Memory

Permalink to this page: https://cwiki.apache.org/confluence/x/RSolBg

Preface

This page discusses various memory issues. In a nutshell - if your computer has less than 128MB of ram - you will probably have trouble. Anyhow, also read the following threads for other memory related issues:

- java.lang.OutOfMemoryError during deploy
- Memory requirements
- Memory Mgmt Tomcat
- Tomcat Out of memory
- Tracking memory usage over time

Also look at YourKit, or maybe you IDE has a profiling tool in it, or other profiling tools are available. (The following tools were recommended by many people in the past, but now seem to be discontinued by their vendors: JProbe by Quest Software — the company was acquired by Dell, OptimizeIt by Borland). This is not an endorsement for them, I just notice other people like them.

Questions

1. How do I adjust memory settings?
2. Why do I get OutOfMemoryError errors?
3. How much memory is Tomcat/webapp/??? using?

Answers

How do I adjust memory settings?

First look at `java -X` to determine what parameters to set. Then you can set them via the environment variable `CATALINA_OPTS` (using `JAVA_OPTS` also works, but is not recommended). This variable is usually set in a file `bin/setenv.sh` or `bin/setenv.bat` that you may need to create by yourselves.

The `setenv` file is documented in RUNNING.txt in your version of Tomcat. The environment variables are described in a comment at the top of catalina. bat or catalina.sh files.

Why do I get `OutOfMemoryError` errors?

Many reasons.

- You're out of memory. Simple as that - add more to your heap.
- You're out of memory. You have code which is hanging onto object references and the garbage collector can't do its job. Get a profiler to debug this one.
- You ran out of file descriptors. If you are on a *nix system, it has been observed that an `OutOfMemoryError` can be thrown if you run out of file descriptors. This can occur if your threshold is too low. The `ulimit` program can help you out here. You also may need to account for socket connections too when thinking about these thresholds. Google is your friend for getting more information about this topic.
- You have too many threads running. Some OS's have a limit to the number of threads which may be executed by a single process. (Which is what the JVM is.) Refer to your OS docs for more information on how to raise this threshold.
- If you have a lot of servlets or JSP's, you may need to increase your permanent generation. By default, it is 64MB. Quadrupling it to be `-XX:MaxPermSize=256m` might be a good start.
- Your OS limits the amount of memory your process may take. OK, this one is grasping at straws.
- The JVM has a bug. This has been known to happen with JVM1.2.? and using EJB's with another servlet engine.
- Not actually a reason - but on your particular platform, look at the `java -X` options. They may be VERY helpful.
- Your classloaders are not being garbage collected.
- You run out of process memory (non java/GC memory), for example when using java.util.zip classes or JNI classes allocating process memory. See Instantiating Inflater/Deflater causes OutOfMemoryError; finalizers not called promptly enough

See also OutOfMemory and MemoryLeakProtection.

How much memory is Tomcat/webapp/??? using?

- To find out how much memory Tomcat is using, you might be able to use the `Runtime` class provided by the JDK.
- You can't find out how much memory a webapp is using. The JVM doesn't give us these detail.
- You can't find out how much memory a ??? is using. The JVM doesn't give us these detail.
- That being said, a memory profiling tool might prove the above statements wrong - but you probably don't want to use them in a production environment.