员与内存的直接联系，Java提供了一些内部命令监测系统的运行状态。

**Jps**

Jps命令：

Jps –l Java 当前运行的进程列表

Jps –m java当前运行的进程列表简单信息

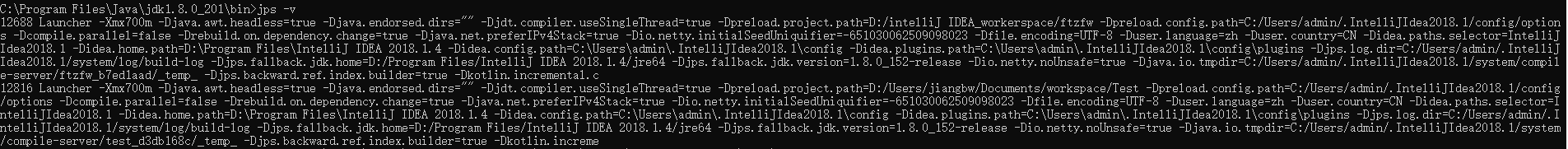
Jps –v 当前Jvm参数配置

查找当前虚拟机运行的进程列表

Jps –l 打印启动的虚拟机进程，以及详细类名



Jps –v 打印虚拟机启动的进程以及对应实例配置的系统参数



Jstat命令：

Jmap命令：

Jmap –head pid 获取堆内存的使用情况

jmap -dump:live,format=b,file=heap.bin 生成dump文件

应用：

**频繁GC问题或内存溢出问题**

一、使用jps查看线程ID

二、使用jstat -gc 3331 250 20 查看gc情况，一般比较关注PERM区的情况，查看GC的增长情况。

三、使用jstat -gccause：额外输出上次GC原因

四、使用jmap -dump:format=b,file=heapDump 3331生成堆转储文件

五、使用jhat或者可视化工具（Eclipse Memory Analyzer 、IBM HeapAnalyzer）分析堆情况。

六、结合代码解决内存溢出或泄露问题。

通过mat进行内存分析

Jstack 查看线程的列表，包括线程状态等信息

使用jps查看线程ID

使用jstack 3331：查看线程情况

Jstack pid | grep nid（十六进制） -A 30

查找pid中的线程十六进制为nid的 后三十行信息