Java HotSpot VM Options recommendations

Recommended Java Settings

The following Java HotSpot VM Options are recommended for running Fedora 4:

```
-Djava.awt.headless=true -Dfile.encoding=UTF-8 -server -Xms1024m -Xmx2048m -XX:MaxMetaspaceSize=512m -XX:
+UseG1GC -XX:+DisableExplicitGC
```

For Linux JVMs, see: http://docs.oracle.com/javase/8/docs/technotes/tools/unix/java.html

-Djava.awt.headless=true

- Headless mode is a system configuration in which the display device, keyboard, or mouse is lacking. Sounds unexpected, but actually you can perform different operations in this mode, even with graphic data.

-server

- The JDK includes two flavors of the VM -- a client-side offering, and a VM tuned for server applications. These two solutions share the Java HotSpot runtime environment code base, but use different compilers that are suited to the distinctly unique performance characteristics of clients and servers. These differences include the compilation inlining policy and heap defaults. Although the Server and the Client VMs are similar, the Server VM has been specially tuned to maximize peak operating speed. It is intended for executing long-running server applications, which need the fastest possible operating speed more than a fast start-up time or smaller runtime memory footprint.

-Dfile.encoding=UTF-8

- Tell the JVM to use UTF-8 as an encoding for text files to be more platform independent

-Xms512m

- Set the initial Java Heap memory size to 512 Megabytes

-Xmx1024m

- Set the maximal Java Heap memory size to 1024 Megabytes

-XX:NewSize=256m

- Set the initial size of new generation to 256 Megabytes

-XX:MaxNewSize=256m

- Set the maximum size of new generation to 256 Megabytes

-XX:MetaspaceSize=64m

- Set the size of the allocated class metadata space that will trigger a garbage collection the first time it is exceeded to 64 Megabytes

-XX:MaxMetaspaceSize=256m

- Set the maximum size of native memory that can be allocated for class metadata to 256 Megabytes. This is especially recommended when redeploying Fedora 4 often, since otherwise OutOfMemory errors will occur.

-XX:+UseG1GC

- Enables the use of the garbage-first (G1) garbage collector. It is a server-style garbage collector, targeted for multiprocessor machines with a large amount of RAM. It meets GC pause time goals with high probability, while maintaining good throughput. The G1 collector is recommended for applications requiring large heaps (sizes of around 6 GB or larger) with limited GC latency requirements (stable and predictable pause time below 0.5 seconds).

-XX:+DisableExplicitGC

- Disable calls to System.gc(), JVM still performs garbage collection when necessary.