김해중
KBS 보도기술국

실무 네트워크 Design - 12 : SLB, Server Load Balancing

이번 시간에는 SLB(Server Load Balancing)에 대해서 알아보려고 합니다. 대부분 방송국의 NPS 시스템은 동일한 기능을 하는 여러 대의 서버를 가지고 있습니다. 예를 들면, 여러 대의 WEB 서버, FTP 서버, streaming 서버, WAS 서버, transcoding 서버가 존재합니다. 여러 대의 서버를 이용해 많은 client의 요청을 처리할 수 있으며, 장애를 대비한 2중화 기능도 제공하게 됩니다. 그럼 어떻게 여러 대의 서버들 간에 균등한 부하분산(load-balancing)을 구현할 수 있을까요? 일반적으로 대부분의 회사는 3가지 방식을 사용합니다.

첫 번째는 DNS 서버를 이용하는 방식과 두 번째는 OS(windows, linux)가 자체적으로 제공하는 cluster 부하 분산을 이용하는 방식이며, 세 번째는 별도의 하드웨어 장비(부하 분산장비, 국내에서는 L4 switch라고 주로 부름)를 이용하는 방식입니다.

현재 가장 많이 사용하는 방식은, 부하 분산을 위한 전용 장비를 사용하는 방식입니다. 이 방식이 비용은 많이 들지만, 복잡한 트래픽을 가장 유연하게 처리할 수 있습니다.

이 글에서는 3가지 부하분산 방식의 특징을 간단히 정리한 후에, 가장 많이 사용하는 부하 분산장치에 대해 자세히 설명을 하겠습니다.

SLB 방법 3가지

방법 1. DNS Round robin

Round robin은 "순서대로, 즉 교대로"라는 의미입니다. DNS Round robin은 1개의 domain name에 대해서, 여러 개의 IP 주소를 DNS 서버에 등록해두고, client로부터 DNS 서버에 요청이 오면, 등록된 IP 주소를 순서대로 전달하는 것입니다. 전달되는 IP 주소가 바뀌기 때문에, 각 client가 접속하는 IP 주소도 다르게 됩니다.

DNS Round robin은 DNS 서버에 설정만 조금 추가하면, 트래픽 분산을 할 수 있으며, 별도의 장비를 구매하지 않아도 됩니다. 제가 KBS 대구에서 근무할 때는 이 방식을 사용했습니다.

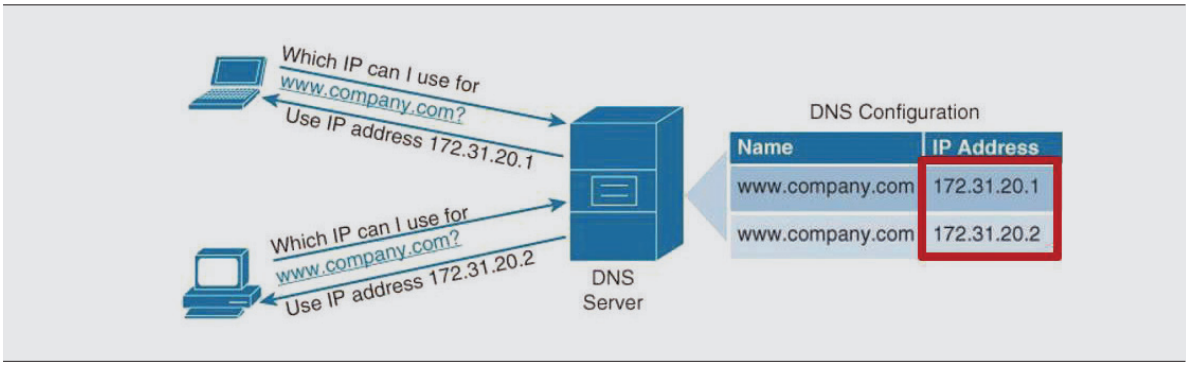


그림 1. DNS를 활용한 부하 분산

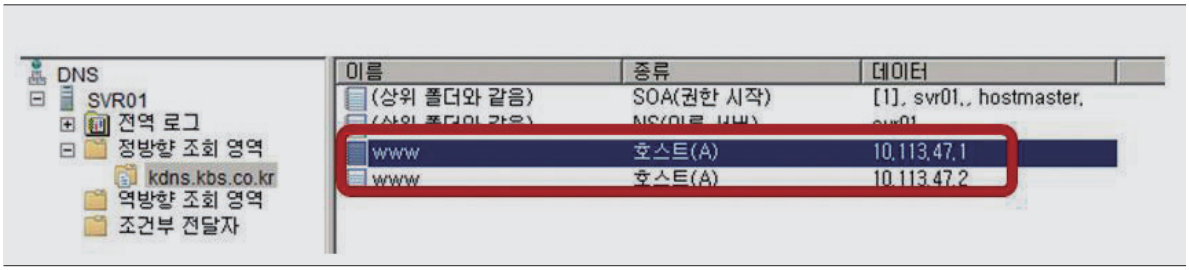


그림 2. windows DNS 서버의 Round robin 구성

그림 3. nslookup 명령을 통해서 google이 Round-robin을 사용하고 있음을 알 수 있습니다

하지만 이 방식은 몇 가지의 문제점을 가지고 있습니다. 첫 번째는 실제 서버에 장애가 발생해도, DNS 서버는 해당 서버가 장애가 발생했다는 사실을 알 수 없습니다. 결국 DNS 서버는 client에, 장애가 발생한 서버의 IP 주소를 계속 전달하게 됩니다.

두 번째는 균등한 load-balancing을 구현하기 어렵습니다. 처음에 client는 DNS 서버에 접속을 한 후에 얻은 정보를 cache에 보관을 하게 됩니다. 그래서 cache 정보가 사라지기 전까지는 계속 같은 서버에 접속을 하다 보니, 서버 간에 균등한 load-balancing을 구현하기 어렵습니다.

세 번째는 client의 traffic type을 반영하지 못 합니다, 즉 client의 device(tablet, phone, PC) 구분을 못해서, 해당 서비스를 제공하는 최상의 서버를 정확히 선택을 못합니다.

방법 2. OS 자체 Load-balancing 기능

OS(windows, linux)가 자체적으로 가지는 기능을 통해서 부하 분산을 구현할 수 있습니다. windows에는 NLB(network load balancing) 기능이 탑재되어 있고, Linux는 LVS(linux virtual server)를 지원합니다. 양쪽 모두 OS의 부속기능으로 제공하고 있어서, 적은 비용으로 쉽게 부하 분산을 구현할 수 있습니다.

window에서 제공하는 NLB에 대해서만 간단히 설명하도록 하겠습니다. NLB는 동일한 서비스를 제공하는 여러 대의 서버를 1개의 cluster로 구성한 후, cluster에 V-IP(virtual-IP)와 V-MAC(virtual-MAC)을 할당합니다.

client가 보낸 요청은 모든 서버로 전달되며, NLB가 사용하는 알고리즘에 의해 선택된 서버만, 해당 요청을 처리하고, 나머지 서버는 해당 요청을 무시하는 방식으로 동작합니다. 어떻게 보면, cluster에 참여한 모든 서버에 traffic이 전달되기에(마치 ARP와 유사하게) 참 비효율적인 방식입니다.

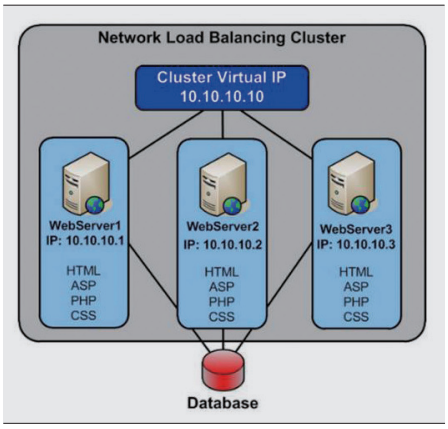


그림 4. windows NLB : cluster를 구성하고, cluster에 V-IP(virtual IP)를 할당합니다

김해중
KBS 보도기술국

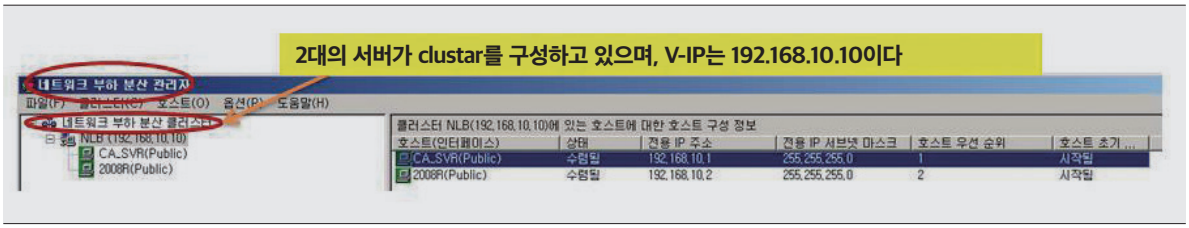


그림 5. OS(window 2008 R2)에서 지원하는 NLB

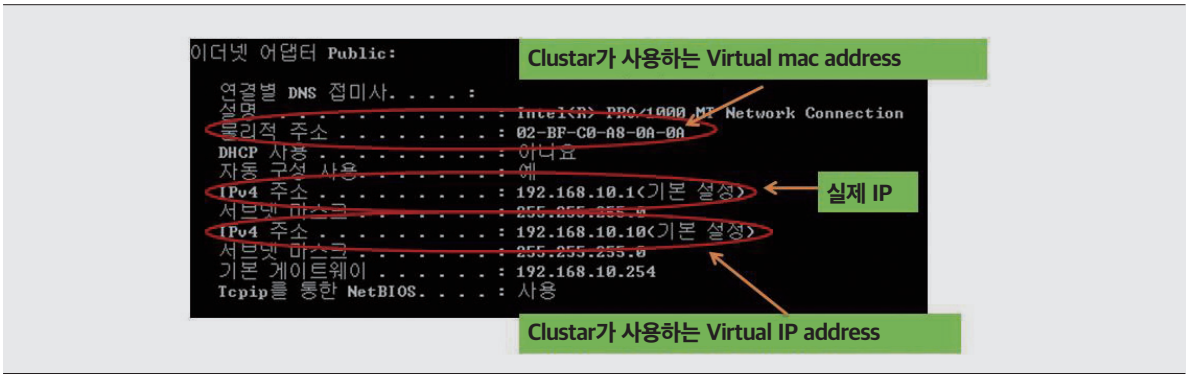


그림 6. Cluster가 사용하는 virtual mac-address와 virtual-IP

방법 3. Load-balancer (부하 분산장비)

별도의 H/W 장비인 부하 분산장비를 구매하여 부하 분산을 구현할 수 있습니다. 대표적인 장비로는 F5의 BIG-IP, Citrix의 NetScale, Cisco의 Ace 4700, 파이오 링크의 PAS, 라드웨어의 alteon이 있습니다. 참고로 국내에서는 부하 분산장치라는 이름보다는 L4 switch로 많이 부르고 있습니다.

이 방식은 장비를 별도로 구매해야 하므로 비용이 발생하지만, 부하분산에 특화된 장비라서, 복잡한 트래픽을 유연하게 처리할 수 있다는 장점을 가지고 있습니다. client의 목적지 IP는 real-server의 IP가 아니라, 부하 분산장치의 virtual-IP(V-IP)입니다. 즉 client로부터 오는 모든 traffic을 일단 부하 분산장비가 수신한 후에, 그것을 뒷단에 있는 실제 서버에 전달합니다.

실제 대부분의 회사들이 부하 분산장치를 이용하기에, 이제 부터는 부하 분산장치만을 기준으로 두고 설명을 하겠습니다.

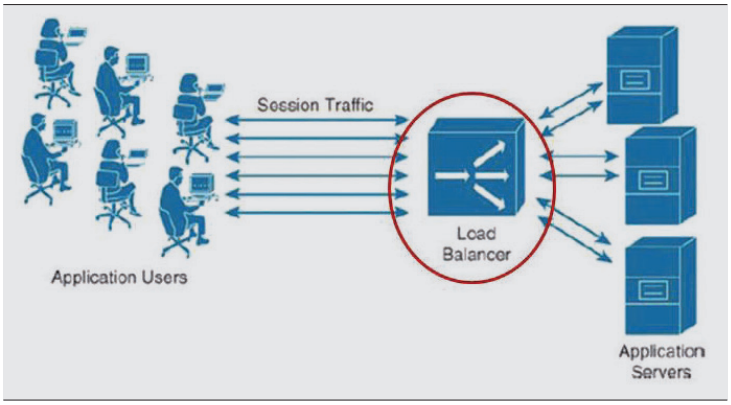


그림 7. 부하 분산장비의 동작방식 : client로부터 오는 traffic을 수신한 후에, 뒷단에 있는 실제 서버에 전달합니다

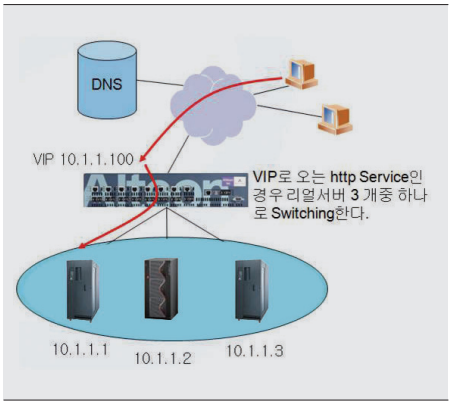


그림 8. 실제 Client 요청의 목적지 IP는 부하 분산장비의 V-IP(virtual-IP)이다

Load-balancer(부하 분산장치)의 부하 분산기법

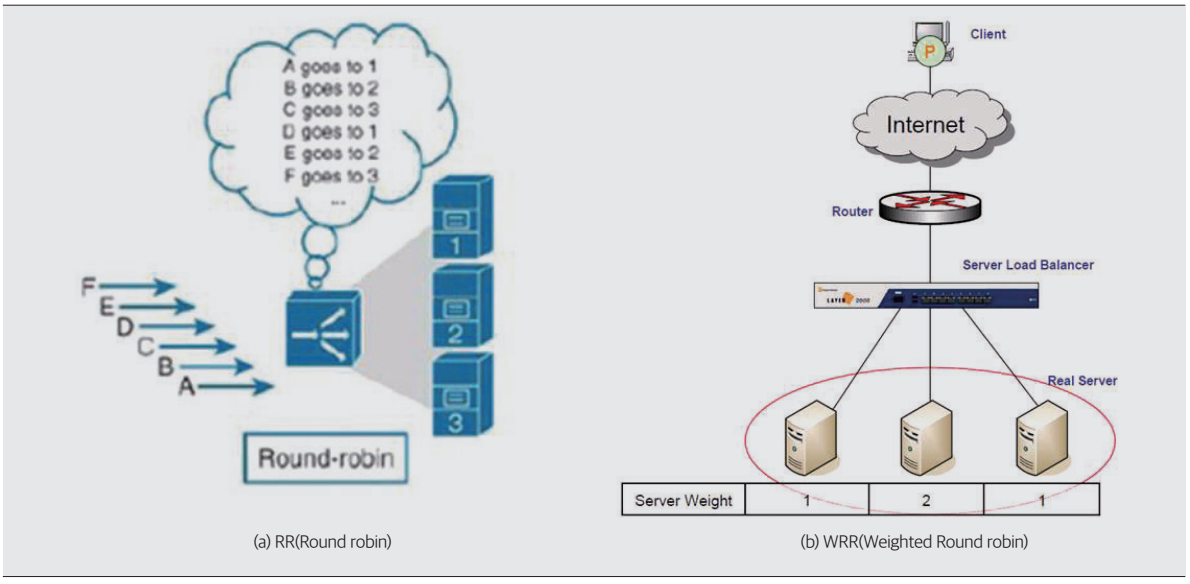
어떻게 부하분산을 구현하는지 부하 분산기법을 알아볼까요? 부하 분산기법은 크게 static과 dynamic 방식으로 구분할 수 있습니다. static은 서버의 현재 상태를 고려하지 않고, dynamic은 서버의 현재 상태를 반영하는 것입니다. 먼저 static 방식을 보도록 하겠습니다.

Static 방식 부하 분산기법 (3가지)

첫 번째는 RR(Round Robin) 방식입니다. 부하 분산장치는 client로부터 받은 요청을, 부하분산 대상 서버에 순서대로 할당하는 방식입니다. 부하분산 대상 서버의 성능이 동일하고, 처리시간이 짧은 application(예 : web 서비스)일 때는, 균등한 부하분산이 이루어집니다. 하지만 처리시간이 긴 ftp 등의 서비스 환경에서는 맞지 않습니다.

두 번째는 WRR(Weighted Round Robin) 방식입니다. 서버별로 가중치를 설정해두고, 그 가중치(weight)에 따라 요청을 서버에 할당하는 방식입니다. 성능이 낮은 서버에 낮은 가중치를 설정하고, 성능이 높은 서버에 높은 가중치를 설정합니다. 이 방식은 부하분산 대상 서버의 성능에 차이가 있을 때 사용합니다.

세 번째는 active/standby 방식입니다. 즉 평상시에는 active 장비만 사용하고, active 장비에 장애가 발생했을 때만, standby 장비를 사용합니다. 즉 이 방식은 부하분산이라기 보다는 서버 2중화를 위한 기능입니다.



(a) RR(Round robin) (b) WRR(Weighted Round robin)

그림 9. 부하 분산장비의 static 부하분산 기법들

대표적인 Dynamic 부하 분산기법(3가지)

첫 번째는 LC(Least Connection) 방식입니다. 부하 분산장치가 요청을 받는 시점에서 가장 연결 수가 적은 서버를 선택하여 요청을 할당합니다. 연결을 길게 지속해야하는 애플리케이션을 부하분산 할 때 적합합니다.

두 번째는 Fastest(최단 응답시간)방식입니다. 부하 분산장치는 client로부터 받은 요청과 서버의 응답 사이의 시간을 계속 확인합니다. 이때 부하 분산장치는 요청을 받으면 가장 빠르게 응답하는 서버를 선택해서 할당하는 방식입니다.

세 번째는 LL(Least Loaded) 방식입니다. 각 서버는 주기적으로 CPU, 메모리 사용량 정보를 SNMP(simple network management

김해중
KBS 보도기술국

protocol)을 이용해서 부하 분산장치에 알려줍니다. 부하 분산장치가 요청을 받으면 취득한 정보를 바탕으로, 부하가 가장 적은 서버에 할당합니다. 다른 방식보다 정보의 신뢰성은 높지만, 그 정보가 실시간 정보가 아니라는 문제점이 있습니다. 물론, 이것을 사용하기 위해서는 서버에 SNMP agent가 설치되어 있어야 하며, SNMP에서 사용하는 MIB(management information base)의 값이 해당 서버의 부하(load)를 표현할 수 있어야 합니다.

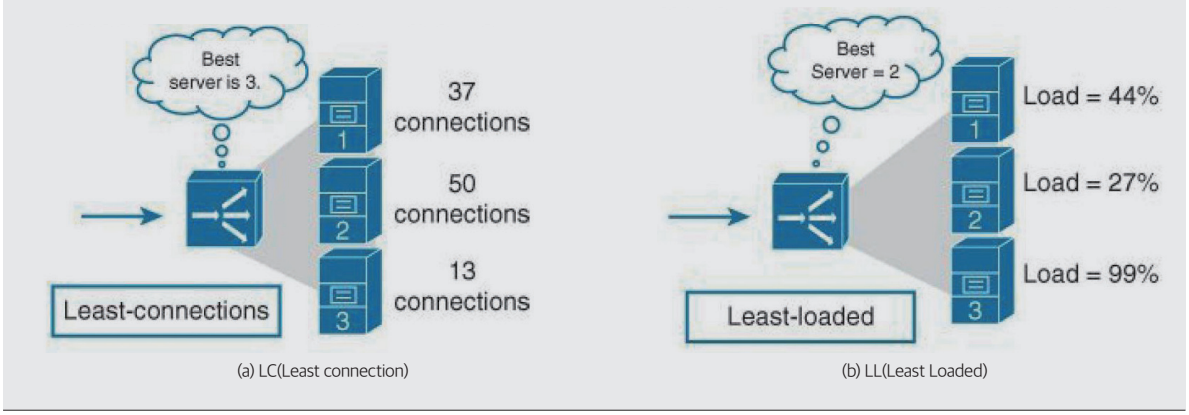


그림 10. 부하 분산장비의 Dynamic 부하 분산기법들

Load-balancer(부하 분산장치)의 서버 모니터링

부하 분산장치는 서버의 부하를 분산할 뿐만 아니라, 부하 분산하는 서버의 상태도 항상 감시하고 있습니다. 장애가 발생했을 때는, 해당 서버를 격리시킵니다. 서버 감시기능에는 'L3 체크', 'L4 체크', 'L7 체크' 세 가지가 있습니다. 체크 계층이 높을수록, 더욱 상세하고 유연한 서버감시가 가능하지만, 서버 부하가 올라간다는 문제점을 가집니다.

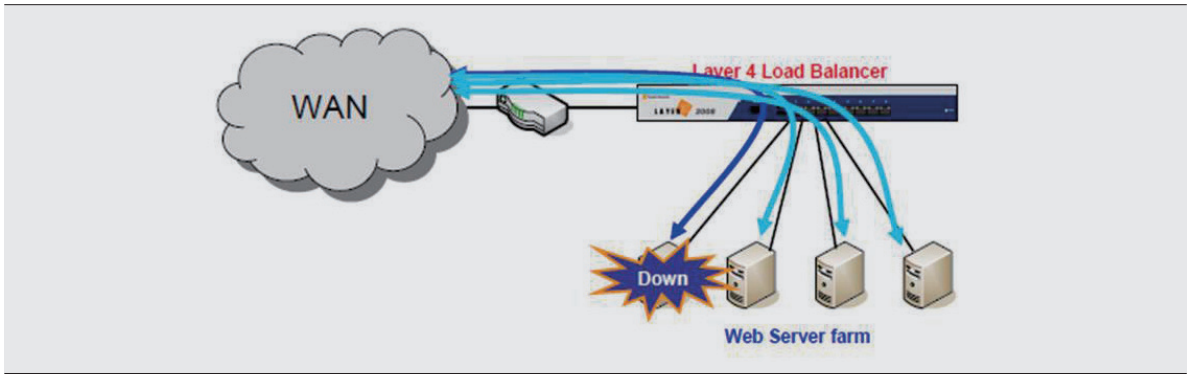


그림 11. 부하 분산장비는 부하를 분산하는 서버의 상태를 모니터링 한다

L3 체크

L3 체크는 서버의 IP가 살아있는지 체크하기 위해 ping(ICMP)을 이용하는 방식입니다. 예를 들어 LAN cable이 빠져 있거나 하면, ping 응답이 오지 않을 것입니다. 이럴 때 부하 분산장치는 이를 감지하고 서버를 격리시키게 됩니다. L3 체크가 실패하면, L4 이상 모든 서비스가 동작하지 않기에, L4, L7 체크도 실패합니다.

L4 체크

L4 체크는 서비스와 관련 있는 TCP/UDP port 번호를 감시하는 기능입니다. 예를 들어서 web 서비스는 80번 포트를 이용해 통신합니다. 부하 분산장치는 이 80번 port에 주기적으로 TCP syn 패킷을 보내서 응답이 있는지 확인합니다. 웹 서비스를 제공하는 Apache, IIS 등에 장애가 발생해서 서비스가 다운되면 응답이 사라지는데, 그때 부하 분산장치가 이를 발견해서 서버를 격리시킵니다.

L7 체크

L7 체크는 OSI 7계층(애플리케이션 계층)까지 감시하는 기능입니다. 예를 들어 웹 애플리케이션이 정상적으로 동작하고 있으면 'OK'라는 응답을 하고, 애플리케이션에 문제가 있으면 'error 응답'을 보냅니다. 이때 error 응답을 받으면, 서버를 서비스로부터 격리시킵니다.

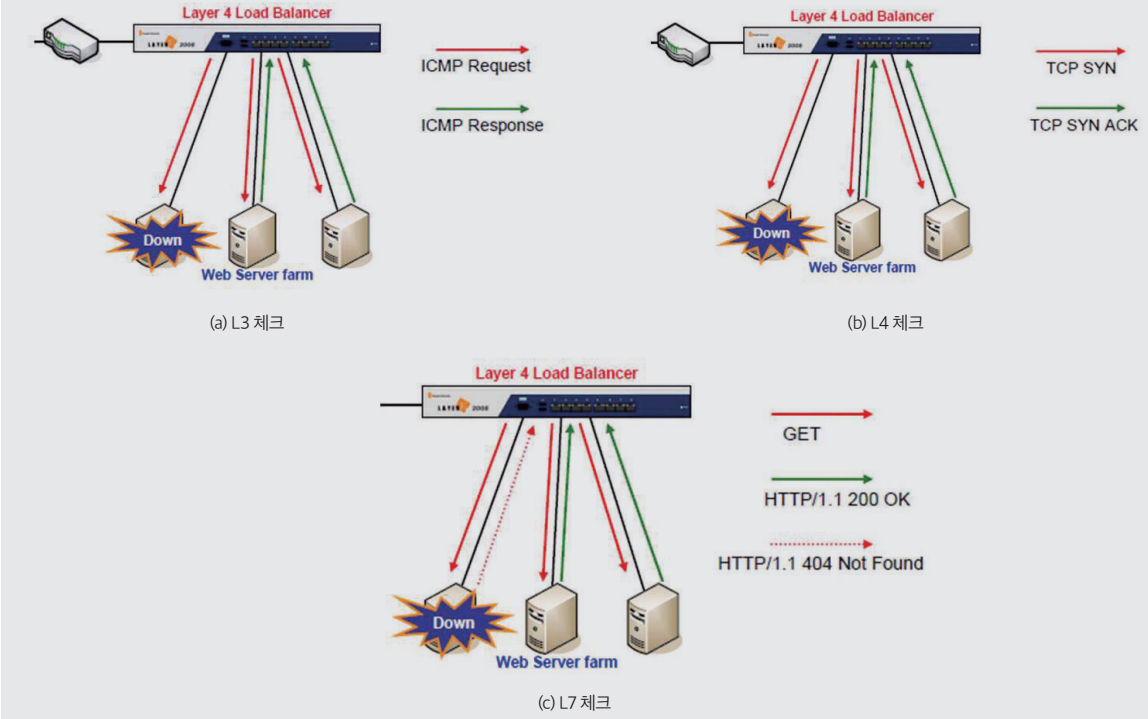


그림 12. 부하 분산장비는 3가지 방식의 서버 모니터링을 지원합니다

Load-balancer(부하 분산장비) 물리적 구성

부하 분산장비를 물리적으로 배치하는 방식은 크게 "two-arm(In-line)" 방식과 "one-arm" 방식이 존재합니다.

Two-arm(In-line)

two-arm 방식은 네트워크에 부하 분산장치를 끼워 넣는 방식으로, 즉 부하 분산장치가 server farm switch와 server를 두 팔을 펼친듯한 모양으로 연결하기에 "two-arm"으로 부릅니다. 또한 네트워크에 끼워 넣는다는 의미에서 "in-line" 구성으로 부릅니다. 이 방식은 구성이 간단해서 문제가 발생했을 때, 장애 대처가 쉽고 관리가 쉬워 현재 많이 사용되고 있습니다. 그러나 모든 트래픽이 부하 분산장치를 경유한다는 문제점을 가지고 있습니다.

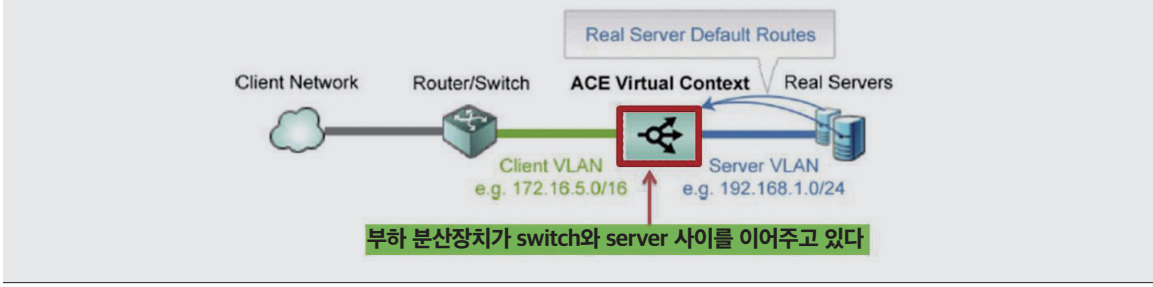


그림 13. two-arm 모드의 물리적 배치

김해중
KBS 보도기술국

One-arm

One-arm 방식은 부하 분산장비를 server에 직접적으로 연결하지 않고, server farm switch에만 연결하는 방식입니다. 부하 분산장비가 서버에 연결되지 않기에, 배치가 자유롭다는 장점이 있습니다. 하지만 구성이 복잡하며, 트래픽 흐름이 직관적이지 않기에 관리자들이 선호하지 않습니다.

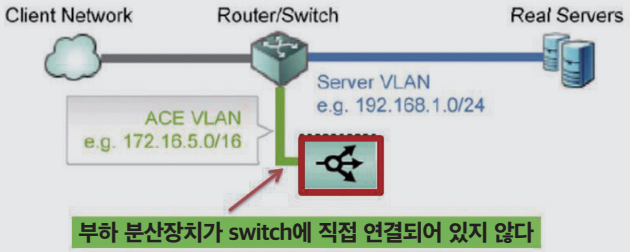


그림 14. one-arm 모드의 물리적 배치

Load-balancer(부하 분산장치) 동작방식(4가지)

Routed Mode

"routed mode"는 부하분산 장치 상단의 IP 대역과 하단의 IP 대역이 다릅니다. 이때 server의 default-gateway는 부하 분산장치가 되고, 부하 분산장치의 default-gateway는 server farm switch로 구성하는 것이 일반적입니다.

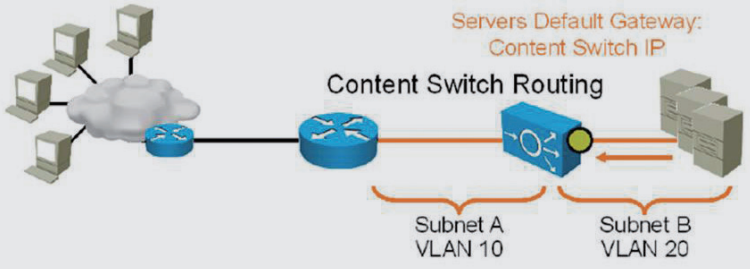


그림 15. routed mode : 부하분산 장치 상단과 하단 네트워크의 IP 대역이 다릅니다

client가 보낸 요청 패킷이 부하 분산장치의 목적지 IP(V-IP)로 도착하면, 부하 분산장치는 D-NAT(destination-Network address translation) 기술을 이용해서 목적지 IP(V-IP)를 실제 server의 Real-IP(R-IP)로 변경합니다. 반대로 server가 보낸 모든 패킷이 부하 분산장치를 통과할 때는 패킷의 출발지의 Real-IP(R-IP)를 부하 분산장치의 V-IP로 변경하는 S-NAT(source-NAT)을 구현합니다.

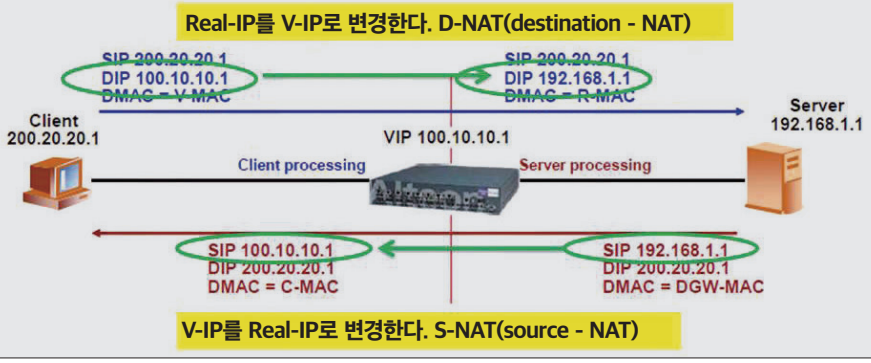


그림 16. 부하 분산장치는 NAT 기술을 이용해 Real-IP와 Virtual-IP를 매치합니다

Bridge Mode

Bridge mode는 부하 분산장치를 L2 switch처럼 구성하는 방식입니다. L2라서 부하 분산장치 상단의 네트워크와 하단의 네트워크 대역이 동일합니다. 그래서 IP 주소를 변경할 필요 없이 부하 분산장치를 배치할 수 있습니다. server의 default-gateway의 IP는 server farm switch로 설정합니다. 앞에서 설명한 routed-mode와 동일하게 요청 패킷은 D-NAT이 발생하고, 응답 패킷은 S-NAT이 발생합니다.

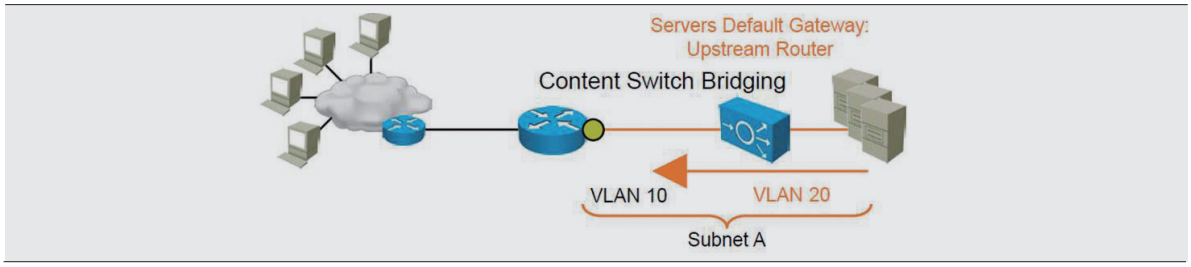


그림 17. bridge mode : server의 default-gateway는 server farm switch이다

One-arm

one-arm 구조는 앞에서 설명한 routed, bridge mode와 달리, 부하 분산장치의 위치가 server와 직접 연결이 안 된 구성입니다. 그래서 client가 보낸 패킷이 부하분산 장치에 도착했을 때, 목적지뿐만 아니라 출발지 주소까지도 변경하게 됩니다. 즉 Dual-NAT(출발지, 목적지 주소 모두 변경)이 동작합니다.

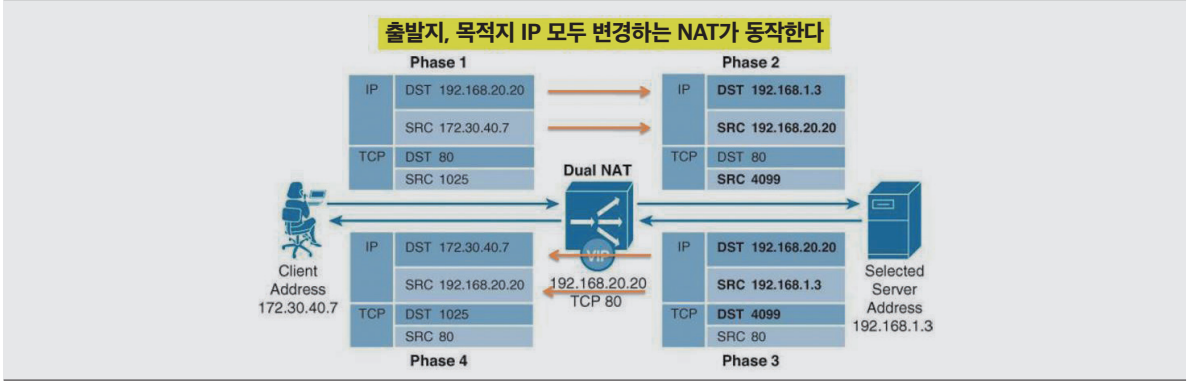


그림 18. one-arm 구조는 일반적으로 Dual-NAT를 사용한다

하지만 특정 부하 분산장치(예 : cisco ACE)는, client가 보낸 요청 패킷이 부하 분산장치에 도착했을 때, Dual-NAT을 사용하지 않고 목적지 주소만을 변경합니다. 그래서 server가 보낸 응답패킷이 server farm switch에 도착했을 때, PBR(policy-based routing)을 통해, 부하 분산장치로 보내서, 패킷이 부하 분산장치를 경유하게 합니다. 참고로 PBR이란 라우터에 특정 패킷이 들어오면 routing table을 보기 전에, 먼저 특정 정책(예를 들면 특정 패킷을 어디로 보내라)을 적용할 때 사용합니다. 이 경우는 서버로부터 패킷이 들어오면 부하 분산장비로 보내도록 PBR을 세팅합니다.

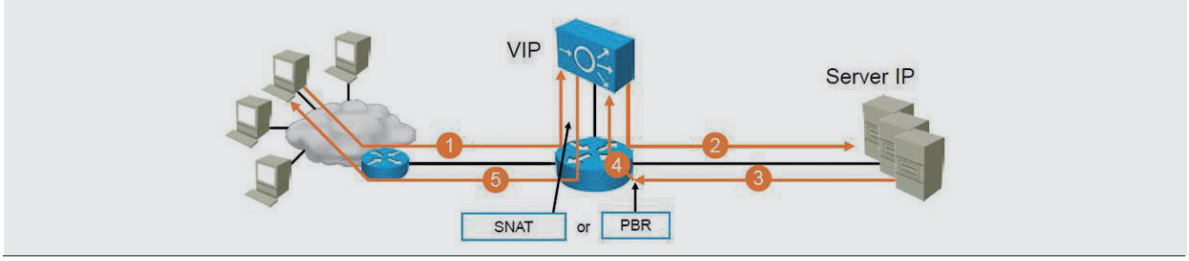


그림 19. one-arm 구조의 traffic 흐름

김해중
KBS 보도기술국

DSR(Direct Server Return)

DSR은 one-arm 구조를 약간 변형한 방식으로, server가 client에 보내는 응답패킷이 부하 분산장치를 경유하지 않고, 직접 client에 보내는 방식입니다. DSR에서 부하 분산장비는 client의 요청만 처리하면 되기에 부하가 많이 감소됩니다. 이 방식은 web과 같이 download가 많은 비대칭 환경에 적합합니다. 하지만 응답 traffic이 부하 분산장비를 경유하지 않기에 세션관리가 안 된다는 문제가 존재합니다.

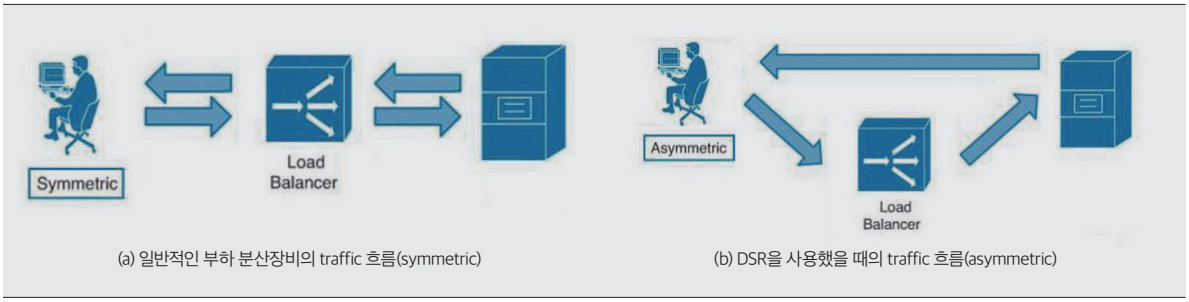


그림 20. DSR은 비 대칭 traffic 환경에 적합하다

DSR을 사용할 때 부하 분산장비는 자신이 받은 요청 패킷을 그냥 relay해서 실제 서버에 전달하는 기능만 합니다. 즉 앞의 설명한 방식들과 다르게, IP 주소를 변경하는 NAT를 전혀 하지 않습니다. 단지 실제 서버에 전달하기 위해서, 패킷의 목적지 mac 주소를 실제 서버의 mac 주소로 변경만 합니다. 이것을 MAT(mac-address translation)이라고 표현합니다.

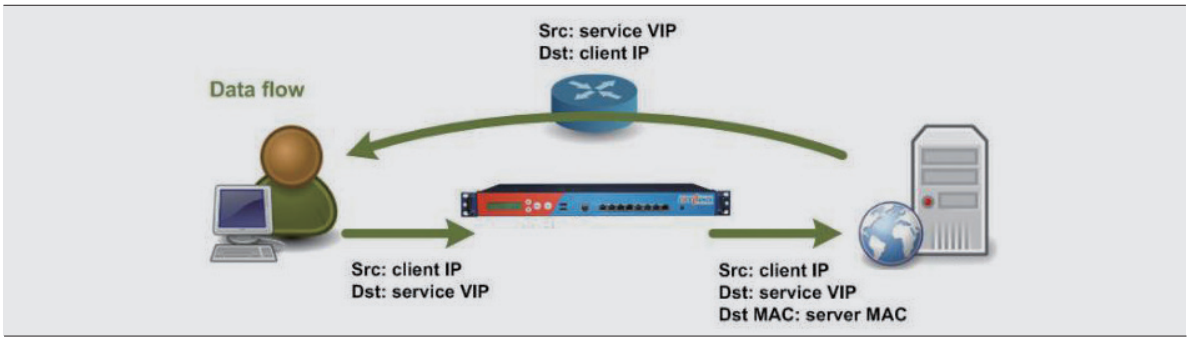


그림 21. DSR은 NAT을 하지 않고, MAT(mac address translation)을 합니다

부하 분산장비가 서버에 전달한 패킷을 서버가 수신하기 위해서는 서버가 부하 분산장비의 V-IP를 가져야 됩니다. 그래서 부하 분산장비가 사용하는 V-IP를 서버의 loopback interface에 반드시 할당해야 합니다. 그리고 서버의 loopback은 절대로 ARP에 응답하지 않도록 세팅해야 합니다.

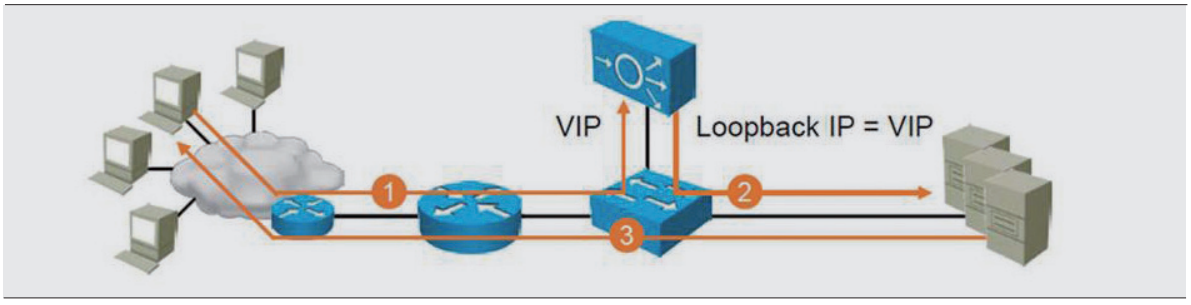


그림 22. DSR을 사용할 때는 서버에 세팅을 해야 한다는 불편함이 존재합니다

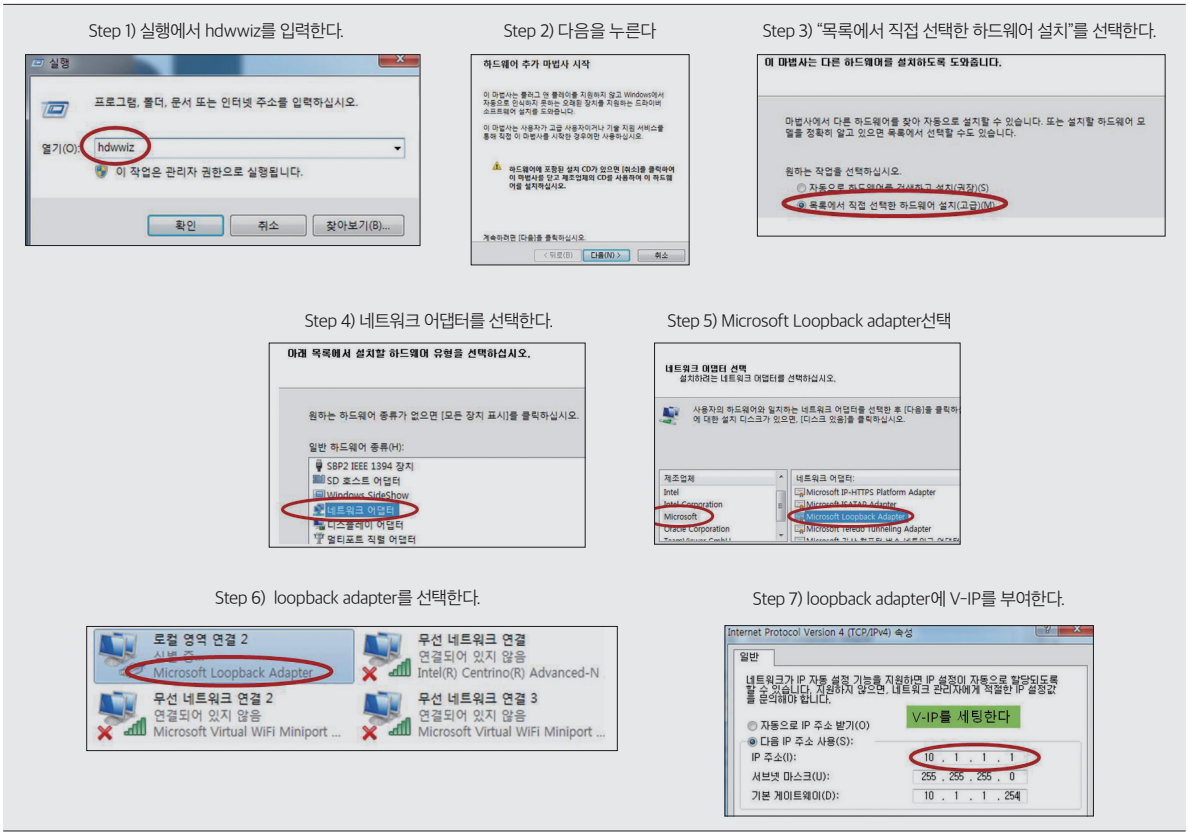


그림 23. DSR을 사용하기 위해서는 server에 loopback adapter를 만들어야 합니다

```
C:₩Users₩haejung>netsh interface ipvr set interface "loopback" metric=254
```

그림 24. DSR을 사용할 때는 서버가 V-IP에 대한 ARP에 응답하지 않도록 metric을 크게 합니다

동작 방식(4가지) 비교

부하 분산장비는 실제로 NAT을 이용해, IP를 변경하는 방식으로 동작합니다. 하지만 DSR은 NAT을 사용하지 않고, MAT(mac-address translation)만을 사용합니다. 그래서 DSR은 다른 기술과 달리, 서버에서의 불편한 IP 세팅을 필요로 합니다.

mode 종류	Destination-NAT 사용 여부	Source-NAT 사용 여부
Bridge-mode	○	○
Routed-mode	○	○
One-arm	○	○
DSR	×	×

맺음말

이번 시간에는 부하 분산장치를 이용한 SLB에 대해서, 핵심적인 내용만 설명했습니다. 사실 국내에서는 부하 분산장치라는 용어보다는, L4 switch라고 거의 부릅니다. 대부분 회사의 NPS 사업 때에 들어간 대부분의 장비의 목록에도 L4 switch라고 표현되어 있을 것입니다. 과거에 부하 분산장치는 L4 switch 역할을 했다면, 현재의 부하 분산장치는 L4뿐만 아니라, L7 switch 기능도 제공하며, 추가적으로 다양한 부가기능을 제공해서, 단지 L4 switch라고 부르기에는 너무 부족한 표현인 것 같습니다.

사실 이번 시간에, 부하 분산장치의 L7 switch기능과 다양한 부가 기능, 세팅까지 설명 드리면 너무 복잡해질 것 같아서, 그것에 대한 설명은 생략했습니다^^. 다음에 기회가 되면 설명을 하겠습니다. 이번 시간에 정리된 내용을 통해서 부하 분산장비를 이해하시는 계기가 되었으면 좋겠습니다.