**Selecting the JVM**

* **The differences between 32 & 64 bit JVM:**

Whenever you download JVM, it will ask you to select 32 or 64 bit JVM.

If your machine is 32bit, then you have to use 32-bit JVM.

If your machine is 64bit, then you can you either one of them.

Table

Description automatically generated

Note: With Mac and Sparc OS, 32 bit is not available.

* **Specifying which compiler to use at runtime:**

We can specify the compiler to be picked by below command

java ***-client*** -XX:+PrintCompilation Main 15000

-client tells the 64Bit JVM to use the client (c1) compiler.

Table

Description automatically generated

Graphical user interface, application

Description automatically generated

With 32 bit JVM you can use either -client or -server.

With 64-bit JVM you can use only server compiler.

* **Turning off tiered compilation:**

**Graphical user interface, application

Description automatically generated**

The above flag allows you to turn off tiered compilation to tell the JVM to run in interpreted mode only.

* **Tuning native compilation within the Virtual Machine:**

Use below command to get number of threads available to compiling our code.

java -XX:+PrintFlagsFinal

Above command will print a long list, in that search for “***CICompilerCount***” that gives the number of available threads.

As above command prints bunch of items as an output, we can use another command which gives only the value we asked for.

jinfo -flag ***CICompilerCount*** java\_process\_id

To get the process\_id:

jps

localhost:~ spandiyan$ jinfo -flag CICompilerCount 4217

-XX:CICompilerCount=4

We can increase the CompilerCount by below command.

java -XX:CICompilerCount=6 -XX:+PrintCompilation Main 15000

We can increase the compiler threshold using below command

java -XX:CIComplierCount=6 -XX:CompileThreshold=1000 -XX:+PrintCompilation Main 5000

Note: When you set this threshold it makes a different at least in applications that are going to be running these same methods again and again relatively quickly or relatively early on in the application.