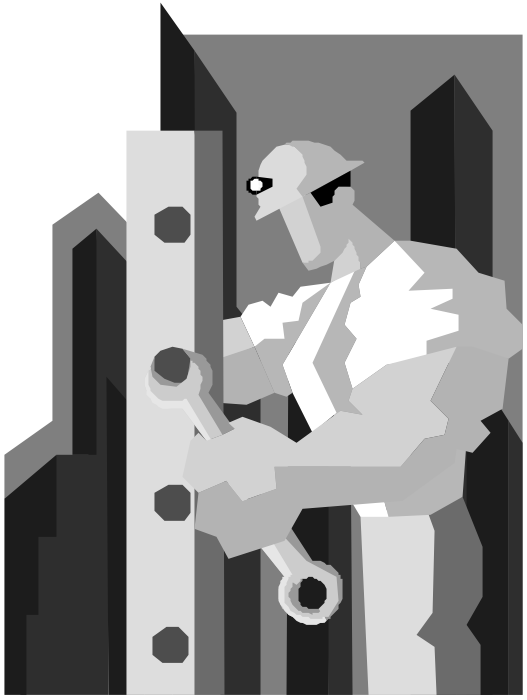
Linux 기반의 고가용
로드밸런싱 웹 서비스 구축방안

- Linux Revolution 2000
- 2000. 12. 7
- 와우리눅스㈜ 기술이사 정진호
- loveme@wowlinux.com

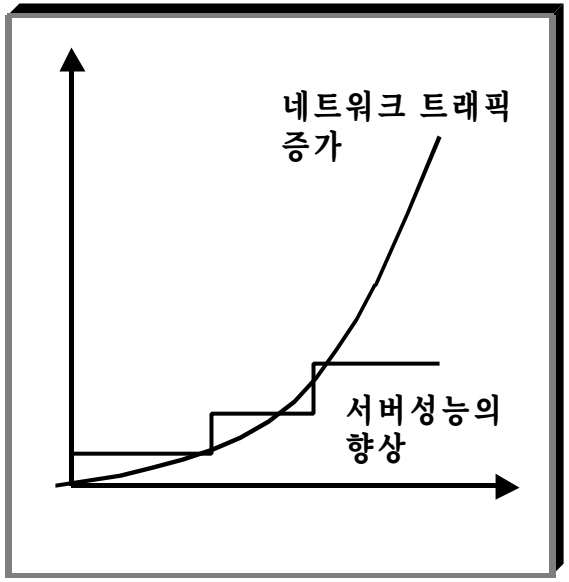
Contents

- 로드밸런싱 서비스 필요성
- 성능향상방안
- Availability & Scalability
- Linux Virtual Server
- Technology
  - Heartbeat
  - IP Address takeover
  - ARP
  - Gratuitous ARP
  - Ldirectord
  - ipvsadm
- ULTRA MONKEY 를 이용한 LVS 구축
  - Topologies
  - Preparing Cluster nodes
  - Real Server 설정
  - Director 설정
  - Heartbeat & LVS 서비스 동작 확인
  - 테스트 시나리오 (1) - real server 다운/복구
  - 테스트 시나리오 (2) - master Director 다운/복구
- 참고 사이트
- 개선필요사항



로드밸런싱 서비스 필요성

- 인터넷에서 트래픽(Traffic)은 매년 100% 이상의 증가 추세
- 서버에 접속하는 트래픽(Traffic)이 빠르게 증가하면 서버들은 짧은 시간 내에 과부하가 발생
- 이에 대응하는 시스템의 실행능력과 가용성을 높이는 문제가 대두



성능 향상 방안

- **수직적 성능 향상**
  - CPU , Memory , HDD 등의 업그레이드
- **수평적 성능 향상**
  - 클러스터링 이용 → node 추가

- **비용문제**
  - 고성능 서버는 상당히 고가
  - 2배의 성능을 위한 비용은 2배 이상
  - 고장으로 인한 가용성 문제

- **낮은 가격과 고가용성 , 고성능과 확장성 필요**

Availability & Scalability

- Availability
  - 가용성 컴퓨터 또는 프로그램에서 어떤 시스템 자원을 필요로 하는 총 시간에 대해 실제로 시스템 자원이 사용 가능한 시간의 비율. 일반적으로 컴퓨터 시스템을 구성하는 하나의 부품이 문제가 발생한 상황에서도 컴퓨터 시스템을 계속 이용할 수 있는 경우에 사용 가능도(Availability)가 높다고 한다.

- Scalability
  - 확장성. 사용자 수나 어떠한 요구 사항이 증가하더라도 서비스 효율을 떨어뜨리지 않고 대처할 수 있는 정도를 말한다. 예를 들어 100명의 사용자가 사용할 때와 1000명의 사용자가 사용할 때 시스템의 효율이 차이 나지않고 지속적인 서비스를 할 수 있을 때, 이 시스템이 scalability가 있다고 한다.

다운타임을 기준으로 한 고가용성 등급

	가용성	년간 누적 다운타임	등급
	90%	1달 이하	1
	99%	4일 이하	2
	99.9%	9시간 이하	3
	99.99%	1시간 정도	4
고가용성	99.999%	5분 이내	5
고가용성	99.9999%	30초 정도	6
고가용성	99.99999%	3초	7

■고가용성 시스템이란 대략 5분 이내의 다운타임을 가짐

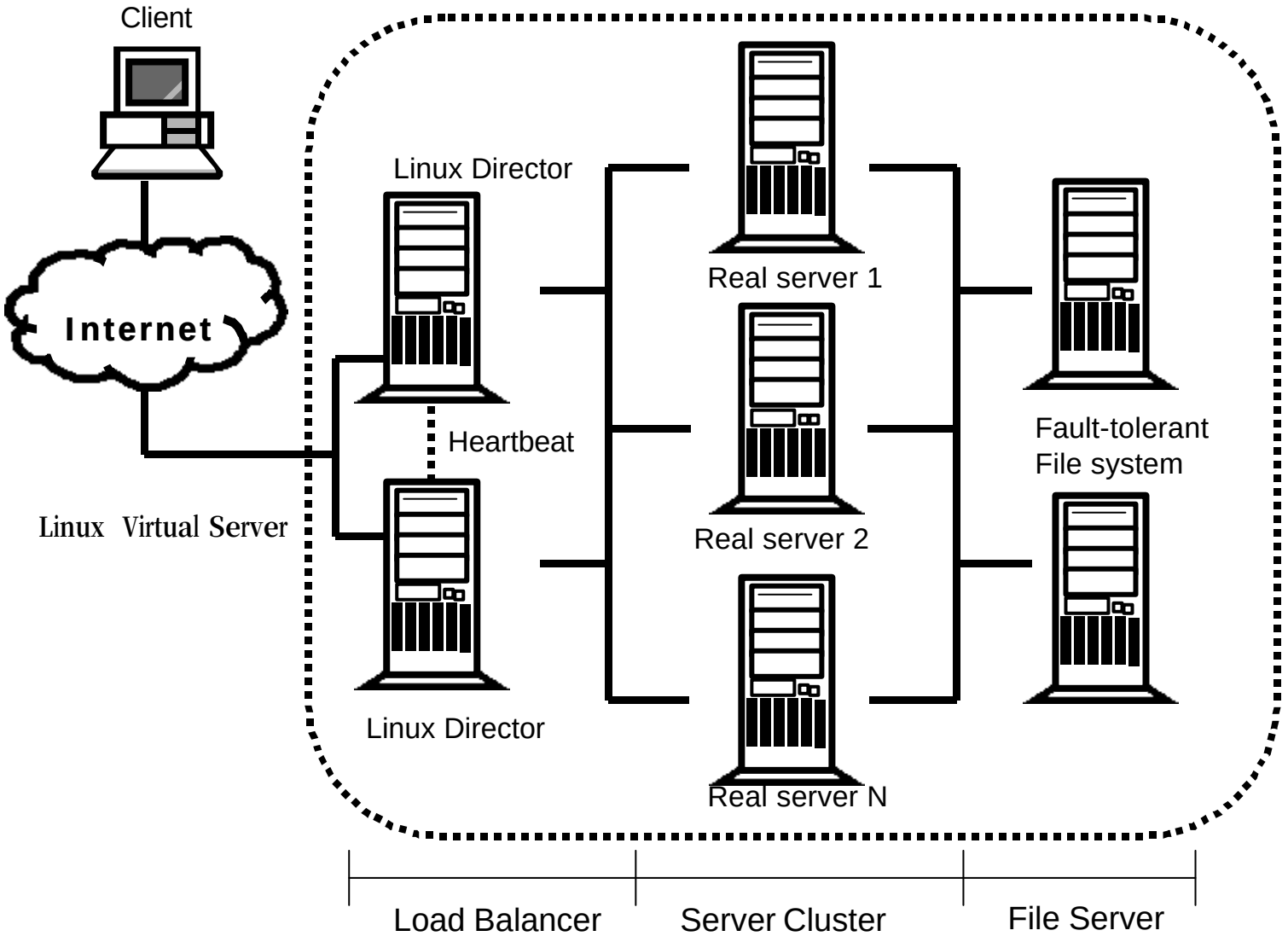
LVS – Linux Virtual Server

- 목표 : 리눅스 기반의 클러스터링을 통하여 고성능과 고가용성의 서버를 구성하여 높은 확장성, 신뢰성, 서비스를 제공하는 것

- 가상 서버는 실제 서버를 클러스터로 구성하여 뛰어난 확장성과 가용성을 구현한 서비스
- 클러스터의 구조는 최종 사용자에게는 투명하며 사용자는 오직 하나의 가상 서버만을 볼 수 있음
- 실제로는 여러대의 서버를 가지고 운영하지만 사용자는 그 사실을 알지 못하고 그냥 한대의 서버에 접속하는 것으로 생각

  - LVS의 전면(front-end)에 있는 것이 부하분산서버
  - 부하분산서버는 들어오는 요청에 대하여 서로 다른 서버로 전달
  - 클러스터로 구성된 병렬의 서비스는 단일 IP의 가상 서버로 인식됨
  - 클러스터에 노드를 투명하게 추가 및 삭제하여 확장성을 확보
  - 노드에 문제가 생기면 이를 바로 인식하여 그 즉시 시스템을 재구성
    → 높은 가용성을 확보

LVS - 구성도

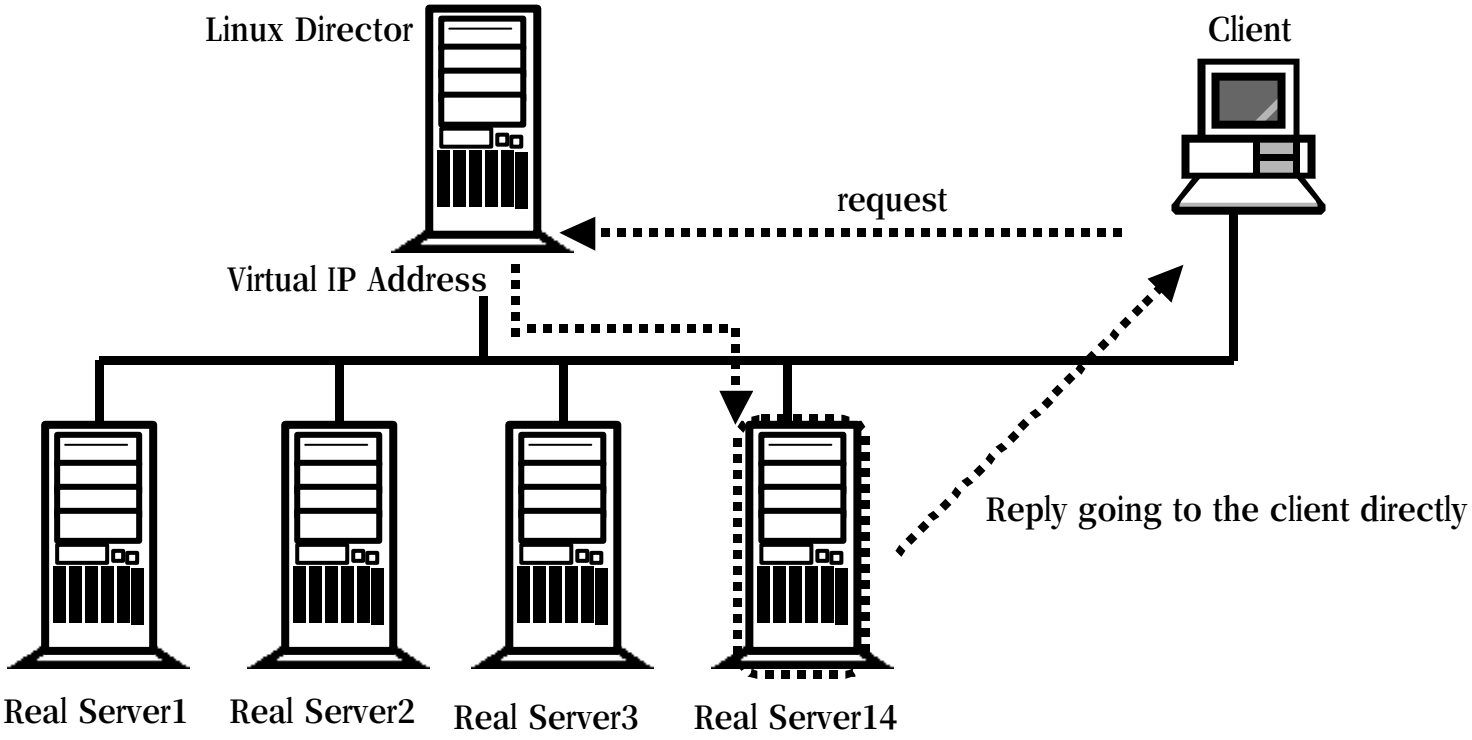

Client
Internet
Linux Virtual Server
Linux Director
Heartbeat
Linux Director
Real server 1
Real server 2
Real server N
Fault-tolerant
File system
Load Balancer
Server Cluster
File Server

LVS Load Balancing method

- NAT 이용한 가상서버
- IP 터널링 이용한 가상 서버
- 다이렉트 라우팅 이용한 가상 서버
  - 리눅스 다이렉터는 다이렉트 라우팅을 이용해 가상 서버로 가는 연결만 처리
  - 분리되어있는 네트워크 라우터를 통해 사용자에게 응답 패킷을 전송
  - 가상 서버의 확장성을 매우 높일 수 있음
  - IP 터널링 방식에 비해 터널링에서 생기는 과부하가 없음

LVS-DR 작동 방법

- 사용자의 요청은 가상 IP 주소를 가진 부하분산서버로 간다
- 부하분산서버는 스케줄링 알고리즘에 따라 클러스터에서 실제 서버를 선택
- 접속을 기록하는 해시 테이블에 새로운 접속을 추가
- 부하분산서버에서 선택한 서버로 직접 패킷을 전송
- 리얼서버는 요청을 처리하고 결과를 사용자에게 직접 전송



Direct Routing 방식의 장점

- Director 는 초기 접속만을 상대하고 실제 접속은 사용자와 서버가 직접 연결
- Director 는 사용자 요청을 받으면 이 패킷을 실제 서버로 전송하며 실제 서버가 사용자에게 응답함으로써 사용자 기계는 실제 서버와 직접 데이터 접속이 이루어 짐.
  - Director 의 하드웨어 성능이 낮아도 문제가 없음
- Director 가 패킷을 실제 서버에 넘겨 준 후에는 서비스 접속에 전혀 관여하지 않으므로 실제 서버에서 DRLB를 고려할 사항이 전혀 없음
- Director 자체도 서비스 서버와 같은 컴퓨터를 사용하며 따로 특수한 하드웨어가 필요하지 않음
- 에러 상황에 대한 대비가 우수 -> DRLB는 실제 서버를 감시하며 에러 난 서버를 서비스에서 제외하고 재작동 가능하게 되었을 때 자동적으로 서비스에 추가

가상 서버 스케줄링 알고리즘

- Round-Robin Scheduling
  - 순차적 방식을 이용해 네트워크 연결을 서로 다른 서버에 연결
  - 실제서버의 연결개수나 반응시간 등은 고려를 하지 않음

- Weighted Round-Robin Scheduling
  - 가중치기반 라운드 로빈 스케줄링
  - 서버에 서로 다른 처리 용량(가중치)을 지정 가능. 기본 가중치는 1이다.
    - 예) 서버가 A,B,C 이고 각각의 가중치가 4,3,2 일 경우 스케줄링 순서는 ABCABCABA

  - 라운드 로빈 스케줄링은 가중치 기반 라운드 로빈 스케줄링의 특별한 한 종류이며 모든 가중치가 동일한 경우

  - 장단점
    - 실제 서버의 네트워크 접속 수를 계산할 필요가 없음
    - 스케줄링의 과부하가 적어 더 많은 실제 서버를 운영 가능
    - 요청에 대한 부하가 매우 많을 경우 실제 서버사이에 동적인 부하 불균형 상태 발생 가능

가상 서버 스케줄링 알고리즘

- Least-Connection Scheduling
  - 최소 접속 스케줄링
  - 가장 접속이 적은 서버로 요청을 직접 연결하는 방식
  - 각 서버에서 동적으로 실제 접속한 숫자를 세어야 하므로 동적인 스케줄링 알고리즘중의 하나이다.
  - 접속부하가 매우 큰 경우에도 아주 효과적으로 분산 수행
  - 가장 빠른 서버에서 더 많은 네트워크 접속을 처리할 수 있음
  - 실제로는 TCP의 TIME_WAIT 상태때문에 아주 좋은 성능을 낼 수는 없음
  - 다양한 처리용량을 지난 서버로 구성되었을 경우 부하분산이 효율적으로 되지 못할 수 있음

- Weighted Least-Connection Scheduling
  - 가중치 기반 최소 접속 스케줄링
  - 최소 접속 스케줄링의 한 부분으로서 각각의 실제 서버에 성능 가중치를 가능
  - 가중치가 높은 서버에서 더 많은 요청을 받을 수 있음
  - 가중치의 비율인 실제 접속자수에 따라 네트워크 접속이 할당된다. 기본 가중치는 1이다.
  - 서버들이 같은 처리 용량을 가졌을 때는 최소 접속 스케줄링과 가중치가 있는 최소 접속 스케줄링 알고리즘이 동일

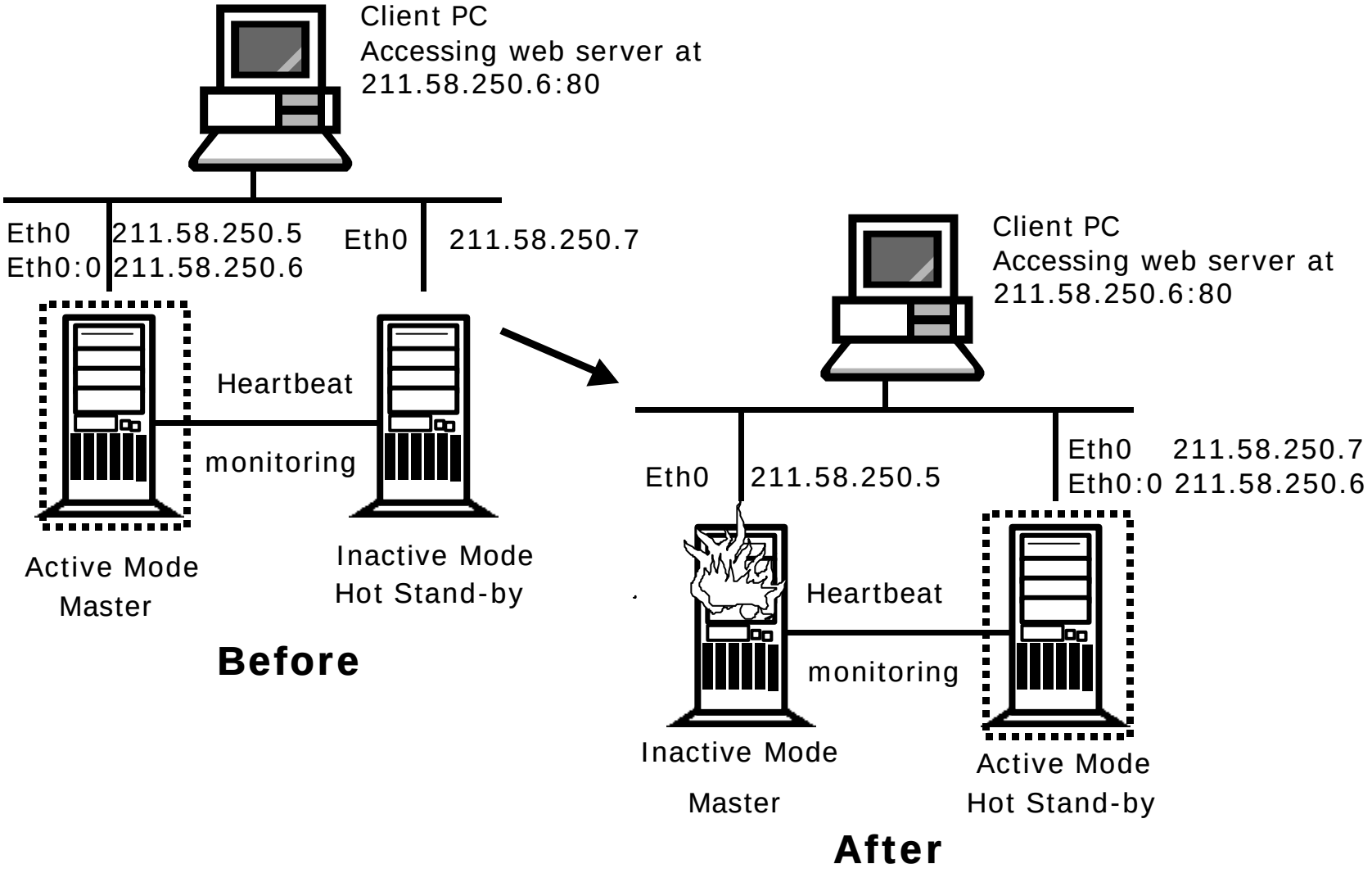
Technology

- Heartbeat
- IP Address takeover
- ARP
- Gratuitous ARP
- Ldirectord
- ipvsadm

Heartbeat

- heartbeat프로토콜을 이용하는 오픈 소스 프로젝트
- 정해진 시간간격을 두고 특정한 패킷을 물리적으로 연결된 호스트에 보내 일정 시간 응답이 없으면 정해진 응급 복구 프로세스를 수행
- 이더넷 카드 혹은 시리얼케이블을 이용 각각의 호스트를 연결
- Heartbeat 프로토콜로 연결된 호스트들은 리얼IP 이외에 가상 IP를 공유
- Master 호스트가 서비스를 책임
- Master 호스트에 문제 발생시 Standby 호스트가 가상 IP를 인계받아 서비스 계속 →IP Address Takeover

IP Address takeover



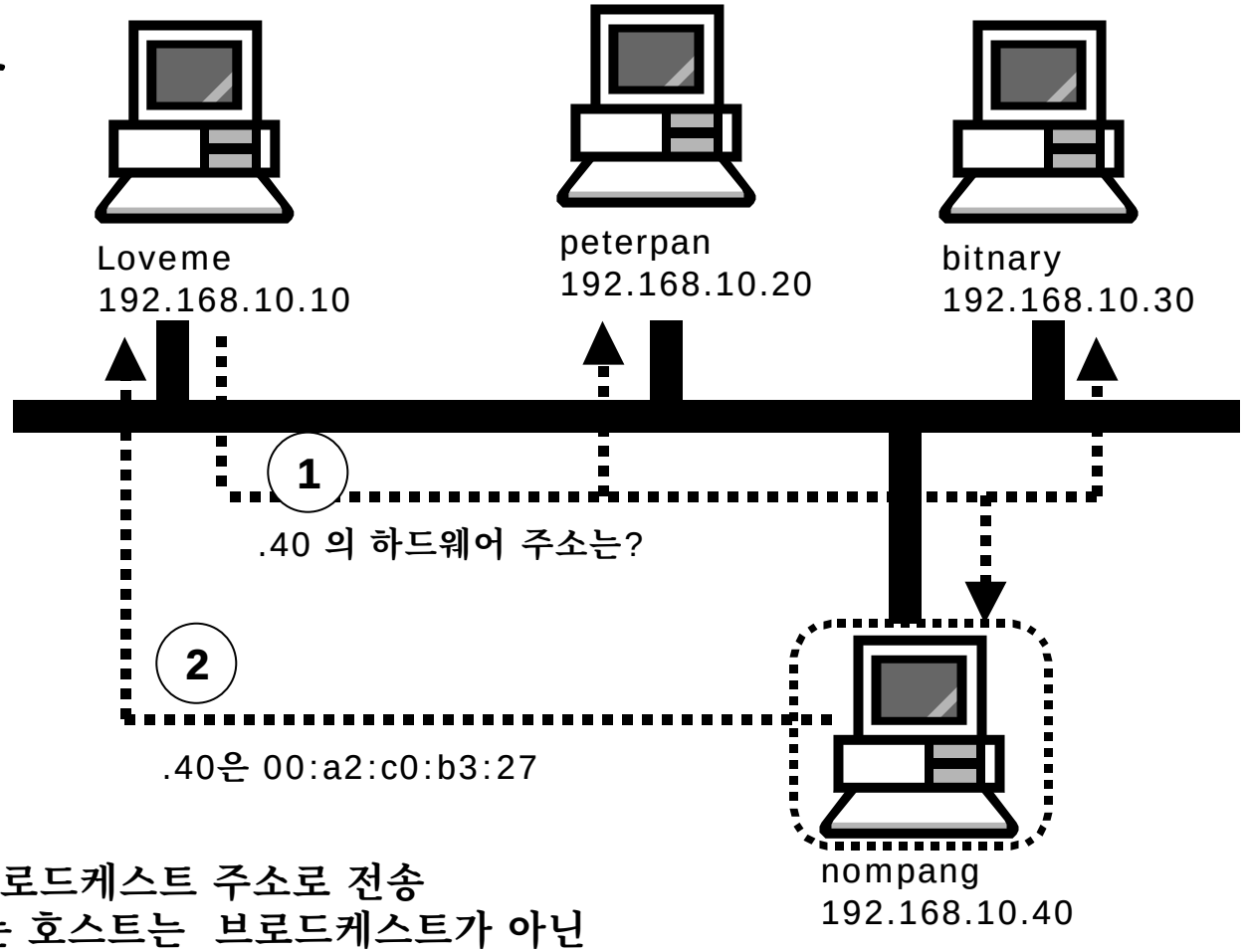
ARP

- 네트워크에 접속돼 있는 컴퓨터의 IP어드레스와 이더넷 주소(MAC어드레스)를 대응시키는 프로토콜
- 동일 네트워크 상의 IP장비의 통신을 위해서는 네트워크에 연결된 다른 장비의 하드웨어 주소 필요
- ARP (Address Resolution Protocol: 주소 변환 프로토콜) 이 이러한 서비스를 제공
- Data Link Layer에서 작동
  - OSI Reference Model 참고 *

- 로컬 네트워크 상의 호스트가 다른 호스트로 IP 패킷을 보내는 방법
  - 목적지 호스트의 IP 주소와 하드웨어 주소(MAC) 확인
  - IP 와 하드웨어 주소를 알 경우 적당한 프로토콜과 주소를 사용해 데이터 전송
  - 목적지 하드웨어 주소를 모를 경우 목적지 호스트의 하드웨어 주소 파악을 위해 ARP 호출

ARP

■ ARP 개요

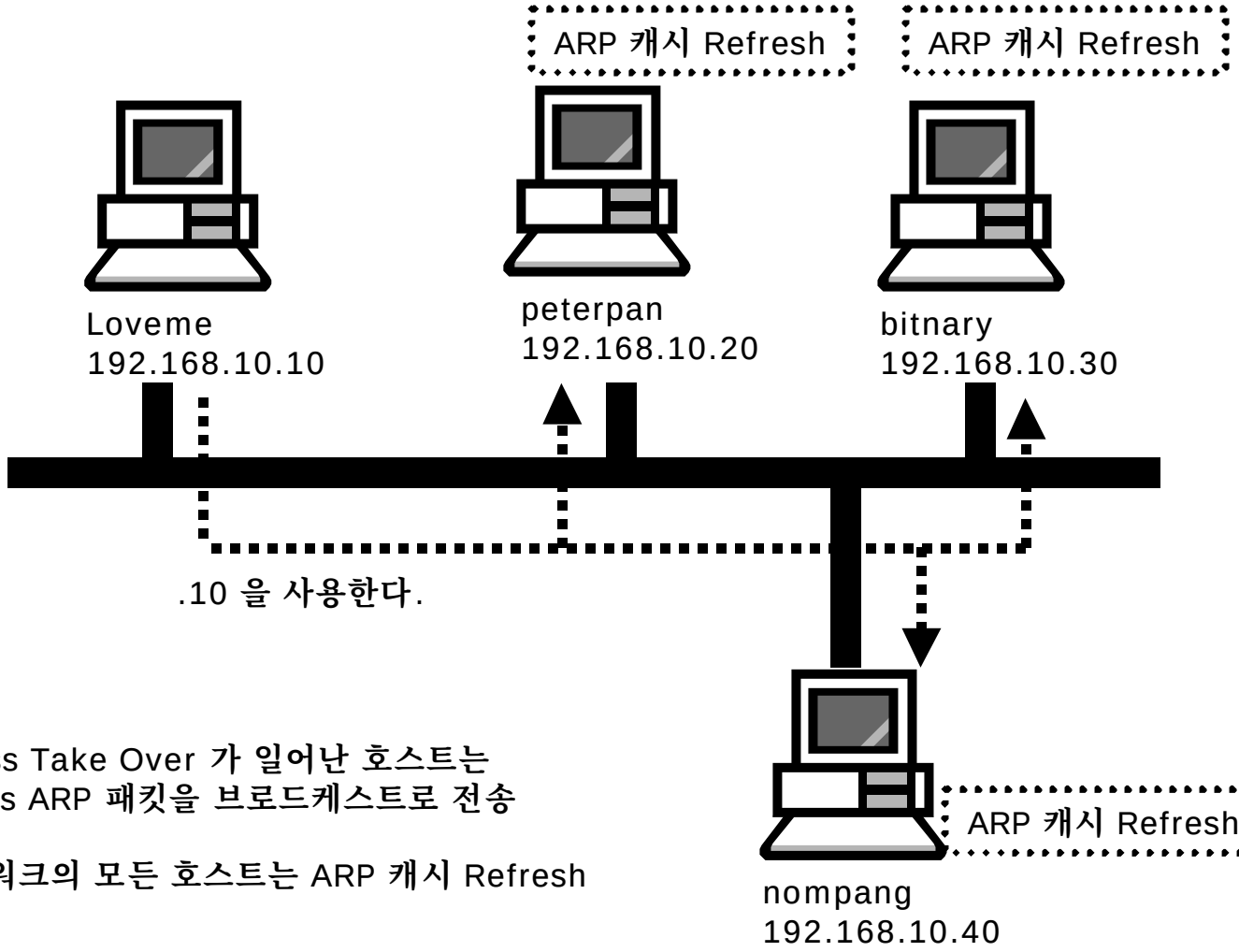


- ARP 프로토콜은 브로드케스트 주소로 전송
- 해당 IP 를 사용하는 호스트는 브로드케스트가 아닌 송신 호스트에 직접 전송
- 송신 호스트는 IP 주소 와 하드웨어 주소를 ARP 캐시에 저장

Gratuitous ARP

- Gratuitous ARP 개요
  - 다른 장비의 ARP캐시를 갱신하려는 목적으로 자신의 하드웨어 주소(MAC) 와 IP주소를 브로드케스트 전송
  - Gratuitous ARP 패킷을 받은 로컬 호스트 들은 ARP 캐시를 Refresh (응답하지는 않음)
  - 장비에 따라 다르지만 일반적인 host 들의 캐시 유효 시간은 120 초 정도
  - 필요성
    - IP Address Take Over 로 인해 특정 IP를 사용하는 NIC (MAC)이 변경되었음에도 ARP 캐시의 로 인해이전 문제가 생긴 서버의 NIC 으로 패킷을 보내는 문제가 발생 가능 → Refresh 필요

Gratuitous ARP



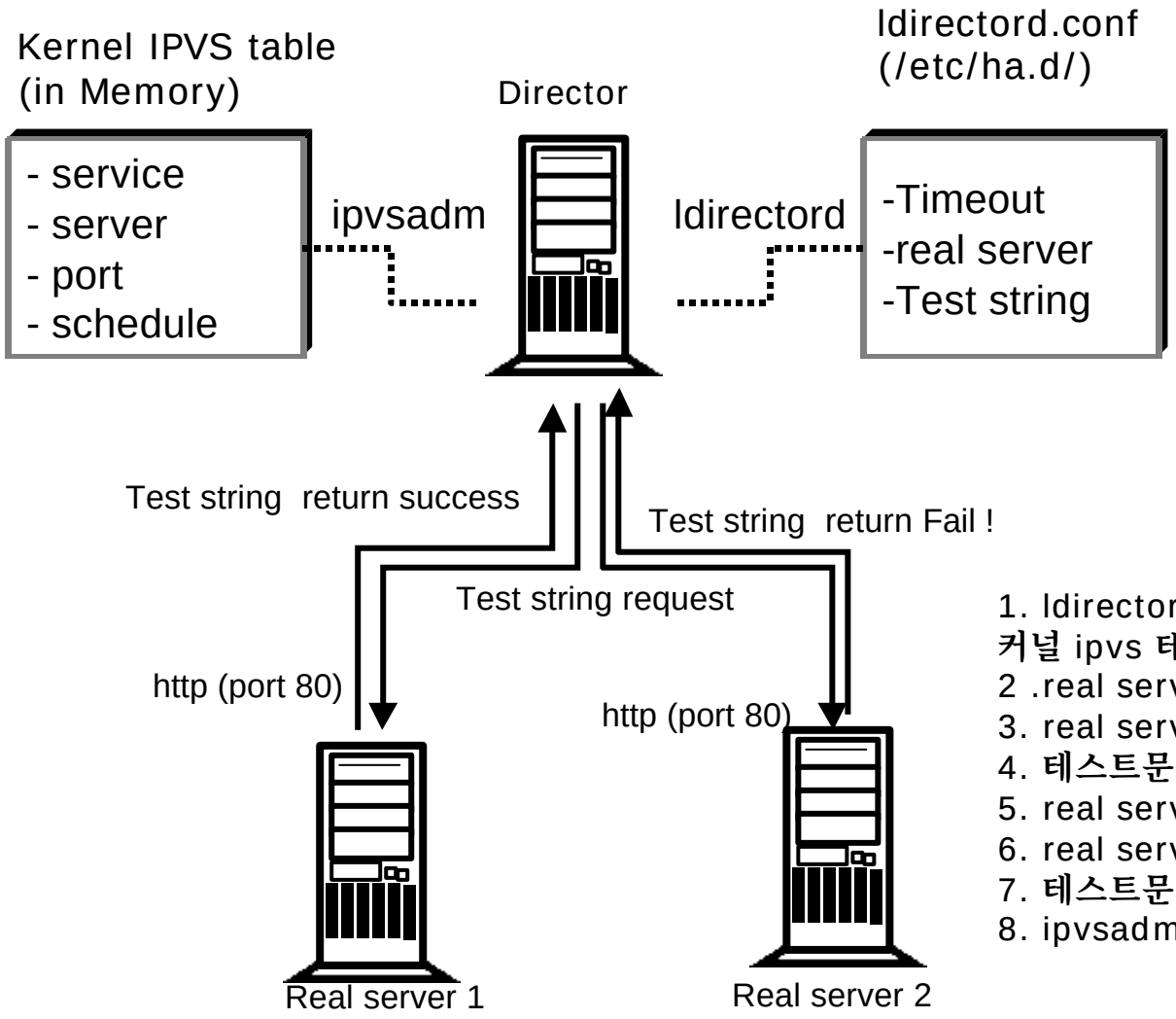
1. IP Address Take Over 가 일어난 호스트는
   Gratuitous ARP 패킷을 브로드케스트로 전송

2. 로컬 네트워크의 모든 호스트는 ARP 캐시 Refresh

Ldirectord

- 리얼서버의 상태를 모니터링 하는 PERL 스크립트
- 설정파일에 정해진 URL에 주기적으로 특정 문자열 요구
- 타임 아웃 , 응답된 문자열 유효성 등으로 서버 상태 결정
- 리얼서버의 다운 시 IPVS 테이블에서 삭제
- 리얼서버의 복구 시 IPVS 테이블에서 재 등록
- 다이렉트 라우팅은 Linux Director 서버의 IPVS 테이블을 이용해 커널에서 처리
- 모든 리얼서버가 다운되었을 경우 fall-back 서버가 응답
  - 일반적인 경우 localhost
  - Linux Director 서버도 웹 서비스 필요

Ldirectord 작동방식



1. ldirectord.conf 설정파일에 따라 커널 ipvs 테이블 설정
2 .real server 1에 테스트 문자열 요구
3. real server 1은 테스트 문자열 전송
4. 테스트문자 확인
5. real server 2에 테스트 문자열 요구
6. real server 2은 테스트 문자열 전송
7. 테스트문자 확인 불가능 (timeout)
8. ipvsadm 이용 ipvs 테이블 변경

ipvsadm

- 2개 이상의 노드를 가진 클러스터링 시스템에서 커널내부의 IPVS 테이블을 설정하고 유지하는 유틸리티
- IPVS테이블은 커널이 로드벨런서(Director)로 들어오는 패킷을 각각의 리얼서버 노드로 전달하기 위해 참조
- 2개의 프로토콜 지원
  - TCP 와 UDP
- 3가지 방식의 패킷 포워딩 방식 지원
  - NAT, tunneling, direct routing
- 4가지 스케줄링 알고리즘 지원
  - round robin, weighted round robin,
  - least-connection weighted least-connection

ULTRAMONKEY 를 이용한 LVS 구축

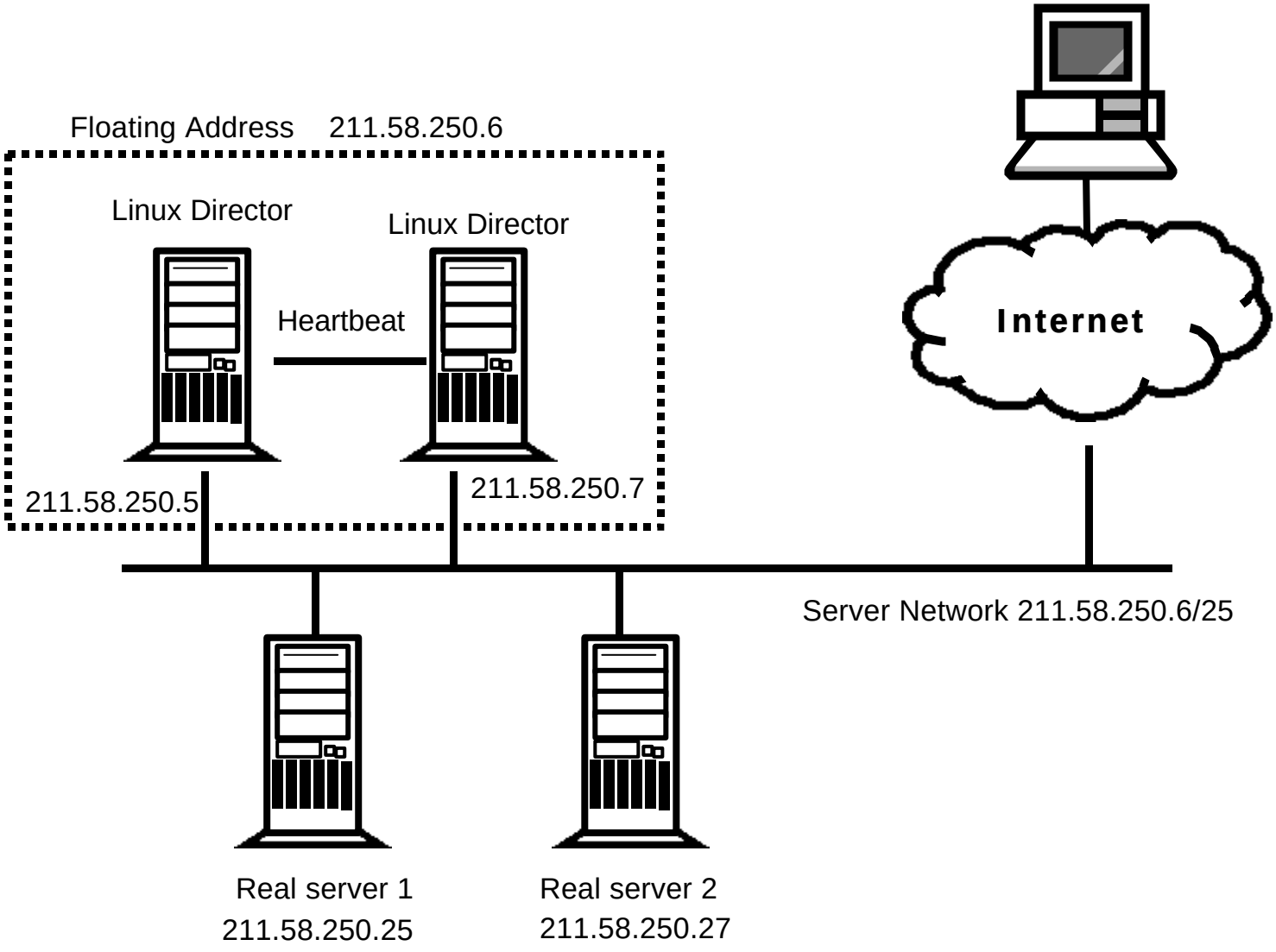
- Topologies
- Preparing Cluster nodes
- Real Server 설정
- Director 설정
- Heartbeat & LVS 서비스 동작 확인
- 테스트 시나리오 (1) - real server 다운/복구
- 테스트 시나리오 (2) - master Director 다운/복구

ULTRA MONKEY

- High Availability and Load Balancing Solution

- Key Features
  - Layer 4 Switching using The Linux Virtual Server
  - Easily expandable to a large number of IP based virtual services
  - High Availability provided by Heartbeat protocol
  - Service level monitoring using Ldirectord
  - All Code is Open Source
- Easily expandable to other services such as email and FTP.

Topologies – High Availability Load Balancing Service



Topologies – overview

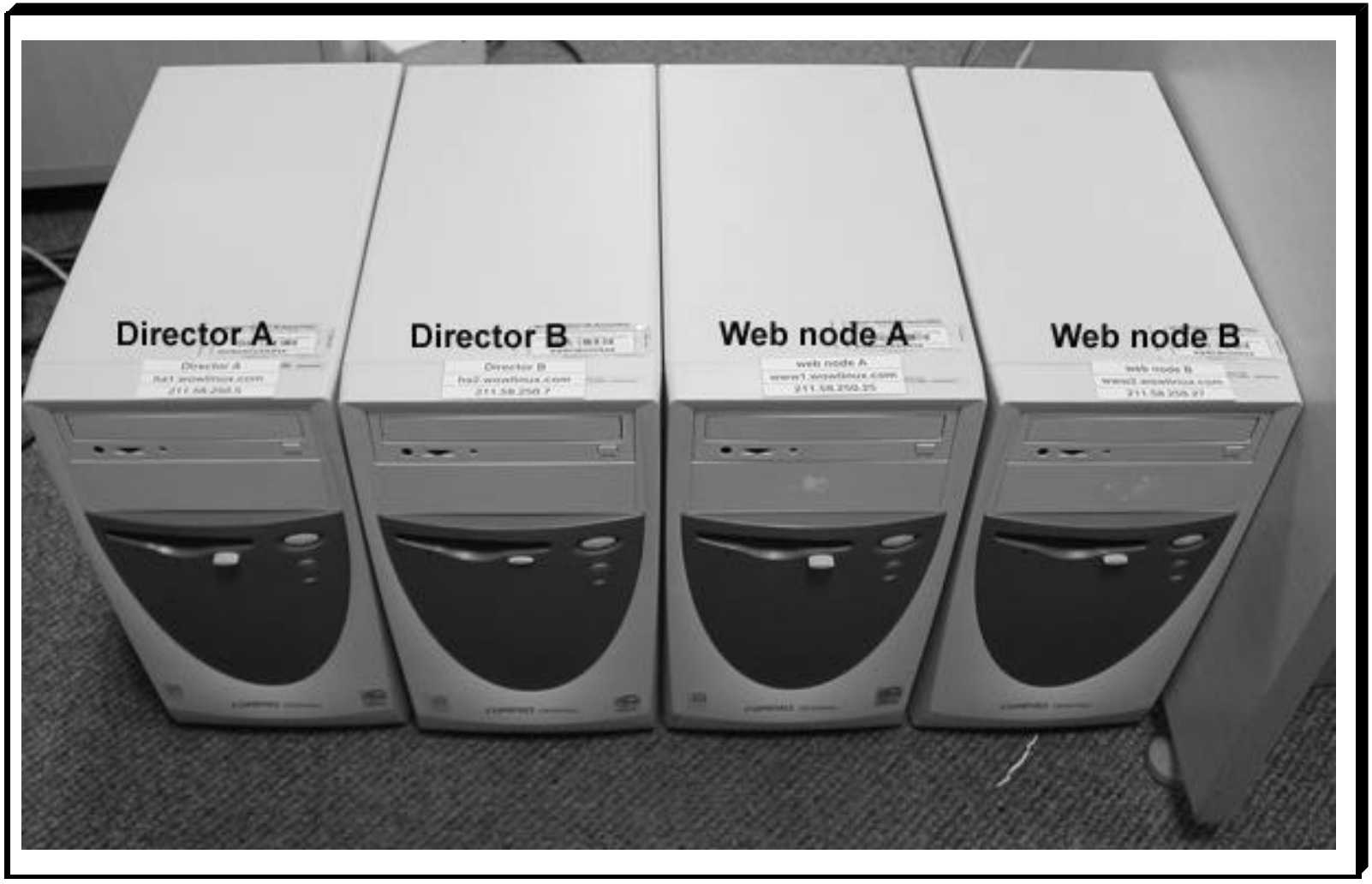
- 2대의 Director 를 Heartbeat 으로 연결
  - 한대의 Directors 는 운영 중(Master)
  - 다른 한대는 운영중인 Director 를 감시 (hot stand-by)

- DNS 및 외부 클라이언트에는 이 가상 IP가 웹 서비스로 등록
- director 는 가상 IP (211.58.250.6) 로 오는 패킷을 처리
- 각각의 리얼서버는 80 port 에서 Apache HTTP 서비스 운영 중
- 클라이언트의 접속은 LVS/DR 방식으로 리얼서버로 부하 분산
- 이후 클라이언트는 리얼서버 직접 연결

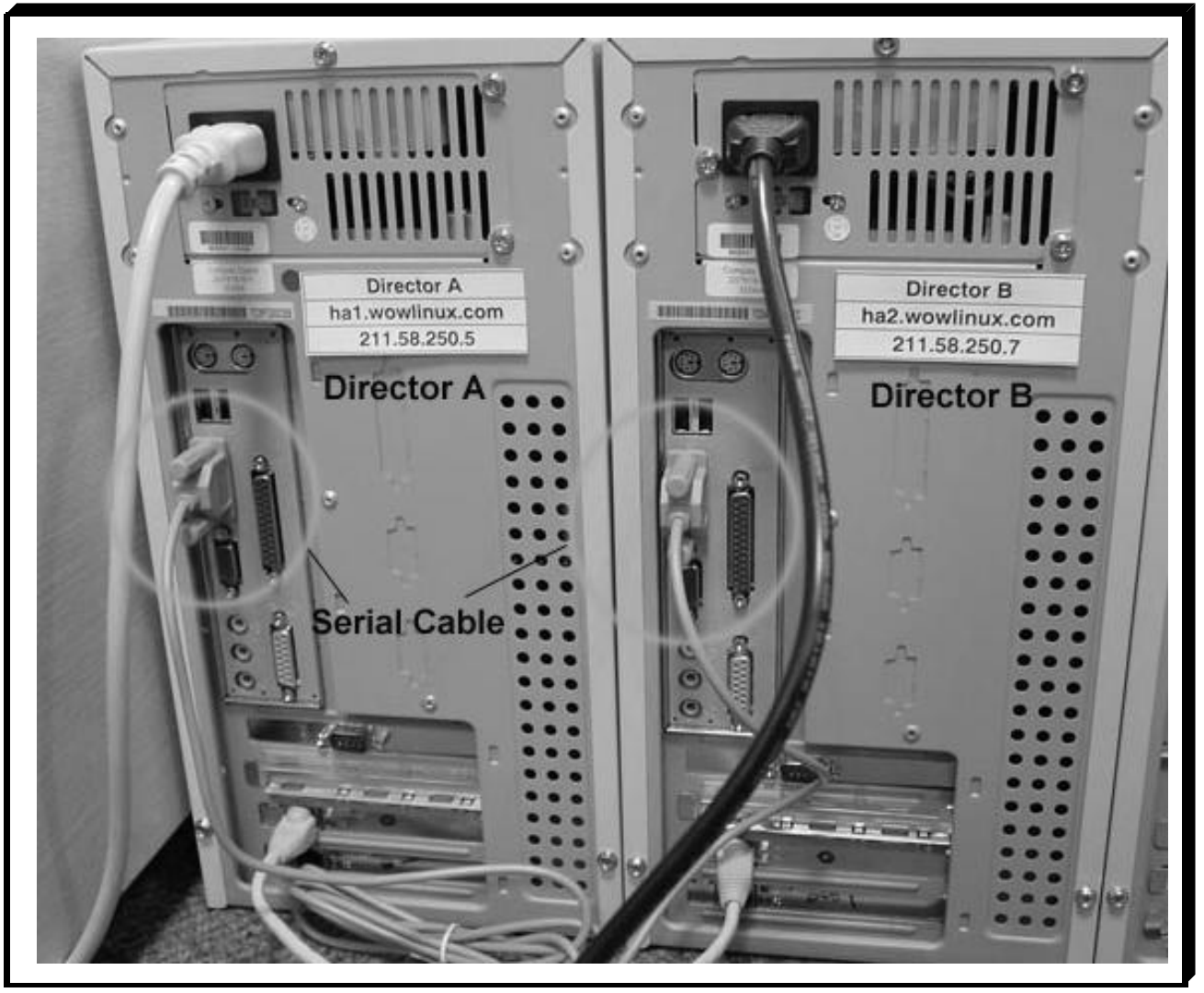
Preparing Cluster nodes

- HW – Director & Web node
  - Compaq 데스크프로 SB
    - Intel Celeron 500MHz, Intel 810 Chipset ,64MB RAM, 8.4GB HDD , RealTeck 8139c NIC
    - 48x CD-ROM, Integrated Graphic Controller(4MB)
- OS
  - Wowlinux 6.2 update (kernel 2.2.16) Full Install
- Network
  - Ha.wowlinux.com : 211.58.250.6 (Virtual IP)
  - Ha1.wowlinux.com : 211.58.250.5 (Director 1)
  - Ha2.wowlinux.com : 211.58.250.7 (Director 2)
  - www1.wowlinux.com : 211.58.250.25 (web node 1)
  - www2.wowlinux.com : 211.58.250.27 (web node 2)
  - Ha1 & ha2 시리얼 케이블 연결

Preparing Cluster nodes



Preparing Cluster nodes



Network 설정 - 공통

■ /etc/hosts 내용

```
127.0.0.1      localhost.localdomain      localhost
211.58.250.6    ha.wowlinux.com           ha
211.58.250.5    ha1.wowlinux.com          ha1
211.58.250.7    ha2.wowlinux.com          ha2
211.58.250.25   www1.wowlinux.com         www1
211.58.250.27   www2.wowlinux.com         www2
```

Real Server 설정 (1)

- 리얼서버는 가상 IP 주소(211.58.250.6)를 로컬주소로 인식하도록 설정되어야 함
  - loopback device 에 IP Alias 기능을 이용
  - /etc/sysconfig/network-scripts/ifcfg-lo:0 스크립트를 다음과 같이 설정

```
DEVICE=lo:0
IPADDR=211.58.250.6
NETMASK=255.255.255.255
NETWORK=211.58.250.0
BROADCAST=211.58.250.127
ONBOOT=yes
NAME=loopback
```

- 불필요한 경고 메시지를 없애기 위해 게이트웨이는 eth0 로 설정
  - /etc/sysconfig/network 파일을 확인한다.
  - GATEWAYDEV=eth0

Real Server 설정 (2)

- 위에서 설정한 IP alias 동작시키기
  - #/sbin/ifup lo
- /sbin/ifconfig 로 확인 가능.

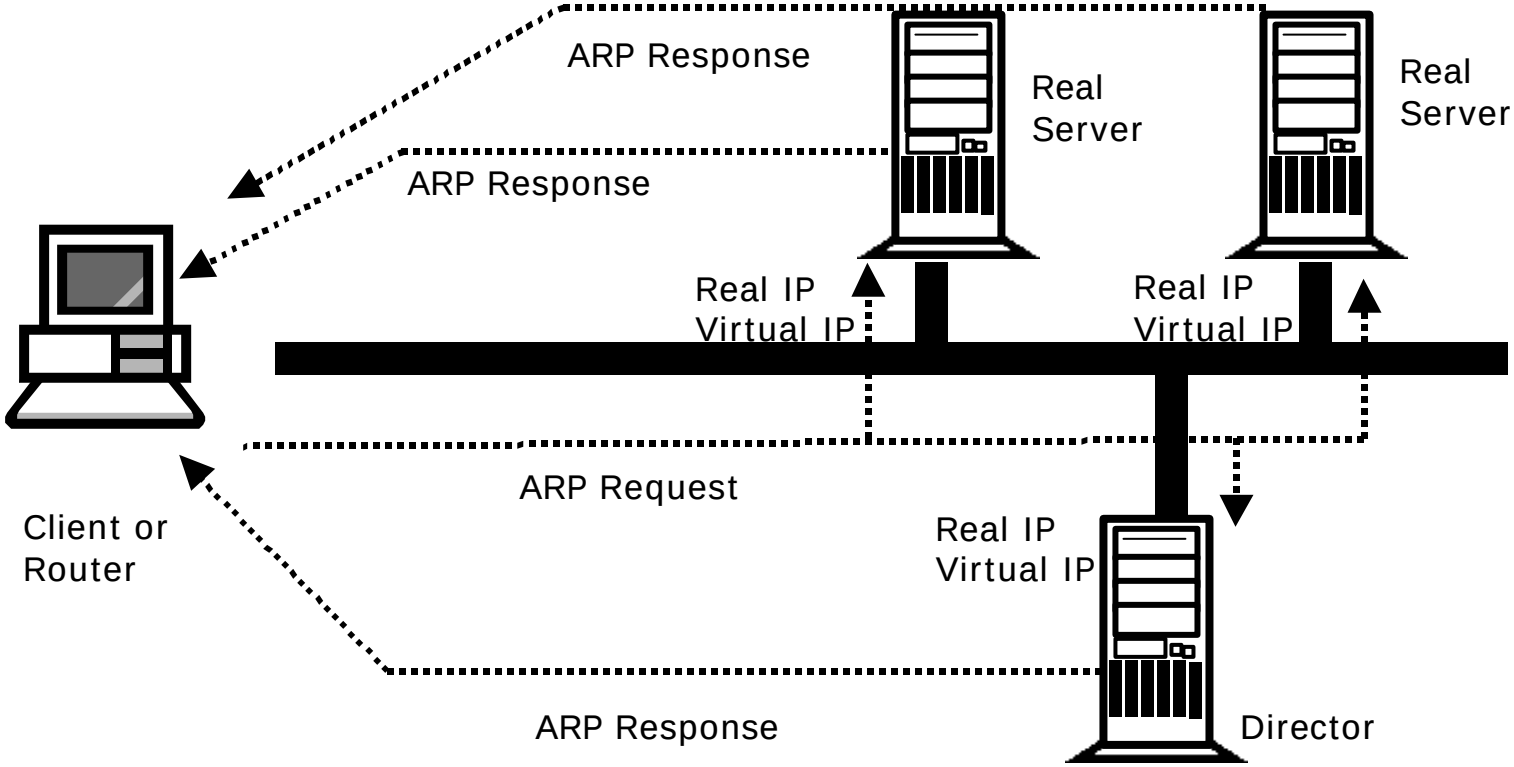
```
lo:0    Link encap:Local Loopback
        inet addr:211.58.250.6  Mask:255.255.255.255
        UP LOOPBACK RUNNING  MTU:3924  Metric:1
```

  - 주의 : Loopback 인터페이스가 네트워크의 모든 트래픽을 받아들일 수 있으니 반드시211.58.250.6 만을 로컬로 인식하도록 넷 마스크를 255.255.255.255 이용

- 각각의 web node에서 아파치 데몬이 서비스 되도록 ntsysv 이용 시작 스크립트 수정
- /etc/ha.d/conf/ldirectord.cf 에 설정된 Index.html Test 페이지에 문자열을 포함하도록 설정
  - Test Page of www1.wowlinux.com
  - Test Page of www2.wowlinux.com

Real Server 설정 (3) - ARP 응답 문제

- 루프 백 디바이스에 엘리어스로 가상 IP를 추가한 경우 로컬네트워크의 ARP 요청에
- 요청이 로드밸런서로 가지 않고 리얼 서버가 먼저 응답하므로 부하분산이 되지 않는 문제가 생길 수 있다.



Real Server 설정 (3) ARP 응답 문제 해결

- 리눅스 커널 2.2.14부터 Hidden Device (ARP Request 에 응답하지 않는 네트워크 디바이스) 기능을 지원 쉽게 ARP 문제를 해결 가능
  - http://www.linuxvirtualserver.org/arp.html 참고

- Real server 의 loopback 디바이스가 ARP 요청에 응답하지 않도록 하기 위해 인터페이스를 hidden 으로 설정
  - /etc/sysctl.conf 파일에 아래 라인을 추가.

```
# Enable configuration of hidden devices
net.ipv4.conf.all.hidden = 1
# Make the loopback device hidden
net.ipv4.conf.lo.hidden = 1
```

- 변경 사항을 활성화 하기 위해
  - /sbin/sysctl -p

Serial cable 연결 / 테스트 (1)

- 2대의 LVS 서버의 포트에 시리얼 케이블 연결
- ha2에 다음과 같은 명령을 내린다.

```
[root@ha2 /root]# cat < /dev/ttyS0
```

- ha1에서 다음과 같이 테스트 해 본다.

```
[root@ha1 /]# echo HELLO > /dev/ttyS0
[root@ha1 /]# echo 'THIS IS HEARTBEAT TEST from
HA1' > /dev/ttyS0
```

- ha2 에서 다음과 같은 메시지가 나오면 정상 동작.

```
[root@ha2 /root]# cat < /dev/ttyS0
HELLO

THIS IS HEARTBEAT TEST from HA1
```

Serial cable 연결 / 테스트 (2)

■ 현재 시리얼 포트의 동작 상태 확인

```
#cat /proc/tty/driver/serial
serinfo:1.0 driver:4.27
0: uart:16550A port:3F8 irq:4 baud:9600 tx:18 rx:0 CTS|DSR|CD
1: uart:16550A port:2F8 irq:3 baud:9600 tx:11 rx:0
2: uart:unknown port:3E8 irq:4
...
```

■ 마지막의 CD 는 Carrier Detect 로 케이블이 정상적으로 연결되었음을 말한다.

Director 설정 (1) - Install 준비

- 커널 버전 확인
  - 2.2.14 이전에서는 커널 패치 필요.
  - kernel 2.2.16은 곧바로 IPVS기능 사용가능
  - 와우리눅스 6.2up은 이미 지원
- Ultramonkey RPM Package 구하기
  - ftp://ultramonkey.sourceforge.net/pub/ultramonkey/ultramonkey-1.0.1/RPMS/
- 이미 설치된 Piranha 제거
  - Redhat 의 piranha 및 클러스터링 패키지는 Ultramonkey 와 함께 사용할 경우 문제 발생
  - Piranha 와 Clustering 관련 패키지를 모두 지운다.
  - /mnt/cdrom/RedHat/base/comps 파일 참고
    - piranha , piranha-docs , pvm , lam, piranha-gui, pvm-gui

Director 설정 (2) - RPM Package install

- HA1 서버에 아래 11 가지 패키지를 설치
  - perl-libnet-1.0607-1.i386.rpm
  - perl-HTML-Parser-3.07-1.i386.rpm
  - MIME-Base64-2.11-2.i386.rpm
  - URI-1.06-1.i386.rpm
  - openssl-0.9.5a-2.i386.rpm
  - perl-Net-SSLeay-1.05-4.i386.rpm
  - LWP.pm-5.48-2.i386.rpm
  - heartbeat-0.4.7bpre1-1.um.1.i386.rpm
  - initscripts-5.00-1.1.um.1.i386.rpm
  - ipvsadm-1.9-1.um.1.i386.rpm
  - ultramonkey-doc-1.0.1-1.noarch.rpm

  - 패키지 간에 의존성 문제가 있으므로 반드시 순서대로 설치
- Rebooting !

Director 설정 (3) - 서버의 라우팅 기능 확인

- Director 서버는 반드시 자신에게 접속하는 클라이언트의 패킷을 리얼서버로 보낼 수 있는 IPVS 기능 필요
  - 따라서 커널의 IPVS4 포워딩 기능이 사용가능 하도록 설정

- /etc/sysctl.conf 파일의 수정을 통해 쉽게 변경가능

```
# Disables packet forwarding
net.ipv4.ip_forward = 1
# Enables source route verification
net.ipv4.conf.all.rp_filter = 1
# Disables automatic defragmentation
net.ipv4.ip_always_defrag = 0
# Disables the magic-sysrq key
kernel.sysrq = 0
```

Director 설정 (3) - 서버의 라우팅 기능 확인

- net.ipv4.ip_forward = 1 로 변경한 후 저장
- 변경 사항을 활성화 하기 위해 아래 명령을 이용

```
[root@ha1 conf]# /sbin/sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_always_defrag = 0
kernel.sysrq = 0
```

Director 설정 (4) - ldirectord 설정

- 실행을 위해 /etc/ha.d/conf/ldirectord.cf 에 설정파일 필요
- 인스톨 직후에는 없음 적당한 것을 하나 복사해서 사용
  - /usr/doc/ultramonkey-doc-1.0.1/topologies/config/

  - /etc/ha.d/conf/ldirectord.cf 설정 예

```
timeout=3
checkinterval=1
fallback=127.0.0.1:80
virtual=211.58.250.6:80
        real=211.58.250.25:80 gate
        real=211.58.250.27:80 gate
        service=http
        request="index.html"
        receive="Test Page"
        scheduler=rr
        protocol=tcp
```

Director 설정 (4) - ldirectord 설정

- checkinterval : 테스트 페이지 요청 주기
- fallback : 모든 리얼서버 다운시 이동할 웹 서버
- virtual : 가상 서비스 네트워크
- real : 리얼 서버 IP 및 포트 , 포워딩 방식
  - gate = Direct Routing
  - masq = Network address translation
  - ipip = IP Tunneling
- service : 서비스 프로토콜
- request : 요청 페이지
- receive : 요청 페이지에서 찾을 문자열
- scheduler : 스케줄링 방식
  - rr = Round-Robin
  - wrr = Weighted Round-Robin
  - lc = Least-Connection
  - wlc = Weighted Least-Connection

Director 설정 (5) - Heartbeat 설정

- LVS 서버의 /etc/ha.d/ 3가지 파일이 필요
  - authkeys : 두 대의 LVS 서버사이에 인증을 위한 파일
  - ha.cf : heartbeat의 주기능 설정 파일.
  - haresources : heartbeat 관리 서비스내용 설정파일
    - http://www.linux-ha.org/download/GettingStarted.html 참고

- authkeys 파일

```
auth 1
1 crc
#2 sha1 ultramonkey
#3 md5 Hello!
```

  - 여기서는 crc 방식을 이용.
  - 시리얼라인과 같이 네트워크 보안이 필요하지 않을 경우 간단히 사용
  - 퍼미션을 600 으로 변경필요

Director 설정 (5) - Heartbeat 설정

- ha.cf

```
debugfile /var/log/ha-debug
logfile /var/log/ha-log
keepalive 3
deadtime 9
serial  /dev/ttyS0
baud    19200
node    ha1.wowlinux.com
node    ha2.wowlinux.com
```

  - keepalive : 각 Heartbeat 신호 사이의 경과 시간
  - deadtime : 호스트에 이상발생 판단을 위한 시간
  - serial : 물리적 연결된 시리얼 포트
  - baud : 전송 속도
  - node : Heartbeat 로 연결된 각각의 노드
    - #uname -n 명령을 통해 나온 이름을 적어준다

Director 설정 (5) - Heartbeat 설정

- Haresource

```
...
ha1.wowlinux.com      211.58.250.6   ldirectrod
...
```

  - ha1.wowlinux.com : master LVS 서버의 node 명
  - 211.58.250.6 : LVS서버의 가상 IP
    - heartbeat 서비스는 자동적으로 이 IP 로 IP Alias 를 실행
    - master node 가 죽으면 hot standby 서버가 이 IP를 인계 받아 eth0:0로 매핑하고 서비스가 계속 (IP Take over)
  - ldirectrod : heartbeat 에 의해 서비스될 스크립트
    - 디폴트로 /etc/rc.d/init.d/ 에서 실행할 스크립트를 찾음

LVS ,리얼 서버 - HDD복사

- 같은 사양의 서버를 준비한 뒤 run level 1 에서 HDD를 복사하도록 한다.
  - #init 1
  - #cat /dev/hda > /dev/hdc
- 복사가 끝나면 각각 서버에 다시 HDD를 설치하고 네트워크 설정을 확인

Heartbeat 서비스 시작

- www1, www2 는 각각의 아파치데몬을 기동

```
[root@www1 /root]# /etc/rc.d/init.d/httpd start
Starting httpd: [  OK  ]
```

- ha1 , ha2 에서 heartbeat 를 기동

```
[root@ha1 ha.d]# /etc/rc.d/init.d/heartbeat start
Starting High-Availability services: [  OK  ]
```

  - 주의
    - ldirectord 가 독립적으로 실행되어서는 안됨
    - ntsysv 를 수정해 ldirevtord 는 부팅시 자동실행을 막고
    - heartbeat 는 자동실행으로.

Heartbeat & LVS 서비스 동작 확인

■ Heartbeat 로그파일 확인

```
[root@ha1 /etc]# cat /var/log/ha-log

heartbeat:11/17_13:01:55 INFO: Running /etc/ha.d/rc.d/status status
heartbeat:11/17_13:02:03 INFO: Running /etc/ha.d/resource.d/IPaddr
211.58.250.6 status
heartbeat:11/17_13:02:03 INFO: Running /etc/ha.d/rc.d/ip-request ip-
request
heartbeat:11/17_13:02:13 Acquiring resource group: ha1.wowlinux.com
211.58.250.6 ldirectord
heartbeat:11/17_13:02:13 INFO: Running
/etc/ha.d/resource.d/ldirectord  start
heartbeat:11/17_13:02:14 INFO: Running /etc/ha.d/resource.d/IPaddr
211.58.250.6 start
heartbeat:11/17_13:02:14 INFO: ifconfig eth0:0 211.58.250.6 netmask
255.255.255.128  broadcast 211.58.250.127
heartbeat:11/17_13:02:14 Sending Gratuitous Arp for 211.58.250.6 on
eth0:0 [eth0]
```

Heartbeat & LVS 서비스 동작 확인

■ ldirectord 로그파일 확인

```
[root@ha1 /etc]# cat /var/log/ldirectord.log

[17 13:02:14 2000|ldirectord] Starting Linux Director Daemon
[17 13:02:14 2000|ldirectord] Adding virtual server:
211.58.250.6:80
[17 13:02:14 2000|ldirectord] Starting fallback server for:
211.58.250.6:80
[17 13:02:14 2000|ldirectord] Adding real server: 211.58.250.25:80
(1*211.58.250.6:80)
[17 13:02:14 2000|ldirectord] Turning off fallback server for:
211.58.250.6:80
[17 13:02:14 2000|ldirectord] Adding real server: 211.58.250.27:80
(2*211.58.250.6:80)
```

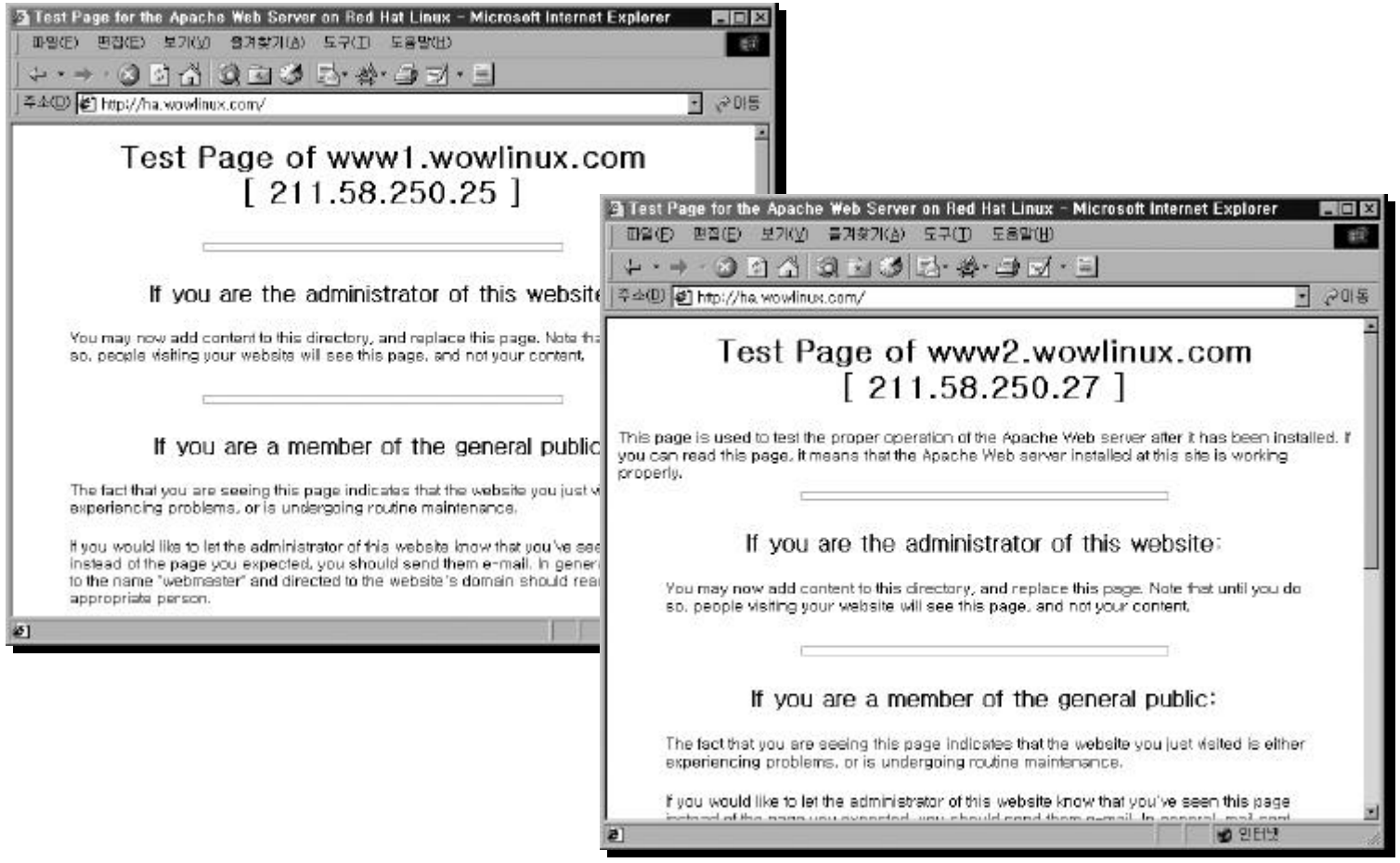
Heartbeat & LVS 서비스 동작 확인

■ ipvsadm 을 이용한 PIVS 테이블 확인

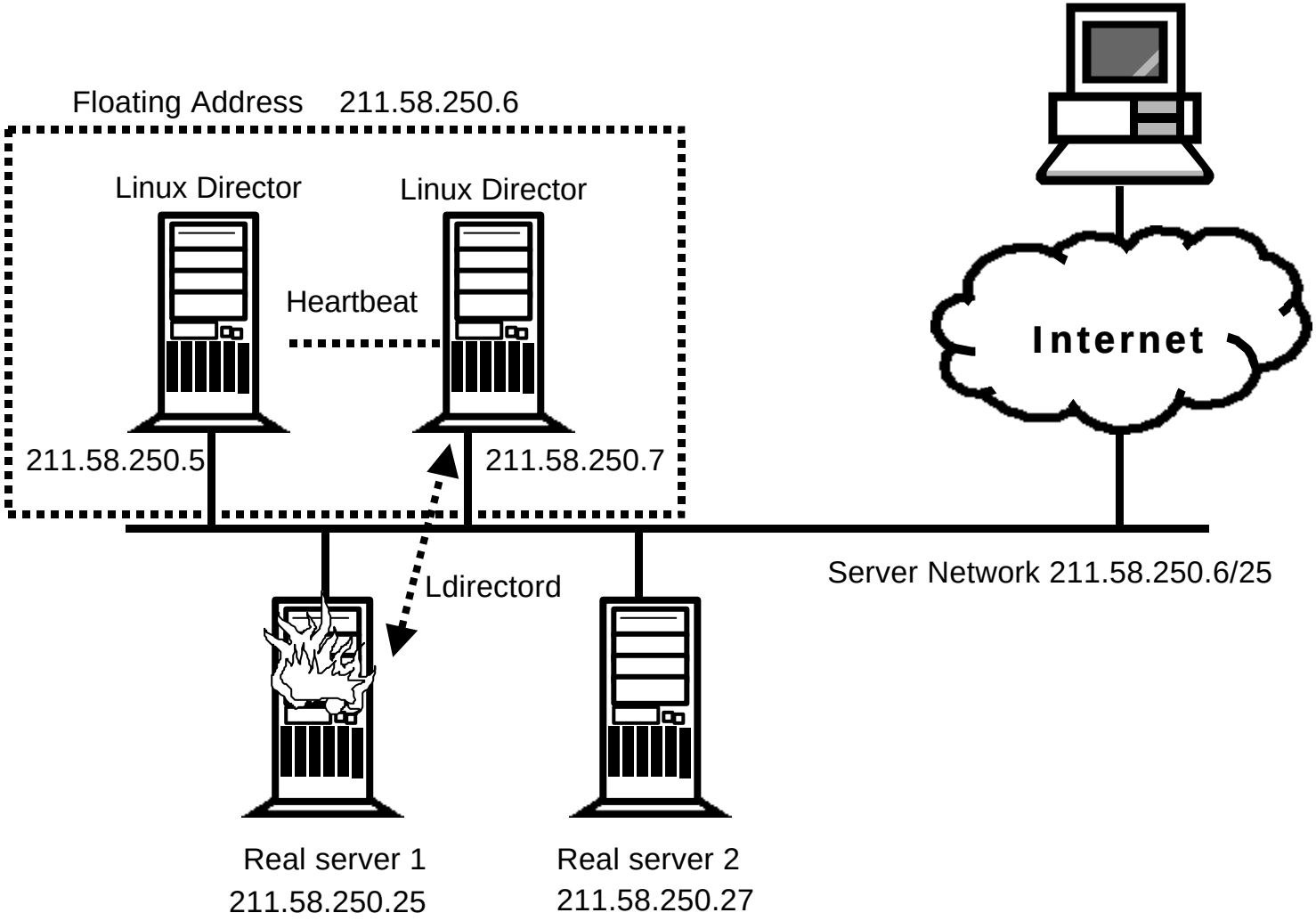
```
[root@ha1 /etc]# ipvsadm

IP Virtual Server version 0.9.12 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn
InActConn
TCP  ha.wowlinux.com:www rr
  -> www2.wowlinux.com:www      Route   1      0          0
  -> www1.wowlinux.com:www      Route   1      0          0
```

웹 브라우저를 통한 확인

■ **http://ha.wowlinux.com 접속**



테스트 시나리오 (1) – real server 다운/복구



테스트 시나리오 (1) – real server 다운

- www1의 웹서비스 중지

```
[root@www1 /root]# /etc/rc.d/init.d/httpd stop
Stopping httpd: [  OK  ]
```

- Ha1 의 /var/log/ldirectord.log

```
...
[Fri Nov 17 13:24:28 2000|ldirectord.cf] Removing real
server: 211.58.250.25:80 (1*211.58.250.6:80)
```

- Ha1 의 ipvsadm

```
[root@ha1 etc]# ipvsadm
IP Virtual Server version 0.9.12 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port        Forward Weight ActiveConn
InActConn
TCP  ha.wowlinux.com:www rr
  -> www2.wowlinux.com:www      Route   1      0          0
```

테스트 시나리오 (1) – real server 복구

■ www1의 웹서비스 재시작

```
[root@www1 /root]# /etc/rc.d/init.d/httpd start
Starting httpd: [  OK  ]
```

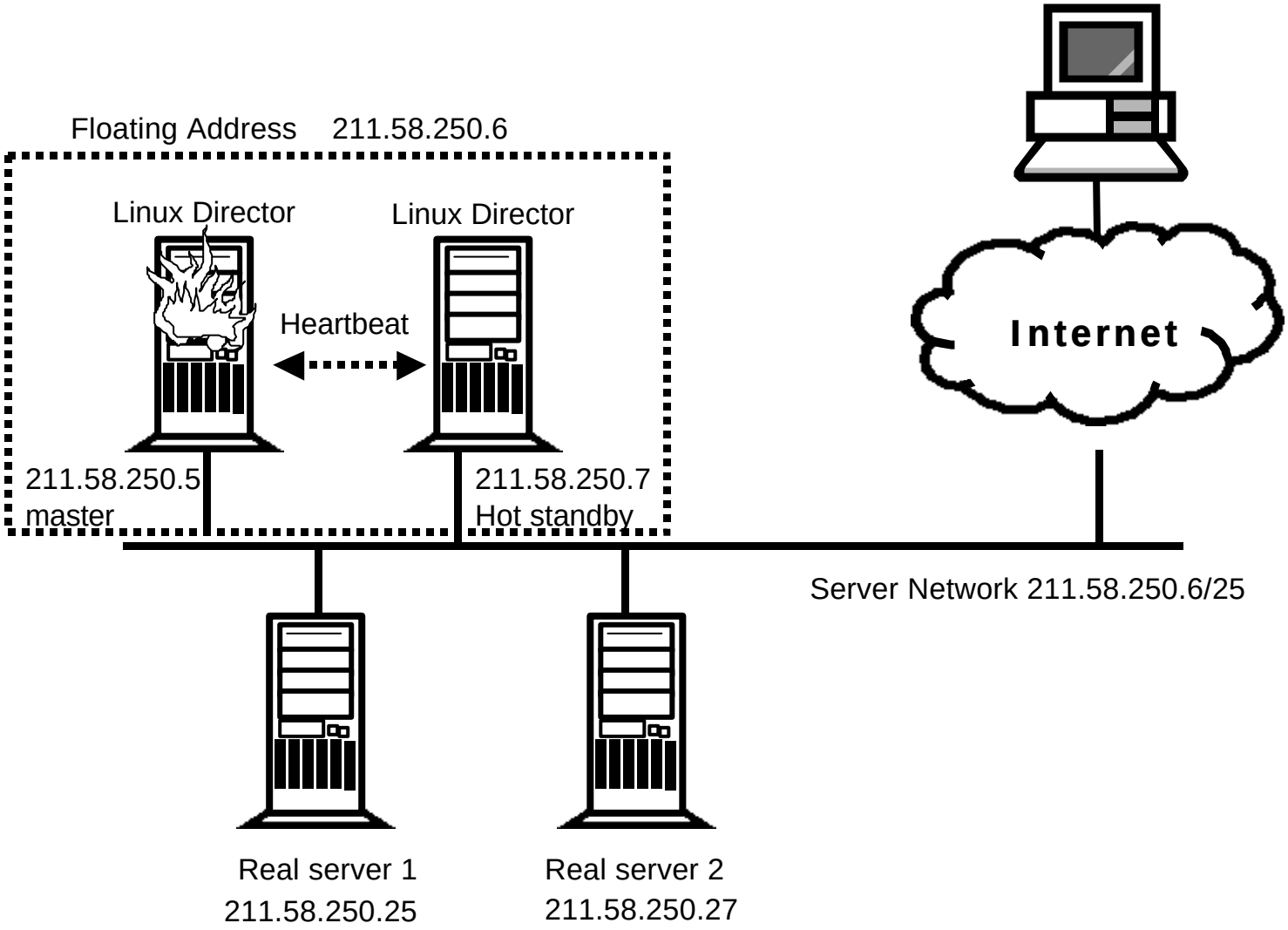
■ Ha1 의 /var/log/ldirectord.log

```
...
[Fri Nov 17 13:29:38 2000|ldirectord.cf] Adding real server:
211.58.250.25:80 (2*211.58.250.6:80)
```

■ Ha1 의 ipvsadm

```
# ipvsadm
IP Virtual Server version 0.9.12 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn
InActConn
TCP  ha.wowlinux.com:www rr
  -> www1.wowlinux.com:www      Route   1      0          0
  -> www2.wowlinux.com:www      Route   1      0          0
```

테스트 시나리오 (2) – master Director 다운/복구



테스트 시나리오 (2) – master Director 다운

■ Ha1 의 heartbeat 중지

```
[root@ha1 /root]# /etc/rc.d/init.d/heartbeat stop
Stopping High-Availability services: [ OK ]
```

■ Ha2 의 /var/log/ha-log

```
...
heartbeat: 2000/11/17_13:10:55 Taking over resource group
211.58.250.6
heartbeat: 2000/11/17_13:10:55 Acquiring resource group:
ha1.wowlinux.com 211.58.250.6 ldirectord
heartbeat: 2000/11/17_13:10:55 INFO: Running
/etc/ha.d/resource.d/ldirectord start
heartbeat: 2000/11/17_13:10:55 INFO: Running
/etc/ha.d/resource.d/IPaddr 211.58.250.6 start
heartbeat: 2000/11/17_13:10:56 INFO: ifconfig eth0:0 211.58.250.6
netmask 255.255.255.128 broadcast 211.58.250.127
heartbeat: 2000/11/17_13:10:56 Sending Gratuitous Arp for
211.58.250.6 on eth0:0 [eth0]
```

테스트 시나리오 (2) – master Director 복구

■ Ha1 의 heartbeat 복구

```
[root@ha1 /root]# /etc/rc.d/init.d/heartbeat start
Starting High-Availability services: [ OK ]
```

■ Ha2 의 /var/log/ha-log

```
...
heartbeat: 2000/11/17_21:13:37 Releasing resource group:
ha1.wowlinux.com 211.58.250.6 ldirectord
heartbeat: 2000/11/17_21:13:37 INFO: Running
/etc/ha.d/resource.d/IPaddr 211.58.250.6 stop
heartbeat: 2000/11/17_21:13:37 IP Address 211.58.250.6 released
heartbeat: 2000/11/17_21:13:37 INFO: Running
/etc/ha.d/resource.d/ldirectord stop
```

테스트 시나리오 (2) – master Director 복구

■ Ha1 의 /var/log/ha-log

```
heartbeat: 2000/11/17_21:18:01 received ip-request-resp
211.58.250.6 OK
heartbeat: 2000/11/17_21:18:01 Acquiring resource group:
ha1.wowlinux.com 211.58.250.6 ldirectord
heartbeat: 2000/11/17_21:18:01 INFO: Running
/etc/ha.d/resource.d/ldirectord  start
heartbeat: 2000/11/17_21:18:01 INFO: Running
/etc/ha.d/resource.d/IPaddr 211.58.250.6 start
heartbeat: 2000/11/17_21:18:01 INFO: ifconfig eth0:0 211.58.250.6
netmask 255.255.255.128     broadcast 211.58.250.127
heartbeat: 2000/11/17_21:18:01 Sending Gratuitous Arp for
211.58.250.6 on eth0:0 [eth0]
```

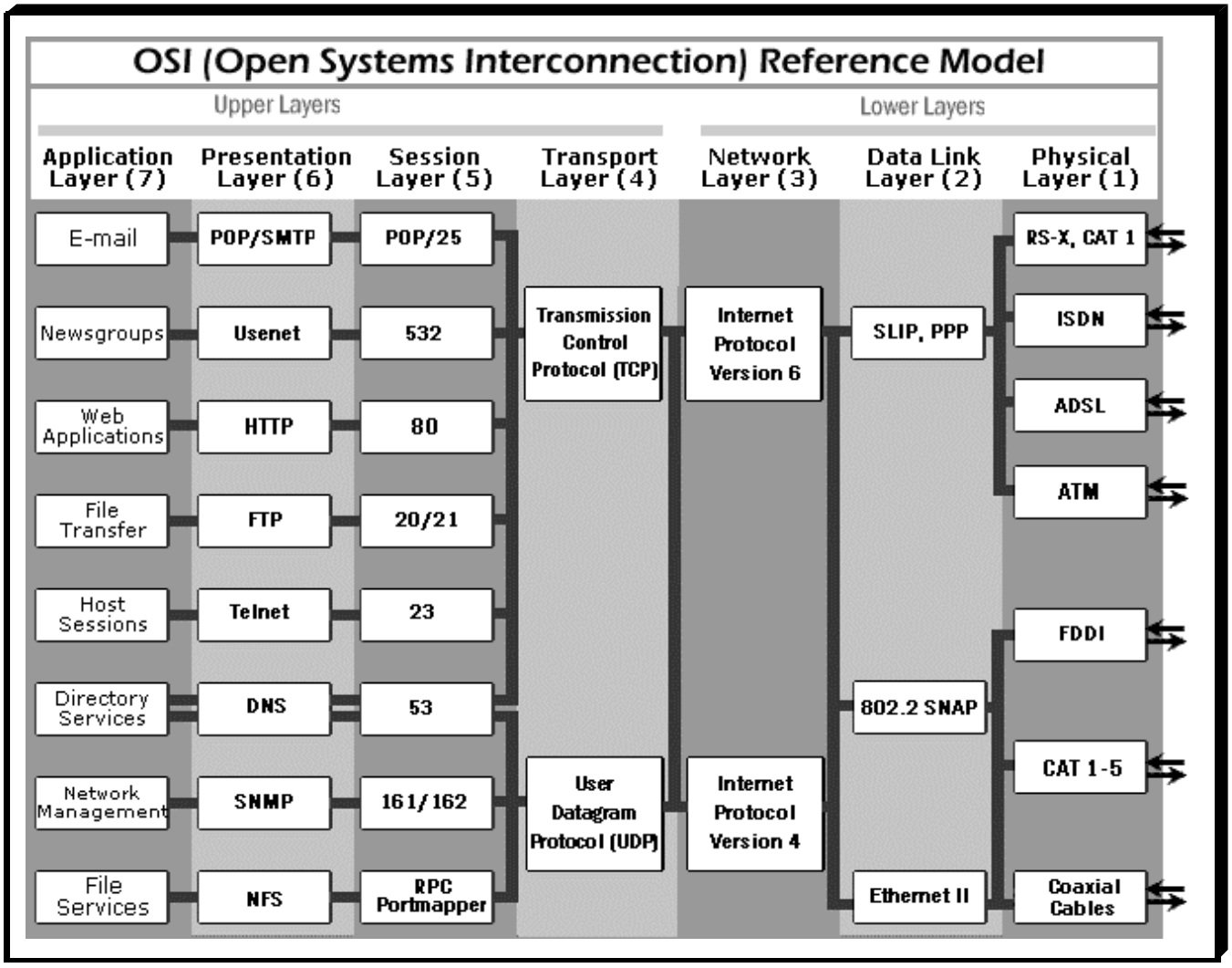
개선 필요 사항

- 단순화된 인스톨러
- GUI 기반 관리툴
- WEB 기반의 관리툴
- Shared File system
  - GFS
  - Rsync

참고 사이트

- 리눅스 가상 서버 프로젝트
  - http://www.linuxvirtualserver.org
  - http://tunelinux.pe.kr/virtual/
- 가상 서버에 대한 간략한 사용글
  - http://kldp.org/KoreanDoc/html/Virtual_Server-KLDP/Virtual_Server-KLDP.html
- ULTRA MONKEY
  - http://www.au.ultramonkey.org/
- High Availability systems under Linux
  - http://www.linuxfocus.org/English/November2000/article179.shtml
- Internet Core Protocols: The Definitive Guide
  - O'reilly , Erick A. hall
- 마이크로 소프트웨어 2000.7
  - 가상의 슈퍼컴, 클러스터링 파워
- 리눅스 에서 IP Aliasing 세팅하기
  - http://kldp.org/HOWTO/mini/html/IP-Alias/IP-Alias.html

참고 : Open Systems Interconnection Reference Model



참고 : OSI Reference Model

- 통신 네트워크로 구성된 컴퓨터가 어떻게 데이터를 전송할 것인가에 대한 표준규약
  - 7 계층 : 응용계층 ... 통신상대, 서비스 품질, 사용자 인증과 비밀을 고려하고, 데이터 구문의 제약을 규정
  - 6 계층 : 표현 계층 ... 운영체계의 한 부분으로 입력 또는 출력되는 데이터를 하나의 표현 형태에서 다른 표현 형태로 변환. 표현 계층을 문법 계층이라고 하기도 함
  - 5 계층 : 세션 계층 ... 종단 호스트 프로그램 사이에서 메시지를 주고받기 위한 설정을 하고, 데이터를 받는 동기를 제어하는 역할. 이 계층은 통신 세션을 구성하는 역할을 함
  - 4 계층 : 트랜스포트 계층 ... 종단간 제어와 에러를 관리. 즉, 신뢰성 있는 데이터 전송을 보장
  - 3 계층 : 네트워크 계층 ... 데이터 경로를 제어 (패킷이 정확한 수신자에게 보내지도록 올바른 경로는 제어하여 수신 쪽에서 받을 수 있게 함) 네트워크 계층은 경로를 설정하고 다른 쪽으로 전송
  - 2 계층 : 데이터링크 계층 ... 물리적 레벨의 에러 제어와 동기를 제공하고, 5를 초과하는 1의 스트링으로 비트화. 이 계층은 전송 확인과 관리를 담당
  - 1 계층 : 물리 계층 ... 전기 기계적으로 체계를 갖춘 네트워크를 통하여 비트 열을 전송. 이 계층은 전송 매체를 통해 데이터를 주고받는 하드웨어 수단을 제공.
- 참고 : Internetworking Basics
  - http://www.cisco.com/cpress/cc/td/cpress/fund/ith/ith01gb.htm

감사합니다!

http://wowlinux.com