

Installer et configurer mod_jk pour l'interfacage entre Tomcat et Apache

Introduction

Nous avons vu dans le guide InstallerTomcat comment faire une installation et une configuration minimale de Tomcat. Vous pouvez donc désormais exécuter la webapp de test fournie avec Tomcat sur l'adresse <http://nsXXXX.ovh.net:8080/jsp-examples/>.

Vous aimeriez maintenant éviter à vos visiteurs de taper systématiquement le port de Tomcat (8080) lorsqu'ils accèdent à une webapp hébergée sur votre serveur. Pour cela, nous allons installer et configurer mod_jk, un module pour Apache qui accepte les connexions sur le port habituel (80) puis traduit la requête pour la transmettre à Tomcat. De cette façon, l'accès à la webapp se fait de manière transparente avec une URL du type <http://nsXXXX.ovh.net/jsp-examples/>.

Procédure

• Installation du module mod_jk

Commençons par récupérer les sources du connecteur puis compilons le module Apache :

```
[root@julien /] wget
http://www.apache.org/dist/jakarta/tomcat-connectors/jk/source/jk-1.2.13/jakarta-tomcat-connectors-1.2.13-src.tar.gz
-O /home/ovh/src/soft/jakarta-tomcat-connectors-1.2.13-src.tar.gz

[root@julien /] cd /home/ovh/src/

[root@julien src] tar xvfz ./soft/jakarta-tomcat-connectors-1.2.13-src.tar.gz

[root@julien src]# cd jakarta-tomcat-connectors-1.2.13-src/jk/native/

[root@julien native]# ./configure --with-apxs=/usr/local/apache/bin/apxs

[root@julien native]# make

[root@julien native]# make install
```

Vérifions ensuite que le module a bien été copié avec les modules chargés dynamiquement par Apache :

```
[root@julien /]#ls -al /usr/local/apache/libexec/
total 2768
drwxr-xr-x 2 root root 4096 jun 25 12:27 .
drwxr-xr-x 11 root root 4096 jan 21 18:42 ..
-rw-r--r-- 1 root root 8442 jun 22 13:24 httpd.exp
```

```
-rw-r--r-- 1 root root 938846 jun 22 14:44 mod_jk.a
-rw-r--r-- 1 root root 709 jun 22 14:44 mod_jk.la
lrwxrwxrwx 1 root root 15 jun 22 14:44 mod_jk.so -> mod_jk.so.0.0.0
lrwxrwxrwx 1 root root 15 jun 22 14:44 mod_jk.so.0 -> mod_jk.so.0.0.0
-rwxr-xr-x 1 root root 330200 jun 22 14:44 mod_jk.so.0.0.0
```

• Configuration d'Apache

Nous allons maintenant éditer le fichier de configuration d'Apache (/httpd.conf) pour prendre en compte le nouveau module. Commençons par repérer la section où Apache charge les modules et ajoutons cette ligne :

```
LoadModule jk_module libexec/mod_jk.so
```

Ajoutons ensuite cette section pour configurer le module (par exemple juste au dessus de la déclaration des Virtual Hosts) :

```
<IfModule? mod_jk.c>
JkWorkersFile? /usr/share/tomcat/conf/workers.properties
JkLogFile? /var/log/httpd/mod_jk.log
JkLogLevel? info
JkMount? /servlets-examples/* ajp13
JkMount? /jsp-examples/* ajp13
</IfModule>
```

Cette section indique où trouver la configuration du connecteur, où stocker les logs générées par le module et surtout les URL à prendre en compte par le module. Ce dernier point est très important car il indique à Apache les requêtes qui doivent être transmises à Tomcat. Par exemple, la directive `JkMount? /jsp-examples/* ajp13` indique que toute requête contenant `/jsp-examples/` doit être transmise à Tomcat.

• Configuration du connecteur 'Apache-Tomcat'

Passons maintenant à la configuration du connecteur. Le fichier `/usr/share/tomcat/conf/workers.properties` existe peut être déjà, vous pouvez le simplifier pour conserver uniquement cette configuration :

```
workers.tomcat_home=/usr/share/tomcat
workers.java_home=/usr/local/java
ps=/
worker.list=ajp13
```

```
worker ajp13.port=8009  
worker ajp13.host=localhost  
worker ajp13.type=ajp13
```

- **Redémarrage et test**

Il reste maintenant à redémarrer Apache :

```
[root@julien /] /etc/init.d/httpd restart
```

Puis vérifier que mod_jk a bien été pris en compte:

```
[root@julien /]# tail -f /var/log/httpd/error_log  
[Tue Jun 22 15:13:11 2004] [notice] Apache/1.3.31 (Unix) mod_jk/1.2.13 mod_gzip/1.3.19.1a PHP/4.3.7  
mod_ssl/2.8.18 OpenSSL/0.9.6m
```

Si tout va bien, l'URL <http://nsXXXX.ovh.net/jsp-examples/> pointe bien vers la webapp de test de Tomcat.

Conclusion

Le module et les connecteurs sont maintenant configurés, vous pouvez ajouter d'autres points de montage pour vos propres webapps et ainsi faire cohabiter des applications en HTML/PHP et des applications en JSP/Servlets tout en interrogeant uniquement Apache. Le module mod_jk ainsi que le connecteur peuvent également être configurés de manière plus fine, de nombreuses options existent, comme la possibilité d'utiliser plusieurs serveurs Tomcat et de configurer la répartition de charge directement au niveau du connecteur 'Apache-Tomcat' (également appelé worker).

Pour plus de détails: <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jk2/>.