Mere Tomcat

Denne artikel beskriver brug af virtual hosts i Tomcat og hvordan man kan sætte en almindelig Apache web-server foran Tomcat.

Den forudsætter lidt kendskab til Tomcat og Apache web-server.

Skrevet den **11. feb 2009** af **arne_v** I kategorien **Programmering / JSP** | ⭐⭐⭐⭐☆

Historie:
V1.0 - 16/04/2005 - original

Virtual host begrebet

Der er grundliggende 2 måder at lade en web-server servicere flere end et domain.

Man kan lade maskinen have flere IP adresser og lade hvert domain pege på hver sin IP adresse.

www.hotel.dk peger på 123.456.789.1 (en A record i DNS)
www.abc.dk peger på 123.456.789.2 (en A record i DNS)
www.xyz.dk peger på 123.456.789.3 (en A record i DNS)

web-serveren lytter på begge adresser og håndterer dem seperat.

Det virker, men er meget upraktisk hvis man har mange domains på samme server.

Derfor har man oprettet muligheden for virtual hosts med samme IP adresse.

Man lader de forskellige domains pege på samme IP adresse.

www.hotel.dk peger på 123.456.789.1 (en A record i DNS)
www.abc.dk peger på 123.456.789.1 (en CNAME record i DNS)
www.xyz.dk peger på 123.456.789.1 (en CNAME record i DNS)

Og det kan lade sig gøre fordi browseren sender det navn den kender med i requesten.

En HTTP request ser så ud som:

GET /index.html HTTP/1.1<CR><LF>
Host: www.abc.dk<CR><LF>
<CR><LF>

og serveren kan så bruge den host header til at håndtere requesten rigtigt.

Det bør virke i alle browsere idag. Sidste browser version der ikke understøttede det var NetScape 1.x.

Vær opmærksom på at diverse hjemmestrikkede programmer til at hente web sider med måske godt kan glemme at sende den med.

Apache og Apache

Apache Group er efterhånden rigtigt mange ting. De står bag hundredevis af projekter og produkter idag.

Men de 2 som vil blive omtalt i denne artikel er:

Apache HTTPD også kaldet Apache web-server:
* kan serve statiske web content (HTML, billeder etc.)
* kan køre CGI scripts, PHP scripts etc.
* hjemme side http://httpd.apache.org/

Apache Tomcat også bare kaldet Tomcat:
* en JSP/servlet engine
* kan køre standalone eller med Apache HTTPD foran
* hjemme side http://jakarta.apache.org/tomcat/

Virtual host i Apache web-server

Der finde en udmærket vejledning hos Apache:

http://httpd.apache.org/docs/vhosts/index.html (Apache 1.3)
http://httpd.apache.org/docs-2.0/vhosts/ (Apache 2.0)

Læs den og forstå den.

Her er et simpelt eksempel:

conf/httpd.conf fragment:

```
NameVirtualHost *:80
<VirtualHost *:80>
    ServerAdmin webmaster@hotel.dk
    DocumentRoot "C:/WWW/www_hotel_dk"
    ServerName www.hotel.dk
    ErrorLog logs/www_hotel_dk-error.log
    CustomLog logs/www_hotel_dk-access.log common
</VirtualHost>
<VirtualHost *:80>
    ServerAdmin webmaster@abc.dk
    DocumentRoot "C:/WWW/www_abc_dk"
    ServerName www.abc.dk
    ErrorLog logs/www_abc_dk-error.log
    CustomLog logs/www_abc_dk-access.log common
</VirtualHost>
<VirtualHost *:80>
    ServerAdmin webmaster@xyz.dk
    DocumentRoot "C:/WWW/www_xyz_dk"
```

```
    ServerName www.xyz.dk
    ErrorLog logs/www_xyz_dk-error.log
    CustomLog logs/www_xyz_dk-access.log common
</VirtualHost>
```

http://www.abc.dk/index.html vil nu ramme C:\WWW\www_abc_dk\index.html

[eksemplet bruger Windows fil system syntax, men det kan nemt
konverteres til Unix fil system syntax]

Vitual host konfiguration i Tomcat

Der findes en udmærket vejledning hos Apache:

http://jakarta.apache.org/tomcat/tomcat-5.5-doc/config/host.html

Læs den og forstå den.

Her er et simpelt eksempel:

conf/server.xml fragment:

```
    <Host name="www.foobar.dk" appBase="C:/WWW/webapps_www_hotel_dk"/>
    <Host name="www.foobar.dk" appBase="C:/WWW/webapps_www_abc_dk"/>
    <Host name="www.foobar.dk" appBase="C:/WWW/webapps_www_xyz_dk"/>
```

conf/Catalina/www.hotel.dk/ROOT.xml:

```
<Context path="/"/>
```

conf/Catalina/www.abc.dk/ROOT.xml:

```
<Context path="/"/>
```

conf/Catalina/www.xyz.dk/ROOT.xml:

```
<Context path="/"/>
```

http://www.abc.dk:8080/index.jsp vil nu ramme C:\WWW\webapps_www_abc_dk\ROOT\index.jsp

[eksemplet bruger Windows fil system syntax, men det kan nemt
konverteres til Unix fil system syntax]

Hvorfor web-server foran Tomcat

Tomcat kan som sagt bruges både standalone og med web-server foran.

Standalone:

browser----(HTTP)---->port 8080/Tomcat
browser----(HTTPS)---->port 8443/Tomcat

Med web-server foran:

browser----(HTTP)---->port 80/web-server----(AJP)---->port 8009/Tomcat
browser----(HTTPS)---->port 443/web-server----(AJP)---->port 8009/Tomcat

Det er næsten altid Apache web-server man bruger til at sætte foran, men
IIS er også supporteret.

Der kan være flere grunde til at sætte en web server foran:
* man har en masse statisk content som der ikke er grund til at
  smide ind i en java web applikation på Tomcat
* man skal bruge andre server side teknologier som f.eks. PHP der
  kræver en web-server
* man skal bruge en hardware accelerator til HTTPS som kun understøttes
  af Apache og/eller IIS
* man skal bruge en web-server til at load balance flere Tomcat servere

Der findes flere connectorer bl.a.:
  mod_jk
  mod_jk2
til Apache web-server som er dem jeg vil beskrive.

Hvordan med mod_jk

Man henter mod_jk her:

http://jakarta.apache.org/site/downloads/downloads_tomcat-connectors.cgi

Og sætter det op som:

conf/httpd.conf fragment:

```
Include conf/mod_jk.conf
```

conf/mod_jk.conf:

```
LoadModule jk_module modules/mod_jk.so
JkWorkersFile conf/workers.properties
JkLogFile  logs/jk.log
JkMount /test/* ajp13
```

workers.properties:

```
workers.tomcat_home=/jakarta/tomcat-5.5.7
workers.java_home=/SUNJava/jdk1.4.2_02
ps=/
worker.list=ajp13
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
```

Ovenstående antager at:
- mod_jk.so er i modules
- at det er URL'er som matcher /test/* der skal forwardes til Tomcat

Desværre er dokumentation for mod_jk ikke særligt god, da den fokuserer meget på den avancerede brug.

Hvordan med mod_jk2

Man henter mod_jk2 her:

http://jakarta.apache.org/site/downloads/downloads_tomcat-connectors.cgi

Og sætter det op som:

conf/httpd.conf fragment:

```
Include conf/mod_jk2.conf
```

conf/mod_jk2.conf:

```
LoadModule jk2_module modules/mod_jk2.so
JkSet config.file conf/workers2.properties
```

workers2.properties:

```
[channel.socket:localhost:8009]
tomcatId=localhost:8009

[uri:/test/*]
```

Ovenstående antager at:
- mod_jk2.so er i modules
- at det er URL'er som matcher /test/* der skal forwardes til Tomcat

Desværre er dokumentation for mod_jk2 ikke særligt god, da den fokuserer meget på den avancerede brug.

Bemærk at på trods af at mod_jk2 er nyere end mod_jk, så er det faktisk mod_jk2 som er deprecated.

Kombineret

Hvis man kører virtual host i en konfiguration med Apache web-server foran Tomcat så:
- konfigurerer man virtual host i Apache web-server
- tester at det virker på port 80
- konfigurerer man virtual host i Tomcat
- tester at det virker på port 8080
- konfigurerer den virtual host i Apache web-server til at forwarde til Tomcat
- tester at det virker på port 80

For mod_jk består det af at have en eller flere JkMount inden i <VirtualHost *:80></VirtualHost>.

Husk at det er meget vigtigt af hensyn til sikkerheden ikke at lade web-server og Tomcat dele directory struktur.

Kommentar af simonvalter d. 17. apr 2005 | 1

En guide jeg godt kunne have brugt for et par år siden. Mod_jk's dokumemtation er bestemt ikke begyndervenlig og i denne artikel står lige precis det man skal bruge for at komme igang.

Kommentar af wicez (nedlagt brugerprofil) d. 17. apr 2005 | 2

God og velskrevet artikel, hjalp mig til større forståelse for Tomcat og Apache.

Kommentar af hyberpreprocessor d. 17. apr 2005 | 3

nice, andet kan man ikke sige :) En artikel både værdifuld for Apache brugere, og Tomcat brugere (også selvom man ikke kender til Tomcat).