ex1-JIT

Step 1

Java -XX:+PrintCompilation demo.MainV1 500

Step 2

Java -XX:+UnlockDiagnosticVMOptions -XX:+LogCompilation demo.MainV1 7000

Note : hotspot_pid19000.log file is created.

Search for random() and see the compiler C1 & C2

Step: 3

Arguments-> Program Arguments:7000

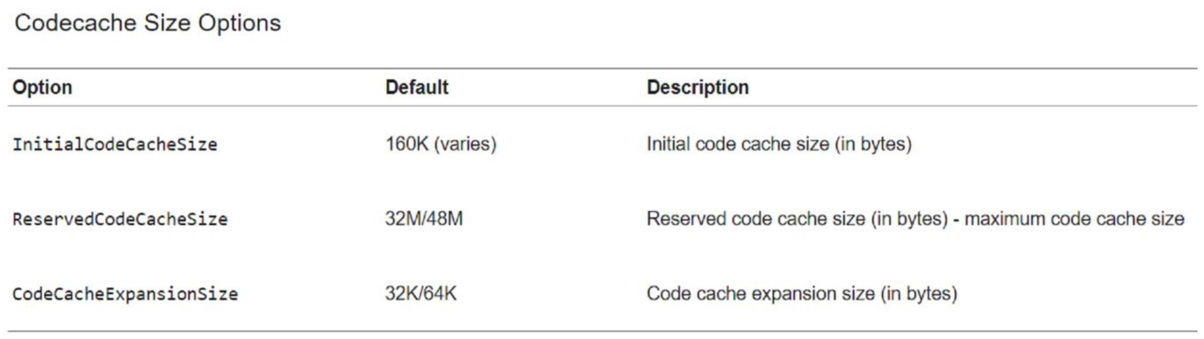
VM arguments: -XX:+PrintCodeCache

Step: 4

Tuning the code cache

VM Warning : Codecache is full. Compiler has been disabled

Set the below options :



Codecache Size Options

Option	Default	Description
`InitialCodeCacheSize`	160K (varies)	Initial code cache size (in bytes)
`ReservedCodeCacheSize`	32M/48M	Reserved code cache size (in bytes) - maximum code cache size
`CodeCacheExpansionSize`	32K/64K	Code cache expansion size (in bytes)

Now try with :

VM arguments: -XX:ReservedCodeCacheSize=28m -XX:+PrintCodeCache

<u>jvm compiler flags: -server, -client</u>

Step1:

Arguments-> Program Arguments: 10000

VM arguments -> -server -XX:+PrintCompilation

Step2:

Arguments -> Program Arguments: 10000

VM arguments -> -client -XX:+PrintCompilation

Note:

Disable TieredCompilation

VM arguments -> -XX:-TieredCompilation -XX:+PrintCompilation

<u>use jconsole to monitor the code cache</u>

note: jconsole process id is not showing, give write access to the below folder:

C:\Users\venkat\AppData\Local\Temp\hsperfdata_venkat

> Memory Pool (Code Cache)

<u>JVM Versions</u>

32-bit : Faster if heap < 3GB, Max heap size=4GB, Client Compiler only

64-but : it would be faster if using long/double, heap > 4GB, Client & Server Compiler

<u>Check the default java final flags</u>

java -XX:+PrintFlagsFinal

Check for the application process-id

Check the CICompilerCount value

[CICompilerCount to two tells the JVM to use number of threads for interpreter]

jinfo -flag CICompilerCount <process-id>

Now change the count and check the output:

VM arguments: -XX:CICompilerCount=6 -XX:+PrintCompilatin

CompileThreashold

[Number of interpreted method invocations before (re-)compiling]

jinfo -flag CompileThreshold <process-id>

Now, try with the lower threashold value

-XX:CICompilerCount=6 -XX:CompileThreshold=1000 -XX:+PrintCompilation

Reading the compiler's mind

The -XX:+LogCompilation flag produces a low-level XML file about compiler and runtime decisions

-XX:+UnlockDiagnosticVMOptions -XX:+LogCompilation -XX:+PrintInlining -XX:+PrintCompilation

Print Assembly:

-XX:+UnlockDiagnosticVMOptions -XX:+PrintAssembly

Exception:

"Could not load hsdis-amd64.dll; library not loadable; PrintAssembly is disabled".

Solution:

Download hsdis-1.1.1-win32-amd64.zip file from http://fcml-lib.com/download.html and copy hsdis-amd64.dll file in jdk bin folder.

(or) copy from this example lib folder

Ref : x86_64 Assembly to understand generate code