Tomcat Administration

VM Start Parameter

Änderungen an den Java VM Start Parametern sind eine Maßnahme des Performancetuning und der Leistungsverbesserung. Durch diese Änderungen können u.A. OutOfMemoryErrors vermieden werden, wie sie z.B. durch einen zu kleinen 'PermGen Space' entstehen können.

Auf 32-bit Betriebssystemem sollte die Max Heap Size (-Xmx) auf *höchstens 1536m* eingestellt werden. Details finden sich hier.

Während die Oracle Implementierung von Java *immer* Heap-Size limitiert ist, stellt IBM auch VMs ohne diese Limitierung zur Verfügung. Daneben gibt es auch normale VM Implementierungen von IBM.

`setenv.sh` und `setenv.bat`

Die Java VM Start Parameter können verändert werden, indem Sie im Tomcat Installationsverzeichnis im Ordner 'bin' die Datei '`setenv.sh`' (Linux) bzw. '`setenv.bat`' (Windows) anlegen. In dieser Datei können Sie einige Umgebungsvariablen setzen, die dann beim Starten herangezogen werden. Diese Umgebungsvariablen sind in der (schon vorhandenen) Datei '`catalina.sh`' bzw. '`catalina.bat`' dokumentiert.

In einer Nuclos Installation befindet sich das oben erwähnte Verzeichnis unter '`<nuclos>/tomcat/apache-tomcat-<version>/bin`'.

Vorschlag für eine `setenv.sh` Datei (Linux):

```
# Created by Thomas Pasch
# see catalina.sh for explanation
JAVA_OPTS=""
CATALINA_OPTS="-server -ea -XX:PermSize=128M -XX:MaxPermSize=256M -XX:+UseThreadPriorities -Xmx1236m -Xms512m -
XX:+PrintGCTimeStamps -XX:+HeapDumpOnOutOfMemoryError -XX:+TraceClassUnloading"

# http://www.theserverside.com/discussions/thread.tss?thread_id=63241
#-Xloggc:$CATALINA_HOME/logs/gc.log or Xloggc:%CATALINA_HOME%/logs/gc.log
#-XX:+PrintHeapAtGC
#-XX:+PrintGCDetails
#-XX:+PrintGCTimeStamps
#-XX:-HeapDumpOnOutOfMemoryError
#-XX:+UseConcMarkSweepGC
#-XX:-TraceClassUnloading
# http://stackoverflow.com/questions/202502/appropriate-tomcat-5-5-start-up-parameters-to-tune-jvm-for-
extremely-high-demand
# http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136373.html
# http://www.oracle.com/technetwork/java/javase/gc-tuning-6-140523.html
```

Vorschlag für eine `setenv.bat` Datei (Windows):

```
rem Created by Thomas Pasch
rem see catalina.sh for explanation
set "JAVA_OPTS="
set "CATALINA_OPTS=-ea -XX:PermSize=128M -XX:MaxPermSize=256M -XX:+UseThreadPriorities -Xmx1236m -Xms512m -XX:
+PrintGCTimeStamps -XX:+HeapDumpOnOutOfMemoryError -XX:+TraceClassUnloading"
```

Die VM Parameter aus `setenv.bat` werden für den Windows Tomcat **Dienst** *nicht* verwendet. Sie müssen wie im nächsten Abschnitt erläutert angepasst werden.

Der Installer **löscht** z.Z. eine vorhandene `setenv.bat` Datei. Sie muss also nach dem 'Drüber'-Installieren wieder in das Verzeichnis kopiert werden.

Eine Alternative zu dem Anlegen einer `setenv.sh/setenv.bat` Datei ist es, das Startscript der Installation `startup.sh/startup.bat` unter '`<nuclos>/bin`' anzupassen. Beispiel für `startup.bat`:

```
@echo off
rem Nuclos Server Script
if "%OS%" == "Windows_NT" setlocal
set CATALINA_HOME=/home/tpasch2/nuclos_tmp/tomcat/apache-tomcat-7.0.30
set JRE_HOME=/opt/java/jdk1.6.0_37/jre
set JAVA_OPTS=-ea -XX:PermSize=128M -XX:MaxPermSize=256M -XX:+UseThreadPriorities -Xmx1236m -Xms512m -
XX:+PrintGCTimeStamps -XX:+HeapDumpOnOutOfMemoryError -XX:+TraceClassUnloading
call "%CATALINA_HOME%\bin\catalina.bat" run
```

Windows Dienst/Service

Das Modifizieren des von Nuclos installierten Dienstes ist mit folgendem Windows Script möglich:

```
set CATALINA_HOME=C:\Nuclos3.6\tomcat\apache-tomcat-7.0.28
set CATALINA_BASE=%CATALINA_HOME%
set JAVA_HOME=C:\Program Files\Java\jre6
C:\Nuclos3.6\bin\nuclosx64.exe //US//nuclos.nuclos --JvmOptions "-Dcom.sun.management.jmxremote.port=30333#-
Dcom.sun.management.jmxremote.ssl=false#-Dcom.sun.management.jmxremote.authenticate=false#-Dcatalina.base=%
CATALINA_BASE%#-Dcatalina.home=%CATALINA_HOME%#-Djava.endorsed.dirs=%CATALINA_HOME%\endorsed#-Djava.io.tmpdir=%
CATALINA_BASE%\temp#-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager#-Djava.util.logging.
config.file=%CATALINA_BASE%\conf\logging.properties#-ea#-XX:PermSize=128M#-XX:MaxPermSize=256M#-XX:
+UseThreadPriorities#-XX:+PrintGCTimeStamps#-XX:+HeapDumpOnOutOfMemoryError#-XX:+TraceClassUnloading" --
JvmMs=512m --JvmMx=2048m --StartMode=jvm --StopMode=jvm --Startup=manual --Classpath "%JAVA_HOME%/lib/tools.
jar;%CATALINA_HOME%\bin\bootstrap.jar;%CATALINA_HOME%\bin\tomcat-juli.jar" --StartClass=org.apache.catalina.
startup.Bootstrap --StartParams=start --StopClass=org.apache.catalina.startup.Bootstrap --StopParams=stop
```

Erläuterungen:

- '`nuclos.nuclos`' ist der Dienstnahme. Der Teil hinter dem Punkt ist der im Installer angegebene Nuclos Instanzname. Dieser kann bei Ihnen anders lauten.
- Eine Alternative zu '`--Startup=manual`' ist '`--Startup=auto`'. Dann wird der Dienst beim Starten des Betriebssystems automatisch gestartet.
- In dem Script sind die VM Start Parameter ab '`-ea`' beispielhaft. Die Parameter davor sind dagegen verpflichtend.
- Die Java VM für Windows kennt ***als Dienst* kein '`-server`' Flag**. Deshalb ist es auch nicht angegeben.
- Nein, ich weiß auch nicht, warum diese Änderung unter Windows kompliziert ist.

Weiterführende Informationen zum Tomcat als Windows Dienst sind im Howto zu finden.

Wenn man `-Xms` und `-Xms` in den `--JvmOptions` angibt, dann führt das dazu, dass die Parameter 2x an die Tomcat Java VM übergeben werden. Damit ist es nicht klar, welcher Parameter verwendet wird. Daher die Start und Maximal Heap Size *ausschließlich* über `--JvmMs` und `--JvmMx` wie im Beispielskript oben setzen.

Die 3 angegebenen JMX Parameter sind wichtig, um mit der `jvisualvm` den Serverprozess überwachen zu können. Der Port darf nicht von einer anderen Anwendung bereits verwendet werden (z.B. von einer weiteren Nuclos Server Instanz).

Die gesetzten Java VM Parameter sollte man ausprobieren, *bevor* man den Service wie hier beschrieben verändert. Denn setzt man z.B. -Xmx zu groß, dann startet der Service nicht, kann aber auch nicht gestoppt werden, so dass ein Reboot fällig wird.

Die für Windows zusätzlich angegebenen JMX Parameter (`-Dcom.sun.management.jmxremote.port=30333 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false`) sind notwendig, um ein Speicherabbild zu erstellen, *wenn Ihr Nuclos Server als Dienst läuft*. Sie müssen selbst entscheiden, ob Sie diese Parameter auch für eine Produktionsumgebung verwenden wollen.

Diese Art von Skript wird auch im Windows Installer verwendet, um die VM Parameter des Tomcat Service zu setzen (in `org.nuclos.installer.unpack.WindowsUnpacker.register`).

NIO Connector benutzen

Der NIO Connector bietet einige Feature (siehe AIO) und bessere Performance. In conf/server.xml statt:

```
 <Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />
```

folgendes eintragen:

```
 <Connector port="8080" protocol="org.apache.coyote.http11.Http11NioProtocol" connectionTimeout="20000"
redirectPort="8443" />
```

Weiter Informationen enthält eine Übersicht über die wichtigsten Tomcat Connectors.

Tomcat Native

Für optimale Performance in der Produktion. Zunächst müssen (unter Ubuntu) die Pakete `libssl-dev` und `libapr1-dev` installiert werden. Danach lässt sich tomcat-native unter Linux recht schnell bauen.

```
cd <tomcat-native-...>/jni/native
./configure --with-ssl --with-apr=/usr/bin/apr-1-config --prefix=/home/tpasch2/apache-tomcat-7.0.59 --with-java-
home=/usr/lib/jvm/java-8-oracle
make
make install
make clean
```

Für die Benutzung in Eclipse/STS in der Tomcat 'Launch Configuration' im Reiter 'Environment' die Variable 'LD_LIBRARY_PATH' mit dem Wert '/home /tpasch2/apache-tomcat-7.0.29' (d.h. der --prefix Wert aus ./configure) eintragen.

Weiterführende Informationen Tomcat Native und Tomcat APR.

Tomcat Logging

Access Log

Das Tomcat Access Log ist konfigurierbar. Hier ist es u.A. möglich, die Verarbeitungszeit und die Länge der Antwort zu loggen. Details finden sich hier.

Erweitertes Logging durch Valves

Durch das Konfigurieren von Valves ist es möglich, sehr viele weitere Informationen (z.B. Informationen zum SSL Handshake, langsame Threads) zu loggen.

Erweitertes Logging durch Servlet Filter

Einige Servlet Filter werden direkt vom Tomcat zur Verfügung gestellt und können (durch eine Konfiguration in web.xml) verwendet werden, um z.B. die Länge der Anfrage in Bytes zu loggen. Eine Übersicht findet sich hier.

HTTPS, SSL/TLS

Zertifikat erstellen mit:

```
keytool -genkeypair -keystore .keystore -alias mykey -storepass storepw -keypass keypw
```

Dann in server.xml den entsprechenden Connector konfigurieren:

```
  <Connector SSLEnabled="true" clientAuth="false" keystoreFile="/home/tpasch2/sslcert/.keystore" keystorePass="
storepw" keyPass="keypw" keyAlias="mykey" maxThreads="150" port="8443" protocol="HTTP/1.1" scheme="https" secure="
true" sslProtocol="TLS"/>
```

> Der Installer richtet HTTPS entsprechend in der Datei `server.xml` ein, setzt aber *kein* Schlüsselpasswort (`keyPass`). Daher muss die Datei nach dem Ausführen des Installers noch manuell angepasst werden, wenn das Schlüsselpasswort nicht leer ist.

Eine Installation, mehrere Tomcat Instanzen

Dies kann man durch ein Trennen des Tomcats in CATALINA_HOME (die Installation) und CATALINA_BASE (einige Konfigurationsdateien u.a.) erreichen. Nähere Informationen finden sich hier.

WAR Deployment über den Manager

Die Tomcat Installation beinhaltet auch den Manager. Mit ihm lassen sich WAR Dateien direkt auf dem Tomcat deployen (auch von Remote!). Um den Manager nutzen zu können, muss man Datei `tomcat-users.xml` in CATALINA_BASE/conf Verzeichnis anpassen. Hier ein Beispiel:

```
<tomcat-users>
        <role rolename="manager-gui"/>
        <role rolename="manager-script"/>
        <role rolename="manager-jmx"/>
        <role rolename="manager-status"/>
        <role rolename="admin-gui"/>
        <role rolename="admin-script"/>
        <user password="nuclos" roles="manager-gui,manager-script,manager-jmx,manager-status,admin-gui,admin-
script" username="nuclos"/>
</tomcat-users>
```

Ferner sind die Nuclos WARs so groß, dass in `CATALINA_HOME/webapps/manager/WEB-INF/web.xml` der Tag `max-file-size` (und evt. `max-request-size`) angepasst werden muss.

Hot Deployment

Der Tomcat unterstützt mehrere Methoden des Hot Deployments, u.a. den Manager (s.o.) und autoDeploy, bei denen auf Änderungen im Dateisystem reagiert wird. Details finden sich hier und da.

Mit einer geschickten Wahl der Java VM Startparameter (und weiteren Maßnahmen) ist es möglich, für die Entwicklung das Hot Deployment Verhalten des Tomcats entscheidend zu verbessern.

Wiki Eintrag von Oliver: JS WebClient Entwicklungsumgebung

Details: https://ducquoc.wordpress.com/2010/11/06/eclipse-wtp-tomcat-hot-deploy/