**AJP(Apache JServer Protocol)**

AJP는 웹서버 뒤에 있는 어플리케이션 서버로부터 웹서버로 들어오늘 요청을 위임할 수 있는 바이너리 프로토콜이다.

어플리케이션 서버로 핑을 할 수 있는 웹서버의 모니터링 기능을 지원한다.

웹 개발자들은 대체로 AJP를 여러 웹서버로 부터 여러개 어플리케이션 서버로의 로드 밸런스 구현에 이용한다. 세션들의 각각의 어플리케이션 서버 인스턴스의 이름을 갖는 라우팅 메카니즘을 사용하는 현재 어플리케이션 서버로 리다이렉트된다. 이 경우 어플리케이션 서버를 위한 리버스 프록시로 웹서버는 동작한다.

AJP는 mod\_jk를 사용하는 Apache HTTP Server 1.x 와 Proxy AJP를 사용하는 Apache 2.x 와 mod\_proxy와 proxy 밸런써 모듈에서 같이 실행된다. 아직 릴리즈 되지 않은 lighttpd 1.5 버전과 ngix, grizzly 2.1 그리고 iis에도 사용할 수 있다. 또 AJP는 Jetty 서블릿 컨테이너로 동작하는 Apache Tomcat servlet 컨테이너에도 사용할 수 있다.

https://en.wikipedia.org/wiki/Apache\_JServ\_Protocol

(아파치 웨서버와 톰캣 연동하기)

<https://www.lesstif.com/pages/viewpage.action?pageId=12943367>

**Configure mod\_jk for Apache 2.2 and Tomcat 7**

<http://jianmingli.com/wp/?p=9656>

아파치 톰캣 연동

<http://joont.tistory.com/55>

Apache httpd web server 와 tomcat 을 연계하는 방법은 세 가지가 있다.

1. 예전부터 많이 쓰던 방법은 [tomcat connector](http://tomcat.apache.org/connectors-doc)(mod\_jk)를 사용하는 방법
2. mod\_proxy를 사용하여 reverse proxy 기능을 사용하는 방법
3. mod\_proxy\_ajp 를 사용하여 AJP Protocol을 reverse proxy 로 사용하는 방법

mod\_proxy 가 mod\_jk 에 비해 설정이 간편하고 AJP 같은 특정 WAS 의존적인 프로토콜을 사용하지 않으므로  성능이 더 좋다고 하지만 mod\_jk 가 오랫동안 써왔고 친숙해서 mod\_jk 를 많이 사용하는 편인데 세 가지 방법 모두 정리해 본다.



## Tomcat Connector mod\_jk

일단 아파치와 톰캣은 설치가 되어 있다 가정

1. <http://tomcat.apache.org/download-connectors.cgi> 에서 소스  download (현재 1.2.37)
   1. wget [http://ftp.daum.net/apache//tomcat/tomcat-connectors/jk/tomcat-connectors-1.2.37-src.tar.gz](http://ftp.daum.net/apache/tomcat/tomcat-connectors/jk/tomcat-connectors-1.2.37-src.tar.gz)
2. tar zxvf tomcat-connectors-1.2.37-src.tar.gz
3. cd tomcat-connectors-1.2.37-src/native
4. ./configure --with-apxs=/usr/sbin/apxs
5. make
6. make install

/etc/local 에서 tomcat connector다운받아서 풀고 함

rpm으로 httd를 설치하면 /apxs 는 /usr/sbin아래에 있음

make install 후 /etc/httpd/modules/**mod\_jk.so** 에 복사가 된다.

Apache httpd 설정

cd /etc/httpd

아래 conf 아래 / httpd.conf 기존에 있고 workers.properties 추가

(httpd.conf파일)

**LoadModule jk\_moudle modules/mod\_jk.so**

modules디렉토리에 mod\_jk.so 모듈을 로드하겠다

(workers.properties)

worker.list=worker1

worker.worker1.type=ajp13

worker.worker1.host=localhost

worker.worker1.port=8009

(설명)

worker.list는 사용할 worker(톰캣) 인스턴스들의 리스트 (이름으로 임의로 지정해 주면 된다.)

worker.worker이름.속성=값 의 형태로 정의 해 주면 된다.

ajp13 프로토콜을 사용하며, host는 localhost이며, 포트는 8009를 사용한다.

(톰캣을 여러 개 사용할 경우)

worker.list=worker1,worker2 // 이름은 임의로 설정

worker.worker1.type=ajp13

worker.worker1.host=localhost

worker.worker1.port=8009

#worker.worker1.lbfactor=1

worker.worker2.type=ajp13

worker.worker2.host=localhost

worker.worker2.port=8010 // 포트만 바꿔주시면 됩니다. 사용할 두번째 톰캣의 server.xml 파일을 수정하셔야 됩니다.(위에서 보여드린 부분)

#worker.worker2.lbfactor=2

추가된 lbfactor는 작업할당량 비율이다.

worker1:worker2 = 1:2 의 비율로 작업을 처리한다고 보면 됨

* 이제 연동할 톰캣의 정보를 가진 properties파일을 생성하였으니, 아파치 실행 시 참조하는 httpd.conf파일에 이를 명시해 주자

**(httpd.conf)**

LoadModule jk\_module modules/mod\_jk.so    // 모듈 위치

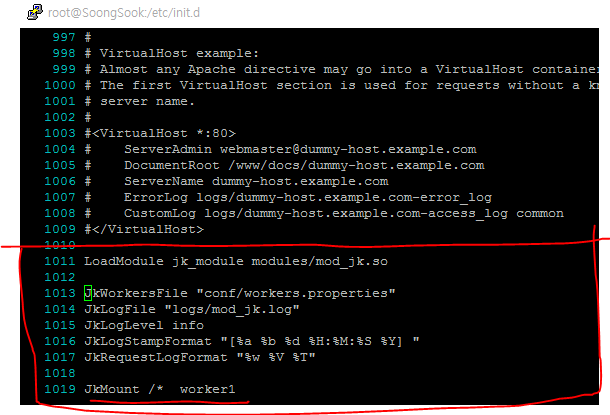
JkWorkersFile conf/workers.properties    // workers.properties 파일 위치

JkLogFile logs/mod\_jk.log    // 로그파일 위치

JkLogLevel info    // 로그 레벨

JkMount /\* worker1

* LoadModule 구문 추가하여 아까 추가한 mode\_jk.so팡리의 위치를 명시
* JkWorkerFile 구문으로 worker 프로퍼티 파일 위치 적어줌
* log 관련 부분은 로그파일의 위치와 로그레벨을 명시(로그 형태 지정도 가능)
* 마지막 중요한 JkMount!
* /\*는 모든 url 요청을 의미
* 즉, 모든 url의 요청에서 서블릿 관련 처리가 필요하다면 workers.properties 파일에 명시된 worker1로 넘기겠다는 의미
* /\* 외에도 /\*.jsp, /\*.php 등등 적어주실 수 있습니다
* JkMount /\*.do worker1                               #해당 확장자를 처리한 worker를 지정합니다
* JkMount /\*.jsp worker1                               #해당 확장자를 처리한 worker를 지정합니다
* JkMount /servlet/\* worker1                          #해당 확장자를 처리한 worker를 지정합니다



tomcat은 server1, sever2 두대에 설치되어 있고 Connector Port는 8009라 가정하자

1. Tomcat의 AJP Connector가 listen하는 Port는 tomcat의 conf/server.xml에서 확인 가능 하다.

  
tomcat 은 기본 URIEncoding 이 ISO-8859-1 이므로 한글이 깨지므로 모든 커넥터 설정에 URIEncoding="UTF-8" 을 추가해야 한다.

worker.list=worker1, worker2

worker.worker1.port=8009

worker.worker1.host=server1

worker.worker1.type=ajp13

worker.worker1.lbfactor=1

## server 2

worker.worker2.port=8009

worker.worker2.host=server2

worker.worker2.type=ajp13

worker.worker2.lbfactor=1

## Mapping the URI /service1 under worker1

/service1/\*.do=worker1

/service1/\*.jsp=worker1

# /service2 요청으로 들어온 것은 worker2 로 mount

/service2/\*=worker2

# png와 jpg 는 apache 가 처리

!/service2/\*.png=worker2

!/service2/\*.jpg=worker2

<https://www.lesstif.com/pages/viewpage.action?pageId=12943367> 좀 다른 방법

포워드 프록시와 리버스 프록시 차이

https://www.lesstif.com/pages/viewpage.action?pageId=21430345

문서

<http://tomcat.apache.org/connectors-doc/index.html>

아파치

conf : /etc/httpd/conf

홈 : /var/www/html

로그 : /etc/httpd/logs

pid-file=/etc/httpd/run/httpd.pid

실행파일들은

/etc/init.d/httpd

/sbin/service httpd

/usr/sbin/apachectl