Performance Tuning DSpace

1 Review the number of DSpace webapps you have installed in Tomcat
2 Give Tomcat (DSpace UIs) More Memory
    2.1 Give Tomcat More Java Heap Memory
    2.2 Give Tomcat More Java PermGen Memory
3 Choosing the size of memory spaces allocated to DSpace
4 Give the Command Line Tools More Memory
    4.1 Give the Command Line Tools More Java Heap Memory
    4.2 Give the Command Line Tools More Java PermGen Space Memory
5 Give PostgreSQL Database More Memory

The software DSpace relies on does not come out of the box optimized for large repositories. Here are some tips to make it all run faster.

Review the number of DSpace webapps you have installed in Tomcat

By default, DSpace includes a number of web applications which all interact with the underlying DSpace data model. The DSpace web applications include: XMLUI, JSPUI, OAI, RDF, REST, SOLR, SWORD, and SWORDv2. The only **required** web application is SOLR as it is utilized by several of the other web applications (XMLUI, JSPUI and OAI). See the Installing DSpace documentation for more information about each of these web applications.

Any of the other web applications can be removed from your Tomcat, if you have no plans to utilize that functionality. *The fewer web applications you are running, the less memory you will require, as each of these applications will be allocated memory when started up by Tomcat.*

Give Tomcat (DSpace UIs) More Memory

Give Tomcat More Java Heap Memory

Java Heap Memory Recommendations

At the time of writing, DSpace recommends you should give Tomcat >= 512MB of Java Heap Memory to ensure optimal DSpace operation. Most larger sized or highly active DSpace installations however tend to allocate more like 1024MB to 2048MB of Java Heap Memory.

Performance tuning in Java basically boils down to memory. If you are seeing "`java.lang.OutOfMemoryError: Java heap space`" errors, this is a sure sign that Tomcat isn't being provided with enough Heap Memory.

Tomcat is especially memory hungry, and will benefit from being given lots of RAM. To set the amount of memory available to Tomcat, use either the `JAVA_OPTS` or `CATALINA_OPTS` environment variable, e.g:

```
CATALINA_OPTS=-Xmx512m -Xms512m
```

OR

```
JAVA_OPTS=-Xmx512m -Xms512m
```

The above example sets the maximum Java Heap memory to 512MB.

Difference between JAVA_OPTS and CATALINA_OPTS

You can use either environment variable. `JAVA_OPTS` is also used by other Java programs (besides just Tomcat). `CATALINA_OPTS` is *only used* by Tomcat. So, if you only want to tweak the memory available to Tomcat, it is recommended that you use `CATALINA_OPTS`. If you set **both** `CATALINA_OPTS` and `JAVA_OPTS`, Tomcat will default to using the settings in `CATALINA_OPTS`.

If the machine is dedicated to DSpace a decent rule of thumb is to give tomcat half of the memory on your machine. **At a minimum, you should give Tomcat >= 512MB of memory for optimal DSpace operation.** (*NOTE: As your DSpace instance gets larger in size, you may need to increase this number to the several GB range.*) The latest guidance is to also set `-Xms` to the same value as `-Xmx` for server applications such as Tomcat.

Give Tomcat More Java PermGen Memory

Java PermGen Memory Recommendations

At the time of writing, DSpace recommends you should give Tomcat >= 128MB of PermGen Space to ensure optimal DSpace operation.

If you are seeing "`java.lang.OutOfMemoryError: PermGen space`" errors, this is a sure sign that Tomcat is running out PermGen Memory. (More info on PermGen Space: http://blogs.sun.com/fkieviet/entry/classloader_leaks_the_dreaded_java)

To increase the amount of PermGen memory available to Tomcat (default=64MB), use either the `JAVA_OPTS` or `CATALINA_OPTS` environment variable, e.g:

```
CATALINA_OPTS=-XX:MaxPermSize=128m
```

OR

```
JAVA_OPTS=-XX:MaxPermSize=128m
```

The above example sets the maximum PermGen memory to 128MB.

Difference between JAVA_OPTS and CATALINA_OPTS

You can use either environment variable. `JAVA_OPTS` is also used by other Java programs (besides just Tomcat). `CATALINA_OPTS` is *only used* by Tomcat. So, if you only want to tweak the memory available to Tomcat, it is recommended that you use `CATALINA_OPTS`. If you set **both** `CATALINA_OPTS` and `JAVA_OPTS`, Tomcat will default to using the settings in `CATALINA_OPTS`.
Please note that you can obviously set **both** Tomcat's Heap space and PermGen Space together similar to:
`CATALINA_OPTS=-Xmx512m -Xms512m -XX:MaxPermSize=128m`
On an Ubuntu machine (10.04) at least, the file `/etc/default/tomcat6` appears to be the best place to put these environmental variables.

Choosing the size of memory spaces allocated to DSpace

psi-probe is a webapp that can be deployed in DSpace and be used to watch memory usage of the other webapps deployed in the same instance of Tomcat (in our case, the DSpace webapps).

1. Download the latest version of psi-probe from https://github.com/psi-probe/psi-probe
2. Unzip probe.war into [dspace]/webapps/

   ```
   cd [dspace]/webapps/
   unzip ~/probe-3.1.0.zip
   unzip probe.war -d probe
   ```

3. Add a Context element in Tomcat's configuration, and make it privileged (so that it can monitor the other webapps):
   EITHER in `$CATALINA_HOME/conf/server.xml`

   ```
   <Context docBase="[dspace]/webapps/probe" privileged="true" path="/probe" />
   ```

   OR in `$CATALINA_HOME/conf/Catalina/localhost/probe.xml`

   ```
   <Context docBase="[dspace]/webapps/probe" privileged="true" />
   ```

4. Edit `$CATALINA_HOME/conf/tomcat-users.xml` to add a user for logging into psi-probe (see more in https://github.com/psi-probe/psi-probe/wiki/InstallationApacheTomcat)

   ```
   <?xml version='1.0' encoding='utf-8'?>
   <tomcat-users>
     <user username="admin" password="t0psecret" roles="manager" />
   </tomcat-users>
   ```

5. Restart Tomcat
6. Open http://yourdspace.com:8080/probe/ (edit domain and port number as necessary) in your browser and use the username and password from tomcat-users.xml to log in.

In the "System Information" tab, go to the "Memory utilization" menu. Note how much memory Tomcat is using upon startup and use a slightly higher value than that for the `-Xms` parameter (initial Java heap size). Watch how big the various memory spaces get over time (hours or days), as you run various common DSpace tasks that put load on memory, including indexing, reindexing, importing items into the oai index etc. These maximum values will determine the `-Xmx` parameter (maximum Java heap size). Watching PS Perm Gen grow over time will let you choose the value for the `-XX:MaxPermSize` parameter.

Give the Command Line Tools More Memory

Give the Command Line Tools More Java Heap Memory

Similar to Tomcat, you may also need to give the DSpace Java-based command-line tools more Java Heap memory. If you are seeing "`java.lang.OutOfMemoryError: Java heap space`" errors, when running a command-line tool, this is a sure sign that it isn't being provided with enough Heap Memory.

By default, DSpace only provides 256MB of maximum heap memory to its command-line tools.

If you'd like to provide **more** memory to command-line tools, you can do so via the `JAVA_OPTS` environment variable (which is used by the `[dspace]/bin/dspace` script). Again, it's the same syntax as above:

```
JAVA_OPTS=-Xmx512m -Xms512m
```

This is especially useful for big batch jobs, which may require additional memory.

You can also edit the `[dspace]/bin/dspace` script and add the environmental variables to the script directly.

Give the Command Line Tools More Java PermGen Space Memory

Similar to Tomcat, you may also need to give the DSpace Java-based command-line tools more PermGen Space. If you are seeing "`java.lang.OutOfMemoryError: PermGen space`" errors, when running a command-line tool, this is a sure sign that it isn't being provided with enough PermGen Space.

By default, Java only provides 64MB of maximum PermGen space.

If you'd like to provide **more** PermGen Space to command-line tools, you can do so via the `JAVA_OPTS` environment variable (which is used by the `[dspace]/bin/dspace` script). Again, it's the same syntax as above:

```
JAVA_OPTS=-XX:MaxPermSize=128m
```

This is especially useful for big batch jobs, which may require additional memory.

Please note that you can obviously set **both** Java's Heap space and PermGen Space together similar to:
`JAVA_OPTS=-Xmx512m -Xms512m -XX:MaxPermSize=128m`

Give PostgreSQL Database More Memory

On many linux distros PostgreSQL comes out of the box with an incredibly conservative configuration - it uses only 8Mb of memory! To put some more fire in its belly edit the `shared_buffers` parameter in `postgresql.conf`. The memory usage is 8KB multiplied by this value. The advice in the Postgres docs is not to increase it above 1/3 of the memory on your machine.

For More PostgreSQL Tips

For more hints/tips with PostgreSQL configurations and performance tuning, see also:

- PostgresPerformanceTuning
- PostgresqlConfiguration