

## 로드 밸런싱 Active-Active 방식의 웹 서버 이중화 구축 및 결함내성 시험

최재원\*

### Implementation and Fault-tolerance Tests of Load Balanced and Duplicated Active-Active Web Servers

Jae-Won Choi\*

School of Computer Science and Engineering, Kyungsung University, Busan 608-736, Korea

#### 요 약

본 논문에서는 Active-Active 방식의 웹서버 이중화 방법에 대해 연구하였다. Rsync과 crontab을 이용하여 웹서버 간에 주기적인 복사가 이루어지도록 하여 동일한 상태를 유지시켰다. 로드밸런싱서버가 웹서버를 교대로 실행하여 부하를 분산하고, 빠른 서비스가 제공되도록 하였다. 또한 임의의 웹서버에 장애가 발생하더라도 또 다른 웹서버에 의해 서비스가 계속 유지되도록 하였다.

#### ABSTRACT

In this paper we researched on the Duplication Techniques for Active-Active Web Servers. Rsync and crontab utilities make copy periodically between web servers and maintain the same status. Load Balancing Server makes web servers load balanced and fast servicing by executing web servers alternatively. Even though one web server stops due to some critical errors, the remaining web server can take over and provide services continuously.

**키워드** : 웹서버 이중화, Active-Active 이중화, 이중화, 결함내성, 가상 서버

**Key word** : Active-Active Duplication, Duplex, Fault Tolerance, Virtual Server, Web Server Duplication

접수일자 : 2013. 09. 13 심사완료일자 : 2013. 12. 30 게재확정일자 : 2014. 01. 07

\* **Corresponding Author** Jae-Won Choi(E-mail:choejw@ks.ac.kr, Tel:+82-51-663-4786)

School of Computer Science and Engineering, Kyungsung University, Busan 608-736, Korea

**Open Access** <http://dx.doi.org/10.6109/jkiice.2014.18.1.63>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

웹 서버는 오늘날의 정보통신과 정보인프라의 중심에 있으며, DNS 서버와 더불어 인터넷 서비스를 위한 핵심 서버이므로 이의 신뢰성과 무결성 향상을 위한 연구는 중요하다. 시스템의 신뢰성 향상을 위한 방법에는 여러 가지가 있지만 오류발생을 사전에 완전히 배제하기란 거의 불가능하므로 주로 오류가 발생했을 때 이에 대처하는 결함내성(fault tolerance)의 방법으로 시스템의 신뢰성을 향상시킨다[1].

일반적으로 시스템의 결함내성을 위해서는 시스템의 중요 요소를 중복시켜 하나의 요소에 오류가 발생하더라도 중복된 다른 요소가 이를 대체하여 서비스를 계속 유지시키는 이중화 방식을 채택한다. 이중화(Duplication, Duplex) 방식은 어느 한 쪽만이 동작하다가 장애 발생 시 다른 한 쪽이 동작을 이어가는데, Active-Standby 이중화와 Active-Active 이중화 방식이 있다[2].

Active-Standby 이중화는 Master-Slave 형태의 이중화로 정상시에는 1차 웹서버가 메인 서버로서 서비스를 제공하고, 2차 웹서버는 1차 서버의 장애를 대비한 보조 서버이다. 2차 웹서버는 정상시에 메인 서버로부터 주기적으로 정보를 받아 1차 웹서버와 동일한 상태를 유지시킨다. 기존의 Master-Slave 형태의 이중화 방식은 master와 slave의 설정이 별개이기 때문에 문제 발생 시 서비스 복구가 지연될 수 있고, 파일들의 동기화 지연으로 인한 문제가 잦다. 그래서 1, 2차 웹 서버를 동일하게 구성하고 디렉토리 경로 내의 모든 파일을 rsync로 동기화시키는 Active-Active 방식의 이중화가 더 적절하다[3].

이에 본 논문에는 고속의 채널을 통해 동작계열과 대기계열이 완전 동일상태를 유지하고 문제 발생 시 빠른 계열변경(fail-over)이 가능하고, 빠른 서비스 응답과 로드 밸런싱을 지원하는 Active-Active 방식의 웹서버 이중화 방법으로 채택하였다. 1, 2차 웹서버는 항상 동일상태를 유지하며 실행 가능한 상태에 있으며, 임의의 웹서버에 장애가 발생하더라도 또 다른 웹서버에 의해 서비스가 계속 유지되어 사용자는 웹서버 장애를 전혀 인식하지 못한 상태에서 계속 서비스를 받을 수 있게 한다.

Active-Active 방식의 이중화는 클라이언트의 서비스 요청 시에 서비스 요청을 받아들이는 로드 밸런싱

서버가 다이렉트 라우팅에 의해 1, 2차 웹서버를 교대로 실행하여 부하를 분산하고, 빠른 서비스 응답이 이루어진다. 이는 Master 형태로 동작하는 2대의 웹서버를 네트워크 적으로 분리된 곳에 분산 배치하여 실행한다. 웹서버 이중화를 위한 기존 Active-Standby 방식과 본 연구의 Active-Active 방식의 비교는 표 1과 같다.

본 연구에서는 웹서버의 이중화를 위해 리눅스 가상서버를 이용하였고, 이중화시스템은 로드밸런싱서버, 웹서버1, 웹서버2로 구성된다. 웹서버를 웹서버1과 웹서버2로 이중화하고, 로드밸런싱서버는 사용자의 서비스 요청을 받아들여 라운드로빈 방식으로 스케줄링하여 웹서버1과 웹서버2가 교대로 실행되도록 한다.

Active-Active 방식의 웹서버 이중화를 위해 Rsync와 crontab을 이용하여 웹서버 간에 주기적인 복사가 이루어지도록 하여 동일한 상태를 유지시켰다. 정상시에는 두 대의 웹서버가 동시에 Active 상태로 부하를 나누어 교대로 실행하고, 두 대의 웹서버 중 임의의 웹서버에 장애가 생겨 서비스가 어렵게 되면 나머지 웹서버가 서비스를 유지하여 이용자는 웹서버 장애를 전혀 인식하지 못한 상태에서 계속 서비스를 받을 수 있는지를 시험하였다.

표 1. 웹 서버 이중화 방식의 비교

Table. 1 Comparison of Web Server Duplex Methods

구분 \ 방식	Active-Standby	Active-Active
동작방식	Master-Slaver 형태 장애 대비 보조 서버	동시에 교대로 실행
전체지연	Takeover 지연	Takeover 지연 없음
부하분산	부하 집중	부하 분산
응답시간	서비스 응답 지연	서비스 응답 빠름
동기화	Tar 방식, 동기화 지연	Rsync 방식, 동기화 지연 없음
연속성	서비스 연속성 보통	서비스 연속성 좋음

## II. 웹 서버 이중화 구축

본 장에서는 웹서버의 신뢰성을 높이기 위한 웹서버 이중화(Duplication) 구축방법에 관해 연구하였다. 리눅스 가상서버를 이용하여 웹서버를 Active-Active 방식으로 이중화하여 서버의 부하를 분산하고, 서버의 신뢰성을 높이기 위한 방법을 연구하였다.

## 2.1. 웹 서버 이중화 방법

본 논문에서는 웹서버 이중화를 위해 리눅스 가상 서버(LVS: Linux Virtual Server)를 이용하였고, 웹서버를 Active-Active 방식으로 이중화하였다.

### 1) 웹 서버 이중화 방법

본 논문에서는 웹서버 이중화를 위해 리눅스 가상 서버(LVS: Linux Virtual Server)를 이용하였다. 이는 적은 비용으로 효과적으로 웹서버의 신뢰성을 높일 수 있는 이중화 방법이다[4].

리눅스 가상 서버란, 현대의 서버로 증가하는 인터넷 사용자를 서비스하기가 힘들어 지면서 고가용성 서버를 구축하기 위해 리눅스 머신을 로드 밸런싱 해주는 운영시스템이다. 만약 하나의 노드에서 처리량이 너무 많아 서비스가 어려울 경우 간단히 하나의 노드를 병렬로 추가하여 부하를 분산 하는 것을 말한다.

본 논문에서는 리눅스 가상서버를 이용하여 웹 서버를 이중화하여 서버의 부하를 분산하고, 응답시간을 줄이고, 서버의 신뢰성을 높이도록 하였다. 사용자의 서비스 요청을 로드밸런싱 서버가 받아들여 이중화된 웹서버로 다이렉트 라우팅하여 라운드-로빈(round-robin) 방식으로 교대로 서비스하도록 하였다.

다이렉트 라우팅 라운드-로빈 방식은 로드밸런싱서버로 들어오는 요청 패킷들을 차례대로 실제 서버에 할당하는 방식이다.

### 2) 이중화 시스템 구성

웹 서버를 RS1과 RS2로 이중화하고, 로드밸런싱 서버는 사용자의 서비스 요청을 받아들여 라운드로빈 방식으로 스케줄링하여 RS1과 RS2가 교대로 서비스되어지도록 한다.

로드밸런싱 서버와 실제 웹 서버에 가상 IP 주소를 부여하여 서로 공유하도록 하고, 클라이언트가 가상 IP로 서비스를 요청하면 로드밸런싱 서버가 실제 웹 서버로 다이렉트 라우팅하여 라운드로빈 방식으로 실행되도록 한다.

웹서버 이중화 시스템의 구조는 그림 1과 같다. 웹서버 이중화시스템은 로드밸런싱서버(LBS), 웹서버1(RS1), 웹서버2(RS2)로 구성된다. 시험환경을 위해 각 서버에는 리눅스 OS(kernel-2.6.18 -238.el5)를 설치하고, RS1 (121.174.8.100), RS2 (121.174.8.101), LBS(121.174.8.

102) IP 주소를 할당하고, LBS는 가상 IP(121.174.8.103)를 추가로 할당하였다.

그리고 DNS 주소변환을 위해 RS1은 1차 네임서버, RS2는 2차 네임서버 환경을 구축하여 네임서버를 이중화하였다.

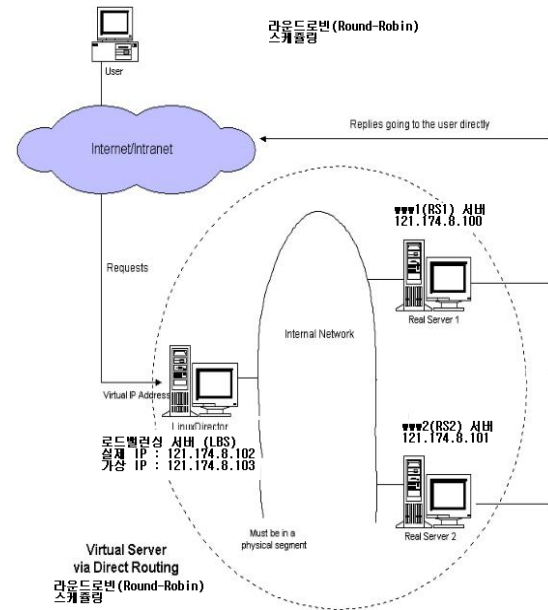


그림 1. 웹서버의 이중화 시스템 구조  
Fig. 1 Architecture of the Duplicated Web Servers

## 2.2. 웹 서버 이중화 구축

웹 서버 이중화를 위한 로드밸런싱 서버의 구축과 웹 서버 구축에 대해 연구하였다. 로드밸런싱 서버는 클라이언트의 접속요청을 웹서버로 분배하여 부하를 분산하고, 웹서버의 이중화로 하나의 웹서버에 장애가 발생하더라도 나머지 웹서버에 의해 지속적인 서비스가 제공된다.

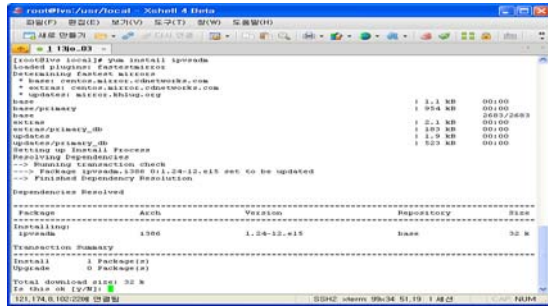
### 2.2.1. 로드밸런싱 서버 구축

로드밸런싱 서버의 설치와 환경 설정은 다음과 같다.

#### 1) 로드밸런싱 서버 설치

로드밸런싱 서버에 리눅스 가상서버 관리를 위한 유틸리티 ipvsadm을 다운로드한 후 설치한다.

```
# yum install ipvsadm
```

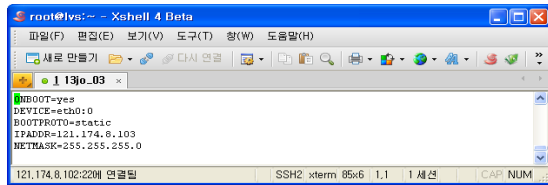


물리 LAN 카드에 IP 121.174.8.102를 부여하고, 가상 IP 121.174.8.103을 할당한다.

```
# ifconfig eth0:0 121.174.8.103 netmask 255.255.255.0 up// eth0는 121.174.8.102
```

네트워크 재시작 시 자동 설정이 되도록 ifcfg-eth0:0 파일 생성한 후, 아래와 같이 내용을 입력한다.

```
#vi /etc/sysconfig/network-scripts/ifcfg-eth0:0
```



## 2) 로드밸런싱 서버 환경설정

/etc/sysctl.conf 파일의 환경변수를 설정한 후, 그림 2와 같이 ipvsadm 명령을 실행하여 로드밸런싱 서버의 실행환경을 설정한다. 스케줄링 방식은 라운드로빈으로 하고, 웹서버의 IP를 스케줄링 리스트에 추가한다.

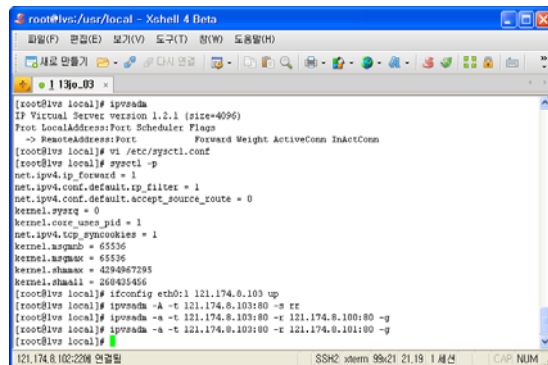


그림 2. 로드밸런싱 서버 실행환경 설정  
Fig. 2 Setup for LBS Execution Environment

```
# vi /etc/sysctl.conf
```

```
-----  
net.ipv4.ip_forward = 1 //0을 1로 변경  
-----
```

```
# sysctl -p // sysctl.conf 환경 적용
```

```
/* ipvsadm options : -A 서비스추가, -a 서버추가,  
-t 대상, -r 서버주소, -g 다이렉트라우팅,  
-w 가중치, -s 스케줄링 (rr, wrr, lc, wlc)*/
```

```
# ipvsadm -A -t 121.174.8.103:80 -s rr  
// 라운드로빈(rr) 방식 웹서버 교대로 실행
```

```
# ipvsadm -a -t 121.174.8.103:80  
-r 121.174.8.100:80 -g // 웹서버1 추가
```

```
# ipvsadm -a -t 121.174.8.103:80  
-r 121.174.8.101:80 -g // 웹서버2 추가
```

## 3) 로드밸런싱 서버 실행

로드밸런싱 서버의 설치 및 환경설정이 끝나면 아래 명령으로 로드밸런싱 서버를 실행한다.

```
# /etc/rc.d/init.d/network restart
```

## 2.2.2. 웹 서버 이중화 구축

웹서버의 이중화 구축을 위해 먼저 웹서버에 아파치 설치와 환경설정을 한 후, DNS 주소변환을 위한 서버 설정과 가상호스트 설정으로 도메인주소(www.maze76.com)로 이중화 웹서버를 교대로 접근 가능하도록 하였다.

Active-Active 방식의 이중화는 구조상 두 서버는 IP 외에는 차이가 없으므로 웹서버2는 기본적으로 웹서버1의 프로그램 설치와 환경설정과 동일하다. 웹서버1의 IP는 121.174.8.100이고, 웹서버2의 IP는 121.174.8.101이다.

## 1) 아파치 웹프로그램 설치

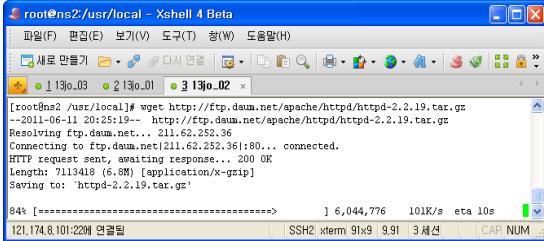
웹서버1과 웹서버2에 아파치 웹프로그램을 다운로드 받아 압축을 풀고, 아래 명령을 따라 컴파일하고 설치한다.

```
# wget http://ftp.daum.net/apache/httpd/httpd-2.2.19.tar.gz
```

```
# tar xvzf httpd-2.2.19.tar.gz
```

```
# cd httpd-2.2.19
```

```
# ./configure --enable-so --prefix=/usr/local/apache
# make // 웹서버 컴파일
# make install // 웹서버 설치
```



웹서버 설치가 완료되면 이중화 시험을 위해 웹서버 1과 웹서버2에 각각 인덱스 페이지를 아래와 같이 만든다.

```
# vi /usr/local/apache/htdocs/index.html
```

```
<html> <!-- 웹서버1 (121.174.8.100) -->
<body><h1>It www1 works!</h1></body>
</html>

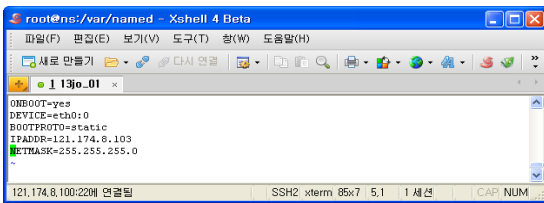
<html> <!-- 웹서버2 (121.174.8.101) -->
<body><h1>It www2 works!</h1></body>
</html>
```

그림 3. 웹 서버 인덱스 페이지 생성  
Fig. 3 Creation of Web Server's Index Page

### 2) 웹서버 네트워크 환경 설정

앞의 2.2.1절에서와 같이 웹서버1의 물리 LAN 카드에 IP 121.174.8.100을 부여하고, 가상 IP 121.174.8.103을 설정한다. 네트워크 재시작시 자동설정이 되도록 ifcfg-eth0:0 파일을 생성한다.

```
# ifconfig lo:0 121.174.8.103 netmask 255.255.255.0 up
```



웹서버2도 동일한 방식으로 IP 121.174.8.101을 부여하고, 가상 IP 121.174.8.103을 설정한 후, 자동설치 설

정을 한다.

가상 IP 설정으로 인해 발생가능한 ARP 문제 해결을 위해 ARP에 대한 응답 없음으로 추가 설정한다.

```
# vi /etc/sysctl.conf
```

```
net.ipv4.ip_forward = 1 // 0을 1로 변경
net.ipv4.conf.lo.arp_ignore = 1
net.ipv4.conf.lo.arp_announce = 2
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.ipv4.conf.default.arp_ignore = 1
net.ipv4.conf.default.arp_announce = 2
net.ipv4.conf.eth0.arp_ignore = 1
net.ipv4.conf.eth0.arp_announce = 2
```

```
# sysctl -p // sysctl.conf 설정 적용
```

### 3) DNS 네임서버 설정

도메인주소(www.maze76.com) 입력시 가상 IP(121.174.8.103)로 주소변환 되도록 DNS 서버를 그림 4와 같이 설정한다.

DNS 서버 역시 이중화하여 NS와 NS2가 1분 단위로 동기화시킴으로 NS2는 별도 설정을 하지 않아도 된다[3].

```
# vi /etc/resolv.conf
```

```
search maze76.com // 도메인 지정
domain maze76.com // 도메인 지정
nameserver 121.174.8.100 // 1차 네임서버 주소
nameserver 121.174.8.101 // 2차 네임서버 주소
```

```
# cd /var/named
```

```
# vi maze76.com.zone
```

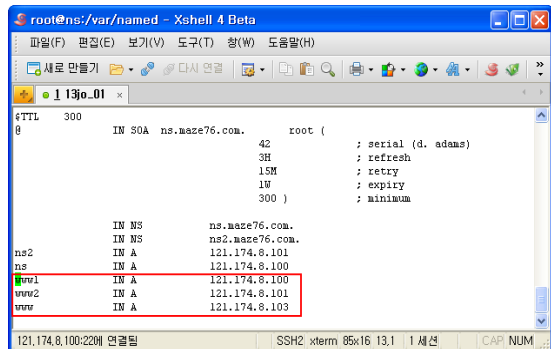


그림 4. DNS 네임서버 설정  
Fig. 4 DNS Name Server Setup

4) Virtual Host 설정

아파치 웹서버의 가상호스트(Virtualhost) 기능을 이용하여 가상 IP(121.174.8.103)로 웹페이지 접근이 가능하도록 웹서버1과 웹서버2를 그림 5와 같이 설정한다.[5-6]

```
# vi /usr/local/apache/conf/httpd.conf
```

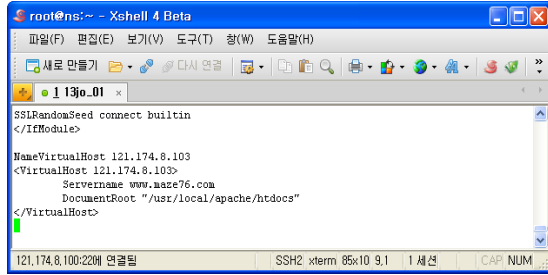


그림 5. 가상 IP의 가상호스트 설정  
Fig. 5 Virtual Host Setup to Virtual IP

5) 웹서버와 네임서버 실행

웹서버와 네임서버의 설치 및 환경설정이 끝나면 아래 명령으로 웹서버와 네임서버를 실행한다.

```
# /etc/rc.d/init.d/named restart
# /usr/local/apache/bin/apachectl restart
```

III. 웹 서버 이중화 시험

이중화의 경우에는 어떻게 양측의 데이터를 동일하게 유지시킬 것인지, 또 장애시 얼마나 빨리 계열을 변경하여 서비스의 연속성을 보장할 것인지가 중요하다. 본 장에서는 구축한 1, 2차 웹 서버 간 동기화로 매 분 주기적인 복사에 의해 동일한 상태가 유지되는지를 시험하였고, 1차 웹 서버가 오류로 인해 정지되더라도 2차 웹 서버가 대신하여 서비스를 지속할 수 있는지를 시험하였다.

3.1. 웹 서버 결함내성 시험

클라이언트가 www.maze76.com을 반복 접속했을 때 로드밸런싱 서버가 www1.maze76.com과 www2.maze76.com을 교대로 실행하여 로드 밸런싱 기능을 수행하는지를 시험하였다. 그리고 웹서버 2대 중 한 대를 중지시키더라도 살아 있는 서버로 접속이 이루어져 서

비스의 연속성이 보장되는지를 시험하였다.

1) 웹 서버 로드 밸런싱 시험

클라이언트가 로드밸런싱 서버를 거치지 않고 www1.maze76.com, www2.maze76.com을 다이렉트로 접속할 경우 각각의 웹 서버를 통해 직접 서비스되는 것을 확인하였고, ipvsadm 명령을 내렸을 때 연결설정 카운터(ActiveConn)와 연결해제카운터(InActConn)는 변화가 없었다.

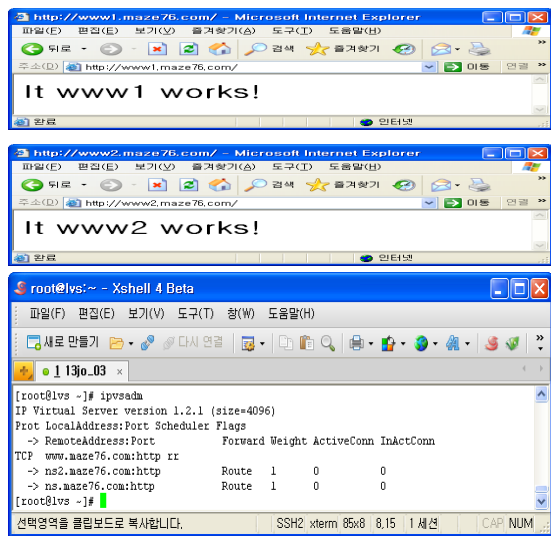
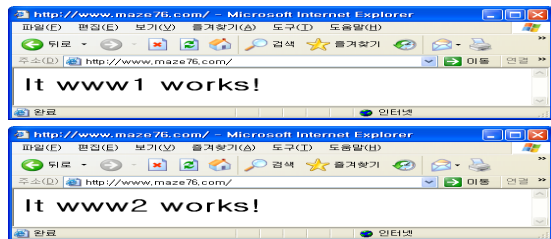


그림 6. 웹서버 직접 실행  
Fig. 6 Direct Execution of Web Servers

만반, 클라이언트가 가상 IP(www.maze76.com)로 로드밸런싱 서버에게 연속해서 접속요청을 하면 로드밸런싱 서버가 www1.maze76.com과 www2.maze76.com 두 대의 웹서버를 교대로 실행하고, ipvsadm 명령을 통해서도 ActiveConn과 InActConn이 교대로 증감하는 것을 확인할 수 있었다.





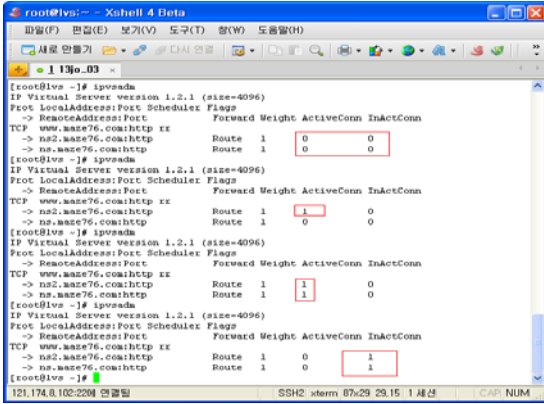


그림 7. 웹서버 로드밸런싱 시험  
Fig. 7 Test of Web Servers Load Balancing

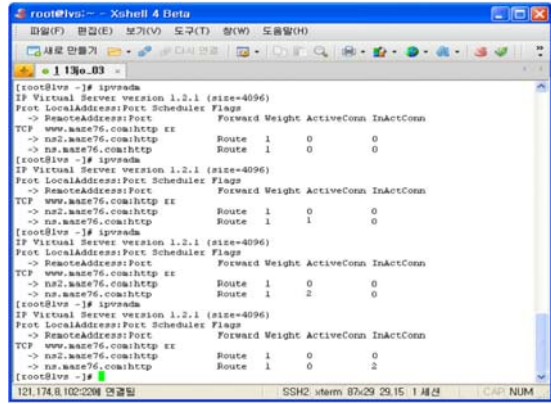


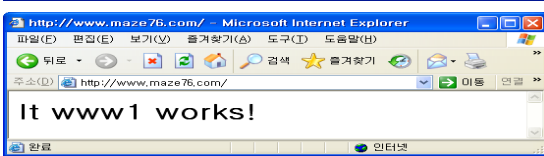
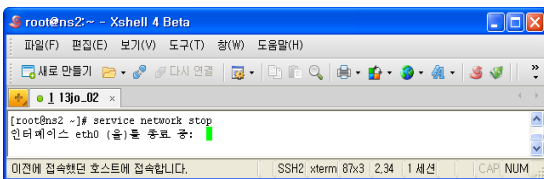
그림 8. 웹서버 결함내성 시험  
Fig. 8 Test of Web Servers Fault Tolerance

2) 웹 서버 결함내성 시험

두 대의 웹서버 중 임의의 웹서버에 장애가 생겨 서비스가 어렵게 되면 나머지 웹서버가 서비스를 유지하여 이용자는 웹서버 장애를 전혀 인지하지 못한 상태에서 계속 서비스를 받을 수 있는지를 시험하였다.

이를 위해 웹서버2를 아래와 같이 강제로 정지시킨 후, www.maze76.com을 접속한 후 F5를 눌러 새로 고침을 반복하면(웹브라우저의 캐시에서 읽어오지 않도록) www2.maze76.com으로 연결을 시도하지만 장애로 인해 응답이 없으면 www1.maze76.com으로 전환되어 실행되는 것을 확인할 수 있었다.

# service network stop // 웹서버2 강제정지



3.2. 웹서버 동기화 시험

Active-Active 방식의 웹서버 이중화를 위해 웹서버 간에 rsync와 crontab을 이용하여 주기적으로 복사하여 동일한 상태를 유지시킨다. 평상시에는 두 대의 웹서버가 동시에 Active 상태로 부하를 나누어 교대로 실행하고, 하나의 웹서버가 장애로 인해 정지되더라도 나머지 웹서버가 계속해서 서비스를 지속하게 된다.

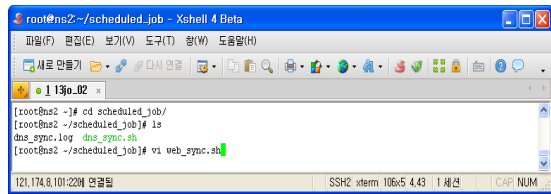
1) 웹서버 동기화 환경 설정

Rsync 프로그램을 사용하여 웹서버1과 웹서버2를 동기화시킨다. Rsync는 전체를 복사한 이후 변경된 부분만을 추가로 업데이트하기 때문에 두 서버의 파일시스템을 지속적으로 동기화시키기 위한 빠르고 효율적인 방법이다[7-8].

웹서버 간 동기화를 위한 셸프로그래밍은 그림 9와 같다. 이는 두 서버의 지정 디렉토리(/usr/local/apache/htdocs)를 비교하여 동일상태를 유지시키고, 웹서버1에서 웹페이지 삭제시 웹서버2에서도 삭제되도록 --delete 옵션을 사용한다.

# cd scheduled\_job

# vi web\_sync.sh

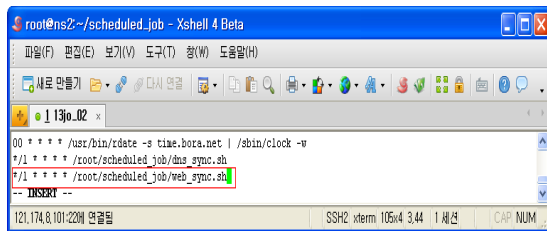


```
#!/bin/sch
# rsync start time
echo "RSYNC START Time"
  >> /root/scheduled_job/web_sync.log
echo -en `date "+%Y/%m/%d %H:%M:%S"`
  >> /root/scheduled_job/web_sync.log
# web site sync
rsync -avre ssh --delete 121.174.8.100:/usr/local/
apache/htdocs /usr/local/apache/htdocs
```

그림 9. 웹서버 동기화 셸프로그램  
Fig. 9 Shell Program for Synchronizing Web Servers

crontab 명령은 주기적으로 어떤 프로그램을 수행시키는데 사용된다. crontab 명령을 이용하여 웹서버 동기화 스크립트(web\_sync.sh)를 1 분마다 실행시켜 웹서버 간에 동일한 상태가 유지되도록 동기화하였다.

```
# chmod 700 web_sync.sh
# crontab -e
```



2) 웹서버 동기화 시험

웹서버1의 웹페이지(index.html)를 수정한 후 1분 뒤 웹서버2의 웹페이지(index.html)가 동일하게 변경된 것을 그림 10과 같이 확인하였다. 즉, 웹서버1의 웹페이지를 수정, 생성, 삭제시 웹서버2에도 동일한 웹페이지가 수정, 생성, 삭제되어 웹서버 동기화가 이루어짐을 확인할 수 있었다.

```
# vi /usr/local/apache/htdocs/index.html
```

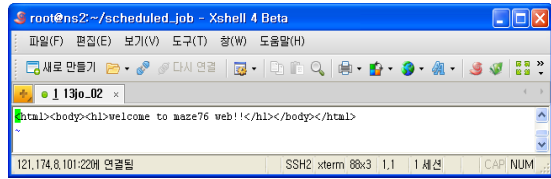
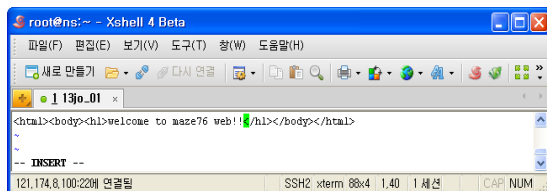


그림 10. 웹서버 간의 동기화  
Fig. 10 Synchronization between Web Servers

IV. 결론

본 논문에서는 웹서버의 신뢰성과 무결성 향상을 위한 웹서버의 이중화 구축방법에 관해 연구하였다. 웹서버를 이중화하고 리눅스 가상서버를 이용하여 로드밸런싱 하여 빠른 서비스 응답과 웹서버의 결합내성이 이루어지도록 하였다. 또한 rsync를 이용하여 웹서버 간의 빠른 동기화를 이루어 클라이언트의 접속 요청에 임의의 웹서버가 서비스할 수 있는 Active-Active 방식의 웹서버 이중화가 가능하도록 하였다.

본 연구의 웹서버 이중화시스템은 로드밸런싱서버, 웹서버1, 웹서버2로 구성된다. 이중화시스템 구축을 위해 로드밸런싱서버에 가상 IP를 할당한 후, 웹서버1과 웹서버2에도 동일한 가상 IP를 설정하였다. 가상서버 관리유틸리티 ipvsadm을 이용하여 로드밸런싱서버의 웹서버 스케줄링 방식을 라운드로빈으로 하고, 웹서버1과 웹서버2를 스케줄링 리스트에 추가하였다.

DNS 주소변환을 위한 네임서버 설정과 웹서버의 Virtual Host 설정으로 사용자가 웹서버이름(www.maze76.com)으로 접속요청 시 가상 IP로 접속요청이 이루어지고, 로드밸런싱서버는 이를 받아들여 라운드 로빈 방식으로 스케줄링하여 웹서버1과 웹서버2가 교대로 실행하는 것을 웹서버 로드밸런싱 시험을 통해 확인하였다.

또한 rsync와 crontab을 이용하여 웹서버 간에 분 단위로 주기적인 복사를 하여 동일한 상태를 유지시켜 Active-Active 방식의 웹서버 이중화를 이루었다. 웹서버1과 웹서버2는 항상 동일 상태를 유지하며 실행 가능한 상태에 있으며, 임의의 웹서버에 장애가 발생하더라도 다른 웹서버에 의해 서비스가 계속 유지되어 사용자는 웹서버 장애를 전혀 인식하지 못한 상태에서 계속 서비스를 받을 수 있는 것을 웹서버 결합내성 시험



을 통해 확인하였다.

본 논문에서는 웹서버 이중화시스템 구축을 위한 절차를 실제 예를 통해 상세히 기술함으로써 웹서버 개발자와 관리자의 신뢰성 있는 웹서버 구축에 많은 도움이 될 것이다. 본 연구에서는 웹서버의 사양이나 실제 부하 등에 대한 고려는 하지 않고 라운드로빈 방식의 웹서버 스케줄링을 하였다. 향후 이를 고려한 연구를 계속하여 좀 더 효율적이고 신뢰성 있는 웹서버 이중화시스템 구축이 이루어지도록 하겠다.

### 감사의 글

본 논문은 2013학년도 경성대학교 학술연구비 지원에 의하여 이루어진 연구로서, 관계부처에 감사드립니다.

### REFERENCES

- [1] Jae-Weon Choi, "Foundation Techniques and Cooperation Test of Fault-tolerant Domain Name Servers for Internet Name Resolution", *Journal of Korean Institute of Marine Information and Communication Sciences*, Vol. 15, No. 1, Jan. 2011.
- [2] Windows Forum, Duplication Technology [Internet]. Available: <http://windowsforum.kr/lecture/1663804>, Jan. 2011.
- [3] Jae-Weon Choi, "Foundation Techniques and Fault-tolerance Tests of Active-Active Duplicated Domain Name Servers", *Journal of Korean Institute of Information and Communication Engineering*, Vol. 17, No. 2, Feb. 2013.
- [4] Korea Hydrographic and Oceanographic Administration, "Completion Report of Maritime Integrated Information System", Ministry of Land, Transport and Maritime Affairs, 2010. (Library of Congress)
- [5] Jae-Weon Choi, "Implementation Methods of Name-based Virtual Host for Multiple Domain Name Service", *Journal of Kysungsung University Research Institute of Engineering Technology*, Vol. 16, Feb. 2010.
- [6] Jae-Weon Choi, "Implementation Methods of IP-based Virtual Host for Multiple Host Service", *Journal of Kysungsung University Research Institute of Engineering Technology*, Vol. 17, Feb. 2011.
- [7] Ja-Ryong Sea, *Fedora Linux Following*, pp.533-560, HaeGeeWon, Nov. 2008.
- [8] KT Cloud Engineering, Replication DR Implementation on uCloud [Internet]. Available:[http://14.63.247.75:8080/export/sites/default/support/pds/download/replication\\_DR\\_guide\\_v3.4.pdf](http://14.63.247.75:8080/export/sites/default/support/pds/download/replication_DR_guide_v3.4.pdf), Feb. 2013.



최재원 (Jae-Won Choi)

1988년 2월 고려대학교 컴퓨터공학과(공학사)  
 1990년 8월 미시간주립대학교 컴퓨터공학과 (공학석사)  
 1995년 8월 건국대학교 전자공학과 (공학박사)  
 1990년 10월 ~ 1997년 8월 삼성전자 정보통신연구소 선임연구원  
 1997년 9월 ~ 2014년 現 경성대학교 컴퓨터공학부 정교수  
 ※관심분야: 정보통신, 정보보안, 인터넷응용, 모바일앱