Hints and Tips for Java on z/OS – Performance

Topics:

1. Garbage Collection
2. JVM
3. z/OS UNIX System Services (USS)

Topic 1 - Garbage Collection

One of the great benefits of the Java platform is that it takes care of much of the work of garbage collection for you, but there are occasions when you still want to tweak the way garbage collection takes place. With the latest Java technology implementation from IBM, you can choose among several garbage collection policies and tune the size of your heap to help obtain the optimal performance out of your application.

For a better understanding about Java Garbage collection policies see the following links:
Java technology, IBM style: Garbage collection policies, Part 1
Java technology, IBM style: Garbage collection policies, Part 2

The IBM Monitoring and Diagnostic Tools for Java™ - Garbage Collection and Memory Visualizer is designed to help diagnose and analyze memory-related Java performance problems. For more details see the following link:
Java diagnostics, IBM style, Part 2: Garbage collection with the IBM Monitoring and Diagnostic Tools for Java - Garbage Collection and Memory Visualizer

For 64 bit Java applications, Java 6 provides the -Xcompressrefs option, which will significantly reduce the size of the Java heap storage required. See the section on 64 Bit Java.

Topic 2 - JVM

The following is a list of performance suggestions for the Java Virtual Machine (JVM) runtime environment:

- **Keep Current with SDK releases.**
  Although Java has matured significantly over the last few years, it is a relatively new technology. Almost every release of the SDK has a double digit performance enhancement. In some case PTFs have double digit performance enhancements. By keeping current with the latest release, you will be able to take advantage of this progress.
- **Prudent use of Zip and Jar files can improve load time.**
  Zip and Jar files can be used to combine many class files into one file for easier loading or file transfer. When done properly, this can improve application load time. However,

avoid adding unneeded class files which will increase file size and increase memory usage. One common mistake is to produce a zip file of every utility and builder class when the application may need only a small proportion of the class files to execute. This results in increased load time.

- **Reorder the order in CLASSPATH**
  When the JVM is looking for a class, it searches the directories in the order they are given in CLASSPATH. You can improve the start-up times of your Java applications by reordering the paths in the CLASSPATH environment variable by placing the most used libraries earlier in the CLASSPATH.

Topic 3 - z/OS UNIX System Services (USS)

The JVM uses UNIX System Services for z/OS (USS) for base operating system functions. For this reason, USS must be correctly installed and tuned in order to get optimum performance from the JVM. See USS performance for detailed tuning suggestions.

For additional information, see: z/OS UNIX System Services

Several Unix based tools for z/OS are available for free via download. Some of these are Java specific. These tools were designed for OS/390 UNIX, by IBM developers and testers. There are no warranties of any kind, and there is no service or technical support available for these from IBM. See the z/OS UNIX Tools .