Active-Active 방식의 DNS 서버의 이중화 구축 및 결함내성 시험

최재원*

Foundation Techniques and Fault-tolerance Tests of Active-Active
Duplicated Domain Name Servers

Jae-Won Choi*

이 논문은 2012학년도 경성대학교 학술연구비지원에 의하여 연구되었음.

요 약

본 논문에서는 기존의 Active-Standby 방식의 이중화가 아닌 Active-Active 방식의 DNS (Domain Name System) 이중화에 관해 연구하였다. Active-Active 방식의 DNS 이중화를 위해 1, 2차 DNS 서버 간에는 rsync와 crontab을 이용하여 주기적으로 복사하여 동일한 상태를 유지하고, 1차 혹은 2차 DNS 서버가 장애로 인해 정지되더라도 살아 있는 DNS 서버가 계속해서 서비스를 지속하도록 하였다.

ABSTRACT

Active-Standby Duplication Techniques are conventionally used for fault-tolerant systems. But in this paper we researched on the Active-Active Duplication Techniques for Fault-tolerant DNS System. Our Active-Active Duplication made the 1st DNS periodically copied to the 2nd DNS and maintained the same status by using Rsync and Crontab. Even though the 1st or the 2nd DNS stops due to some critical errors, the remaining DNS can take over and provide continuous services.

키워드

네임서버, 결함내성, 이중화, Active-Active 이중화, Active-Standby 이중화

Key word

DNS, Domain Name System, Fault Tolerance, Duplication, Active-Active Duplication

* 종신회원 : 경성대학교 컴퓨터공학부 정교수 (choejw@ks.ac.kr)

접수일자 : 2012. 08. 11

심사완료일자 : 2012. 09. 02

Open Access <http://dx.doi.org/10.6109/jkiice.2013.17.1.90>

©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

DNS(Domain Name System)는 네트워크에서 도메인이나 호스트 이름을 숫자로 된 IP 주소로 해석해 주는 TCP/IP 네트워크 서비스이다.[1] 인터넷 상에서 IP 주소 대신에 도메인 주소를 사용하는 것은 222.122.195.6과 같은 IP 주소보다는 www.naver.com과 같은 문자 주소를 사용하는 것이 편리하기 때문이다.

DNS 서버는 웹 서버와 더불어 인터넷 서비스를 위한 최소한의 기본 서버이므로 이의 신뢰성과 무결성 향상을 위한 연구는 중요하다. 시스템의 신뢰성 향상을 위한 방법에는 여러 가지가 있지만 오류발생을 사전에 완전히 배제하기란 거의 불가능하므로 오류가 발생했을 때 이에 대처하는 결합내성(fault tolerance)의 방법으로 시스템의 신뢰성을 향상시킨다.[2]

일반적으로 시스템의 결합내성을 위해선 시스템의 중요 요소를 중복시켜 하나의 요소에 오류가 발생하더라도 중복된 다른 요소가 이를 대체하여 서비스를 계속 유지시키는 이중화 방식으로 구현한다.[3] 이중화(duplication) 방식은 어느 한 쪽만이 동작하다가 장애 발생시 다른 한 쪽이 동작을 이어가는데, Active-Standby 이중화 방식과 Active-Active 이중화 방식이 있다.

Active-Standby 이중화는 Master-Slave 형태의 이중화로 1차 DNS는 주소변환 서비스를 제공하는 메인 DNS이고, 2차 DNS는 메인 DNS의 이상을 대비하여 동작하는 보조 DNS이다. 평상시에는 1차 DNS가 주소변환 서비스를 제공하고, 2차 DNS는 주소변환을 위한 각종 정보를 1차 DNS로부터 주기적으로 받아 1차 DNS와 동일한 상태를 유지시킨다. 만약 1차 DNS에 장애가 발생하여 서비스 제공이 어렵게 되면 2차 DNS가 서비스를 유지하여 이용자는 시스템 이상을 전혀 인식하지 못한 상태에서 계속 서비스 받을 수 있게 한다.[4]

호스팅 업체나 다량의 도메인을 보유한 업체의 경우 DNS 서버 내에 설정되는 zone 파일만도 백여 개가 넘는다. 이 때 기존의 Master-Slave 형태의 이중화 방식으로는 각 개별 zone 파일마다 설정된 옵션에 맞춰 내용을 전송 받는데 걸리는 시간만도 만만치 않고 오류 발생이나 파일들의 동기화 지연으로 인한 문제가 잦다. 또 master와 slave의 설정이 별개이기 때문에 문제 발생 시 서비스 복구가 지연될 수 있다. 이런 이유로 인해 1, 2차 DNS를

동일하게 구성하고 디렉토리 경로 내의 모든 파일을 rsync로 동기화 시키는 Active-Active 방식의 이중화를 선호한다. 이에 본 논문에는 빠른 서비스 응답과 로드 밸런싱을 지원하는 Active-Active 방식의 DNS 이중화에 관해 연구하였다.

본 논문에서는 기존의 Active-Standby 방식의 이중화가 아닌 Active-Active 방식의 DNS 이중화에 관해 연구하였다. Active-Active 방식의 이중화는 Client가 지정한 1, 2차 DNS 서버의 순서에 무관하게 DNS 쿼리를 보내 응답이 빠른 DNS 서버로부터 주소변환 서비스를 받게 되는 이중화 방식으로 서비스 응답시간이 빠르고, Master 형태로 동작하는 2대의 DNS 서버를 네트워크 적으로 분리된 곳에 분산 배치하여 구동하는 방식이다.

구축한 1, 2차 DNS 서버 간에는 주기적 갱신에 의해 동일 상태가 유지되고, 1차 DNS 서버가 오류로 인해 정지하더라도 2차 DNS 서버가 대신하여 서비스를 지속할 수 있도록 하였다. 본 연구에서는 DNS 서버의 이중화 구축과 시험을 위해 도메인(maze76.com)을 신규 등록하고, 해당 도메인을 관장하는 1차 네임서버(121.174.8.100)와 2차 네임서버(121.174.8.101)로 구성하여 실제 예를 통해 리눅스 기반의 1, 2차 DNS 프로그램의 설치, DNS 서버의 환경설정 및 구축절차를 상세히 기술하였다.

II. DNS 서버의 이중화 구축

결합내성을 갖는 DNS 시스템 구축을 위해 1차 DNS와 2차 DNS로 이중화(Duplication) 한다. 본 장에서는 1, 2차 DNS를 Active-Active 방식으로 이중화하기 위한 1, 2차 DNS 서버의 프로그램 설치 및 환경설정, DNS 서버의 구동 및 동작시험에 관해 연구하였다.

2.1. 1차 DNS 서버 구축

BIND(Berkeley Internet Name Domain)는 DNS 서버 구축을 위한 패키지로 현재 9 버전이 사용되고 있다. 9 버전의 경우도 기존 8 버전의 기본환경과 동일하게 설정하더라도 정상적으로 동작한다. 기존과의 차이점은 dnssec-key를 이용해 update 부분의 보안을 강화한 점이다.

Bind 패키지는 소스 파일, rpm 패키지, yum 명령의 3가지 설치방법이 있다.[5] 본 논문에서는 프로그램의 설치가 가장 용이한 YUM 명령을 이용하여 설치하였고, 1차 DNS 서버의 IP 주소는 121.174.8.100으로 지정하였다.

2.1.1. DNS 프로그램 설치

YUM(Yellowdog Updater Modified)은 redhat 기반의 플랫폼에서 인터넷을 통한 편리한 설치와 업데이트를 가능케 해주는 프로그램이다.

아래 명령을 실행한 후 확인에 y만 입력하면 자동으로 BIND 패키지를 설치한다. 기 설치된 BIND 프로그램이 존재하더라도 yum을 이용해서 손쉽게 제거 혹은 업데이트할 수 있다.[6]

```
# yum install bind*
```

그림 1. YUM을 이용한 DNS 프로그램 설치
Fig. 1 Installation of DNS Program by using YUM

2.1.2. DNS 환경 설정

DNS 서버가 정상적으로 동작하기 위해서는 기본적인 네트워크 설정, 방화벽 및 보안 관련 설정 후, BIND의 named.conf 환경 파일을 설정해 주어야 한다.

2.1.2.1. 네트워크 기본 환경 설정

① 랜카드 설정

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
BOOTPROTO=static           // IP 수동 설정
IPADDR=121.174.8.100        // IP 주소
NETMASK=255.255.255.0       // Netmask
```

```
GATEWAY=121.174.8.1        // 게이트웨이
```

② 호스트 명 설정

```
# vi /etc/sysconfig/network
HOSTNAME=ns.maze76.com // 호스트명 지정
```

③ DNS 주소 지정

```
# vi /etc/resolv.conf
search maze76.com           // 도메인 지정
domain maze76.com           // 도메인 지정
nameserver 121.174.8.100     // 1차 DNS 주소
nameserver 121.174.8.101     // 2차 DNS 주소
```

④ 자주 사용하는 서버 주소 등록

```
# vi /etc/hosts
127.0.0.1      ns.maze76.com  ns localhost
121.174.8.100  ns.maze76.com  ns
121.174.8.101  ns2.maze76.com ns2
```

2.1.2.2. 방화벽 및 보안 관련 설정

① 방화벽 포트 열기

방화벽이 가동 중인 경우 DNS는 로컬 상에서만 정상 동작하고 외부에서는 포트가 막혀 서비스가 불가능해진다. 그래서 DNS가 사용하는 TCP와 UDP의 53 포트를 열어 주어야 한다. 즉, 방화벽이 동작 중일 때는 iptables를 중지한 후, 아래 줄을 추가한 후 재구동한다.

```
# service iptables stop // 방화벽 중지
# vi /etc/sysconfig/iptables // 아래 줄 추가
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 53 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 53 -j ACCEPT
# service iptables start // 방화벽 재구동
```

② 방화벽 사용 중지

보안 상 문제가 없어 방화벽을 완전히 내릴 경우에는 아래와 같이 실행한다.

```
# ntsysv // 실행 후 iptables 체크 해제
# service iptables stop
```


① directory "/var/named";

DNS 환경설정 파일들이 위치하는 경로를 지정한다. 이 디렉토리에 named.ca, localhost.zone, named.local, maze76.com.zone, maze76.com.rev 파일들이 존재한다. 보안상 chroot를 적용하면 /var/name/chroot/var/named 디렉토리에 환경설정 파일들이 위치하게 된다.

② 옵션 설정

```
options {
    // zone 전송 제한
    allow-transfer {127.0.0.1; 121.174.8.101;};
    // 순환 질의 제한
    allow-recursion {127.0.0.1; 121.174.8.0/24;};
};
```

설정된 zone 파일의 전송 요청이 왔을 때 전송을 허용할 IP를 지정한다. 자신이 가지고 있지 않은 도메인 정보에 대한 순환 질의를 허용할 IP를 지정한다.

③ dnssec-key / rndc 설정

```
key "maze76-key" {
    algorithm hmac-md5;
    secret "gDGzha3zfCHZXqwBuCTqLA==";
};

controls {
    inet 127.0.0.1 port 953
    allow {127.0.0.1; keys {"maze76-key"};};
};
```

Bind 9 버전부터 사용하게 된 rndc, dnssec-key를 설정한다.

```
④ zone "maze76.com" IN {
    type master;
    file "maze76.com.zone";
    allow-update {none;};
};
```

maze76.com 도메인 내에 있는 모든 컴퓨터들의 IP가 maze76.com.zone 파일에 정의되어 있고, update를 허용하지 않는다는 설정이다.

```
⑤ zone "8.174.121.in-addr.arpa" IN {
    type master;
```

```
file "8.174.121.rev";
allow-update {none;};
};
```

망주소가 121.174.8.0인 네트워크에 속해있는 모든 컴퓨터들의 IP주소를 도메인주소로 변환하기 위한 정보가 8.174.121.rev 파일에 정의되어 있음을 의미한다. 네트워크 주소부분은 거꾸로(예: 121.174.8 → 8.174.121) 적어 주어야 하며, 하나의 NIC에 연결된 네트워크 당 하나의 rev가 설정된다. 즉, zone 파일 당 한 개의 rev 파일이 설정되는 것은 아니다.

(3) /var/named/maze76.com.zone

maze76.com 도메인 내에 있는 컴퓨터들의 도메인주소에 대한 IP주소를 저장하는 파일이다.

그림 4. DNS 환경 설정 (maze76.zone)

Fig. 4 Set up for DNS Environment (maze76.zone)

① @ IN SOA ns.maze76.com. root.maze76.com.

- @ : named.conf 파일에 설정된 도메인 maze76.com을 가리킨다.

- ns.maze76.com. : 네임서버를 의미한다.

- root.maze76.com. : 관리자의 메일주소 (root@maze76.com)로 사용된다.

② () 괄호로 둘러싸인 부분

Active-Standby 방식의 이중화에 사용되는 옵션이기에 수정 없이 디폴트 값을 유지한다.

③ IN NS ns.maze76.com.

NS(Name Server)는 해당 도메인을 관장하는 네임서버를 가리킨다.

④ IN MX 10 mail.maze76.com.

MX(Mail eXchange)는 해당 도메인의 메일서버를 가리킨다.

⑤ 호스트명 IN A IP주소

A(Address)는 해당 호스트의 IP 주소를 지정한다.

(4) /var/named/8.174.121.rev

121.174.8.x 네트워크 내에 있는 컴퓨터들의 IP주소에 대한 도메인주소가 저장된 파일이다.

그림 5. DNS 환경 설정 (8.174.121.rev)

Fig. 5 Set up for DNS Environment (8.174.121.rev)

① PTR

PTR(PointeR)는 IP 주소에 대응하는 도메인 이름을 매핑시켜 준다.

② 도메인명 끝에 ‘.’ 추가

maze76.com.zone과 8.174.121.rev 파일 설정 시 도메인 이름을 표시할 때 반드시 끝부분에 ‘.’을 추가해 주어야 한다.

(5) /var/named/named.ca, localhost.zone, named.local

name.ca 파일은 루트 네임서버의 정보가 들어 있는 데이터베이스 파일이다. 만약 이 파일이 없으면 “# dig . ns > /var/named/named.ca”를 입력하면 된다. localhost.zone과 named.local 파일은 루트백 시험용 파일로서 기본 파일을 수정 없이 그대로 사용한다.

(6) 보안을 위한 기본 디렉토리 변경 및 퍼미션 설정
named.conf 설정에 의하면 maze76.com.zone과 8.174.

121.rev파일은 /var/named에 위치해야 하지만 보안상의 이유로 실제파일은 /var/named/chroot/var/named에 두고 /var/named에는 링크파일만을 둔다. 이를 위해 다음과 같이 심볼릭 링크를 생성하고 퍼미션을 설정해 준다.

ln -s [실제파일경로] [링크의위치]

```
# ln -s /var/named/chroot/etc/named.conf /etc/named.conf
```

```
# ln -s /var/named/chroot/var/named/maze76.com.zone /var/named/maze76.com.zone
```

```
# ln -s /var/named/chroot/var/named/8.174.121.rev /var/named/8.174.121.rev
```

실제 파일	심볼릭링크 파일
/var/named/chroot/var/named/maze76.com.zone	/var/named/maze76.com.zone
/var/named/chroot/var/named/8.174.121.rev	/var/named/8.174.121.rev
/var/named/chroot/etc/named.conf	/etc/named.conf

2.1.3. 1차 DNS 서버 구동 및 시험

DNS 환경설정이 끝나면 1차 DNS 서버를 구동하여 DNS 서버의 정상 동작여부를 확인한다.

2.1.3.1. DNS 서버 구동

1차 DNS의 정상 동작 여부를 확인하기 위해 resolv.conf 파일에 1차 DNS 서버의 IP를 지역 네임서버의 IP로 지정한다.

그림 6. DNS 시험 환경 설정

Fig. 6 Environment Set up for DNS Testing

DNS 서버의 구동은 아래의 명령을 이용하여 구동한다.

① 서버구동: # /etc/rc.d/init.d/named start

② 서버정지: # /etc/rc.d/init.d/named stop

③ 서버재구동: # /etc/rc.d/init.d/named restart

2.1.3.2. DNS 동작 시험

1차 DNS 서버 구축이 정상적으로 이루어져 nslookup 명령으로 도메인주소에 대응하는 IP주소의 변환을 확인할 수 있다.

그림 7. 1차 DNS 동작 시험
Fig. 7 Test of Primary DNS Working

2.2.2차 DNS 서버 구축

Active-Active 방식의 DNS 이중화를 위해 1, 2차 DNS 서버 간에는 rsync과 crontab을 이용하여 주기적으로 복사하여 동일한 상태를 유지한다. 그래서 1, 2차 DNS 서버는 동시에 Active 상태로 동작하게 되어 1차 혹은 2차 DNS 서버가 장애로 인해 정지되더라도 살아 있는 DNS 서버가 계속해서 서비스를 지속한다.

2.2.1. DNS 프로그램 설치

Active-Active 방식의 이중화는 구조상 두 서버는 IP 외에는 차이가 없으므로 2차 DNS 서버의 프로그램 설치의 앞의 2.1.1~2.1.2절에서와 같이 DNS 프로그램 설치와 기본 환경 설정은 동일하다. 1차 DNS 서버의 IP는 121.174.8.100이고, 2차 DNS 서버의 IP는 121.174.8.101로 하였다.

2.2.2. DNS 동기화 환경 설정

2차 DNS 서버의 환경 설정은 1차 DNS 서버의 zone 파일들을 복사하여 저장하면 되므로 2차 DNS 구축을 위해 추가된 환경 설정 부분만을 중점적으로 다루었다.

2.2.2.1. rsync 동기화

Rsync라는 프로그램을 사용하여 1차 DNS와 2차 DNS를 동기화시킨다. 파일시스템의 특정 부분을 원격 시스템에 복사할 때는 tar와 ssh를 사용하는 것이 이상적이다. 반면, 두 시스템 간의 파일시스템을 지속적으로 동기화

하기 위해서는 rsync가 가장 적절하다. 일반적으로 tar는 전체를 복사할 때 사용하고, rsync는 전체를 복사한 이후에 마지막으로 변경된 부분만 업데이트하는 데 쓰인다. 목적지에 파일시스템의 내용을 모두 복사할 때는 tar가 좀 더 빠르지만, 두 파일 시스템 간에 파일의 내용이 일부만 다르다면 rsync가 훨씬 빠르기 때문이다.

그러므로 DNS 운영 시 기본적인 zone 파일의 수정과 환경 설정의 변경은 1차 DNS를 기준으로 작업하며 2차 DNS는 rsync를 이용하여 두 서버의 지정 디렉토리를 비교하여 업데이트 되도록 하는 것이 좋다. 일반적으로 redhat 배포판7.1에는 rsync-2.4.6-2 버전이 기본으로 깔려있고, <http://rsync.samba.org/ftp/rsync>에서 최신판을 받아서 아래와 같이 설치하면 된다.

```
# tar xvfz rsync-2.4.6.tar.gz
# ./configure
# make
# make install
```

(1) rsync 환경 설정

rsync는 873포트 혹은 rsh나 ssh 프로토콜을 사용하여 동기화한다. 873포트를 이용한 동기화는 /etc/rsyncd.conf에 지정된 path에서만 동기화가 가능하다는 문제점이 있다. 반면, rsh 또는 ssh를 사용하는 경우에는 특정 경로에 종속되지 않아 사용이 자유롭고 키 설정에 의한 보안 강화의 효과도 있으므로 본 논문에서는 ssh를 이용하여 DNS간 동기화를 수행하였다.

① client에서 RSA Public Key 생성

server는 rsync 사용 시 원본 데이터를 제공하는 서버이고, client는 데이터를 복사해 갈 서버이다. Client에서 아래 명령을 실행하면 사용자 디렉토리에 .ssh/id_rsa.pub 파일이 생성된다.

```
# ssh-keygen -t rsa
```

② server에서 .ssh 디렉토리 생성

```
# cd /root
# mkdir .ssh
```

③ client의 Public Key를 server로 복사

.ssh/id_rsa.pub 파일을 server의 사용자 디렉토리에 .ssh/authorized_keys로 복사한다.


```
# scp /root/.ssh/id_rsa.pub root@121.174.8.100: /root/.ssh/authorized_keys
```

④ client의 /etc/ssh/sshd_config 수정

아래 항목을 확인하여 주석 처리되어 있을 경우 주석을 해제한다.

```
Protocol 2
#HostKey /etc/ssh/ssh_host_rsa_key
#ServerKeyBits 768
#PermitRootLogin yes
#RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile .ssh/authorized_keys
#HostbasedAuthentication no
PasswordAuthentication yes
#PermitEmptyPasswords no
Subsystem sftp /usr/libexec/openssh/sftp-server
```

⑤ client에서 ssh 재시작

```
# service sshd restart
```

(2) rsync의 사용 예

아래 명령으로 2차 DNS에서 1차 DNS의 특정 디렉토리 내용을 가져올 수 있다. 소스 측 경로의 마지막에 /가 있으면 디렉토리 내의 모든 내용을 복사하되 디렉토리 자체는 복사하지 않는다. 디렉토리를 포함시킬 때는 /를 생략한다.

```
# rsync -avr 121.174.8.100::chroot/ /var/named/chroot/

ssh를 통해서 rsync를 실행하려면 -e 스위치를 사용한다. -e는 execute의 약자로 rsh나 ssh같은 명령을 실행할 수 있다. 그리고 소스 측에서 제거된 내용을 복사본에서 제거하여 동일한 복사본을 유지하기 위해서는 --delete 옵션을 사용한다.

# rsync -avre ssh --delete 121.174.8.100::chroot:/var/named/chroot/
```

2.2.2.2. dns_sync.sh 스크립트

1차 DNS 서버의 /var/named/chroot와 2차 DNS 서버의 /var/named/chroot를 동기화시킨 후 BIND 설정을 재적용하기 위한 스크립트이다.

```
# vi /root/dns_sync.sh
```

```
#!/bin/sh
# dns data sync
# rsync start
echo -en 'date "+%Y/%m/%d %H:%M:%S"' >> /root/dns_sync.log
echo " RSYNC START Time" >> /root/dns_sync.log
/usr/bin/rsync -avre ssh --delete 121.174.8.100:/var/named/chroot/ /var/named/chroot/
/usr/sbin/mdc reconfig
/usr/sbin/mdc reload
```

2.2.2.3. crontab 스케줄링

보라넷의 타임 서버와 시간을 동기화시킨 후, 위의 DNS 동기화 스크립트를 1분에 한번 씩 실행되도록 crontab 명령을 이용하여 스케줄링 하였다. 즉, 주소변환을 위한 DNS 환경파일들(maze76.com.zone, 8.174.121.rev 등)을 1차 DNS 서버에서 2차 DNS 서버로 매 분 복사하여 1, 2차 DNS 간에 동일한 상태가 유지되도록 한다.

```
# vi /root/dns_sync.sh
```

```
00 * * * * /usr/bin/rdate -s time.bora.net | /sbin/clock -w
*/1 * * * * /root/dns_sync.sh
```

2.2.3. 2차 DNS 서버 구동 및 시험

2차 DNS의 정상 동작 여부를 확인하기 위해 resolv.conf 파일에 구축한 2차 DNS 서버의 IP를 지역 네임서버의 IP로 지정한 후, 2차 DNS 서버를 구동하여 정상 동작여부를 확인한다.

그림 8. DNS 시험 환경 설정
Fig. 8 Environment Set up for DNS Testing

nslookup 명령을 통해 2차 DNS 서버 구축이 정상적으로 이루어진 것을 확인할 수 있다.

그림 9. 2차 DNS 동작 시험
Fig. 9 Test of Secondary DNS Working

III. DNS 동기화 및 결함내성 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 DNS 서버 간 연동으로 매 분 주기적인 복사에 의해 동일한 상태가 유지되는지를 시험하였고, 1차 DNS 서버가 오류로 인해 정지하더라도 2차 DNS 서버가 대신하여 서비스를 지속할 수 있는지를 시험하였다.

3.1. DNS 동기화 시험

2차 DNS 서버는 rsync를 통해 지정된 디렉토리와 1차 DNS 서버의 지정 디렉토리를 비교하여 파일이 생성되거나 수정되었는지, 또는 삭제되었는지를 1분 단위로 비교해서 1차 DNS 서버와 동일한 상태를 유지시킨다.

3.1.1. 파일수정 시험

1차 DNS 서버의 /var/named/chroot/etc/ named.conf 파일을 열고 “test용 주석 생성”이란 주석을 달아 수정한다.

잠시 후 2차 DNS 서버의 /var/named/chroot/etc/named.conf 파일을 확인했을 때 1차와 똑같이 수정된 것을 확인할 수 있었다.

그림 10. 파일 수정 시험
Fig. 10 Test of File Update

3.1.2. 파일생성 시험

1차 DNS 서버의 /var/named/chroot/var/named 디렉토리에서 test.com.zone 파일을 생성한다. 잠시 후 2차 DNS 서버의 /var/named/chroot/var/named 디렉토리에도 test.com.zone 파일이 생성된 것을 확인할 수 있었다.

그림 11. 파일 생성 시험
Fig. 11 Test of File Creation

3.1.3. 파일삭제 시험

2차 테스트에서 생성된 test.com.zone 파일을 삭제해 본다. 잠시 후 2차 DNS 서버에서도 해당 파일이 삭제된 것을 확인할 수 있었다.

그림 12. 파일 삭제 시험
Fig. 12 Test of File Removal

3.2. DNS 결합내성 시험

1차 DNS 서버에 시스템 오류가 발생하여 서비스 제공이 어렵게 되면 2차 DNS 서버가 서비스를 유지하여 이용자는 시스템의 이상을 전혀 인식하지 못한 상태에서 계속 서비스를 받을 수 있는지를 시험하였다.

3.2.1. DNS 동작시험

DNS의 동작시험을 위해 1차 DNS 서버를 기본 DNS로 2차 DNS 서버를 보조 DNS로 설정한 후, 임의의 클라이언트에서 nslookup 명령으로 서버의 도메인 이름을 입력하면 서버의 IP주소로 변환이 1차 DNS 서버(121.174.8.100)에 의해 이루어짐을 확인할 수 있었다.

그림 13. 2차 DNS 동작 시험
Fig. 13 Test of Secondary DNS Working

3.2.2. DNS 결합내성 시험

1차 DNS 서버(121.174.8.100)를 강제로 정지시킨 후, 임의의 클라이언트에서 nslookup 명령으로 서버의 도메인 이름을 입력하면, 2차 DNS 서버(121.174.8.101)로부터 DNS 정보를 가져오는 것을 확인할 수 있었다.

그림 14. DNS 결합내성 시험
Fig. 14 Test of Fault Tolerance of DNS

IV. 결 론

본 논문에서는 도메인주소와 IP주소의 변환 서비스를 제공하는 DNS 서버의 구축과 이의 신뢰성 및 무결성 향상을 위한 연구를 하였다. DNS 서버를 Active-Active 방식으로 이중화하여 응답이 빠른 DNS 서버에서 주소변환 서비스를 제공하고, 1차 또는 2차 DNS 서버가 장애로 서비스 제공이 어렵게 되면 남은 DNS 서버가 서비스를 계속하여 이용자는 시스템 이상을 전혀 인식하지 못한 상태에서 계속 서비스를 받을 수 있도록 하는 결합내성을 갖는 DNS 시스템 구축에 관해 연구하였다.

DNS 서버의 이중화를 위해 1, 2차 DNS의 등록, YUM을 이용한 DNS 프로그램의 설치, DNS 동작 시 참조되는 named.conf 파일과 해당 도메인 내 서버들의 도메인주소와 IP주소를 가지는 zone 파일과 rev 파일 등 1, 2차 DNS의 환경설정에 관해 연구하였다. 그리고 Active-Active 방식의 DNS 이중화를 위해 1, 2차 DNS 서버 간에는

rsync와 crontab을 이용하여 주기적으로 복사하여 동일한 상태를 유지하고, 1차 혹은 2차 DNS 서버가 장애로 인해 정지되더라도 살아 있는 DNS 서버가 계속해서 서비스를 지속하도록 하였다.

구축한 1, 2차 DNS의 연동시험을 통해 2차 DNS는 crontab에 등록된 스케줄에 따라 1분 단위로 dns_sync.sh 스크립트에 지정된 내용대로 1차 DNS의 지정 디렉토리 내에 있는 모든 파일을 2차 DNS와 비교 후 수정, 등록, 삭제된 내용에 대해 동기화함으로써 1, 2차 DNS 서버 간에 연동이 원활하게 이루어짐을 확인할 수 있었다. 그리고 결합내성 시험을 통해 1차 DNS 서버가 시스템 오류로 인해 정지하더라도 2차 DNS 서버가 대신하여 서비스를 계속 유지해 나가는 것을 확인함으로써 결합내성을 갖는 DNS 시스템 구축이 이루어졌음을 검증하였다.

본 연구에서는 실제 예를 통해 결합내성을 갖는 DNS 시스템 구축을 위한 1, 2 DNS 서버의 이중화 구축 절차를 상세히 기술함으로써 DNS를 처음 접하는 사용자나 관리자에게 서버 구축과 활용에 많은 도움이 될 것이다.

저자소개

최재원(Jaewon 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 ~ 2013년 현재 경성대학교 컴퓨터공학부
정교수

※관심분야: 정보통신, 모바일앱, 인터넷응용, 운영
체제, 시스템 및 네트워크 보안, RFID&USN

참고문헌

- [1] 정진욱, 김현철, “TCP/IP와 인터넷”, pp.250-296, 생능출판사, 2004.
- [2] Behrouz A. Forouzan, “TCP/IP Protocol Suite”, pp.247-270, McGraw-Hill.
- [3] Andrew S. Tanenbaum, “Computer Network”, pp.665-673, Prentice Hall
- [4] 최재원, “결합내성을 가진 도메인네임 서버의 구축 및 연동시험”, pp42-51, 한국해양정보통신학회 논문지 제 15권 1호, 2011년 1월.
- [5] 서자룡, “페도라 리눅스 그대로 따라하기”, pp.533-560, 해지원, 2008.
- [6] 이소문, “페도라 리눅스 10”, 대림출판사, 2009년 3월.