

---

# 클라우드 환경에서 Linux Virtual Server 로드밸런싱 구현

서경석 · 이봉환

대전대학교

Implementation of Linux Virtual Server Load Balancing in Cloud Environment

Kyung-Seok Seo · Bon-Hwan Lee

Daejeon University

E-mail : portars@naver.com

## 요 약

최근 에너지 소비의 지속적 증가 및 에너지 가격의 급격한 상승으로 그린IT 도입·운영이 필수적인 요소로 인식됨에 따라 서버 발열 및 데이터센터 에너지 절감을 위해IT 인프라는 클라우드 컴퓨팅 플랫폼으로 대체 되어가고 있다 본 연구에서는 일반 저비용 웹서비스 인프라를 오픈소스 기반 클라우드 플랫폼으로 변환하고 Linux Virtual Server 로드 밸런싱을 구현하여 그 성능을 분석하였다

## ABSTRACT

Recently adoption of the Green IT is regarded as an essential element in order to decrease server heat and save energy in data center because of continuous increase of energy consumption and energy price. Consequently the conventional IT infrastructure is replaced with cloud computing platform. In this paper, we have implemented a Linux virtual server load balancing in open source-based cloud platform and the performance of the LVS load balancing is analyzed.

## 키워드

클라우드, Linux Virtual Server, 로드밸런싱

## I. 서 론

글로벌 환경위기 즉, 온난화로 인해 국제적인 온실가스 감축 규제 및 각국의 환경규제 강화로 국가와 기업 모두가 환경 경영이라는 슬로건으로 기업 경영 마인드가 변화되고 있다 배출, 각종 유해물질 제한 등 환경 규제의 증가, 에너지 소비의 지속적 증가, 국제적 에너지 자원 가격의 급격한 상승으로 최근 각 기업에서는 친환경 경영을 위한 그린 IT 도입 운영 및 공급자 관리가 선택이 아닌 필수적 요소로 등장하게 되었다[1]. 모든 IT 관련 전력비용의 절반 이상이 IDC(Internet Data Center)에서 발생하고 있기 때문에 데이터 센터에 초점을 두고 있고 이 문제를 해결하기 위해서 클라우드 컴퓨팅 기술이 급부상하고 있다 클라우드는 새로운 컴퓨팅 환경에서 중요한 역할을 한다. 클라우드 컴퓨팅 플랫폼은 업무처리, 소프트웨어, 인프라, 응용프로그램을 공유할 수 있는 가상화 기술이 적용 가능하며 서버팜으로 구

성되는 클라우드 플랫폼의 자원을 공유하여 사용할 수 있다. 기존 웹서비스는 가용성을 위해 자원을 높게 잡아 유효자원 낭비가 발생하고 또한 전력낭비도 발생한다. 그러나 클라우드 컴퓨팅을 사용하면 많은 유효 자원을 90% 이상 사용하여 효율적으로 전체 자원 및 비용을 절감 시킬 수 있다. 전문가들은 실제적으로 클라우드 컴퓨팅을 통해 데이터 센터 공간을 80% 이상 줄일 수 있고, 에너지 소비량도 40% 가량 절감할 수 있다고 한다[2]. 따라서 클라우드 컴퓨팅은 그린 IT를 실현할 수 있는 친환경적인 플랫폼이다. 본 논문에서는 일반적인 방식의 호스팅서버들을 클라우드 컴퓨팅 플랫폼에 적용하고 Linux Virtual Server (LVS) 로드밸런싱을 이용하여 저비용으로 효과적인 트래픽 분산방법을 구현한다

## II. 관련 연구

### 가. OpenNebula[3]

OpenNebula는 Universidad Complutense de Madrid의 DSA-research Group에서 시작된 오픈소스 기반의 IaaS Management 툴킷으로 그림 1과 같은 구조를 가지고 있다. 클러스터링 관리 측면에서 다양한 종류의 네트워크 및 스토리지 클러스터 노드 등을 관리하고 private, public, 그리고 hybrid 등 모든 형태의 클라우드 환경 구축이 가능하다.

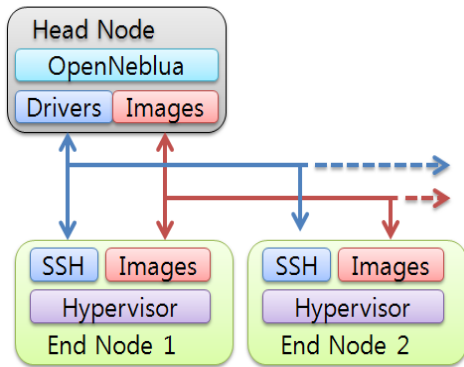


그림 1. OpenNebula의 구조

### 나. KVM (Kernel-based Virtual Machine)[4]

KVM은 리눅스에서 x86 하드웨어의 Intel VT 또는 AMD-V 가상화 기술을 이용하여 전가상화하는 솔루션이다. KVM을 이용한 하이퍼바이저는 그림 2와 같이 여러 모듈로 구성되고 KVM은 가장 하위 레벨에서 전가상화 동작을 하게 된다.



그림 2. KVM의 구조

### 다. HDFS (Hadoop Distributed File System)[5]

Apache Hadoop은 많은 컴퓨터들의 클러스터링을 통해 데이터 집중 분산 애플리케이션을 지원하는 오픈소스 소프트웨어 프레임워크이고 그림 3과 같은 구조로 애플리케이션이 수천 개의 노드를 이용하여 페타 바이트 단위의 데이터를 다루

는 것을 가능하게 한다. Hadoop 프로젝트는 원래 Nutch 검색 엔진 프로젝트의 분산 환경 지원을 하기 위해서 시작되었고 구글의 MapReduce와 구글 파일 시스템 연구에 영향을 받아 개발되었다. Yahoo는 하둡 개발에 많은 기여를 하였으며 이를 웹 검색과 광고 사업에 활용하고 있다. IBM과 구글은 하둡을 대학의 분산 컴퓨터 프로그래밍 과정에 포함하도록 지원하고 있다.

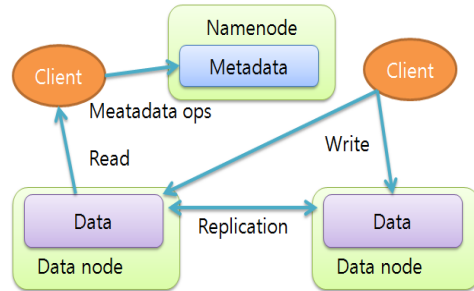


그림 3. HDFS의 구조

### 라. LVS (Linux Virtual Server)[6]

LVS는 한 대의 서버로 증가하는 인터넷 사용자를 처리하기가 힘들어 지면서 고가용성 서버를 묶어 한 대처럼 사용할 수 있도록 하는 부하 분산 솔루션이다.

- LVS 스케줄링

- 1) 라운드-로빈(round-robin)

라운드-로빈 방식은 로드밸런서에 들어오는 요청 패킷들을 차례대로 할당할 뿐이다. 하지만, 이렇게 하더라도 로드밸런싱을 위해 이전에 사용되던 라운드-로빈 DNS 방식에 의해 서버를 할당하는 방식에 비해서는 우수한데, DNS의 경우는 한번 서버가 지정되면 해당 서버에 수많은 요청 패킷이 몰릴 수 있기 때문이다.

- 2) 가중 라운드-로빈(weighted round-robin)

가중 라운드-로빈 방식은 기본적으로 라운드-로빈 방식인데, 각 서버에 서로 다른 가중치를 주어서 할당하는 방식이다. 이 방식을 사용해야 하는 경우는 실제 서버들이 CPU의 수와 성능, 메모리 용량 등 서로 다른 성능을 가지고 있어서, 각 서버를 동등하게 취급할 수 없는 경우이다.

- 3) 최소 연결(least connection)

최소 연결 방식은 실제 서버들 중에서 현재 가장 적은 수의 요청을 처리하고 있는 서버를 선택하여 요청패킷을 할당하는 방식이다. 이 방식은 실제 서버의 현재 부하 상황을 동적으로 판단하여 요청을 처리하기 때문에 앞의 두 방식에 비해서 동적으로 우

수한 부하 분산 효과를 얻을 수 있다

4) 가중 최소 연결(weighted least connection)

가중 최소 연결 방식은 기본적으로 최소 연결 방식인데 가중 라운드-로빈 방식과 마찬가지로 각 서버에 서로 다른 가중치를 주어서 할당하는 방식이다.

● LVS 인프라 구조

1) Direct Routing 방식

Direct Routing 방식은 그림 4와 같이 실제 서버가 직접 패킷을 전달하는 구조로 트래픽 분산에 우수하다.

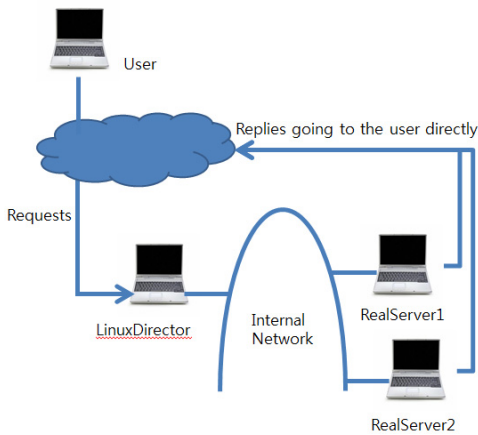


그림 4. Direct Routing의 구조

2) NAT 방식

NAT 방식은 그림 5와 같이 한 회선에서 요청과 응답이 이루어지기 때문에 트래픽 분산에는 취약하나 분산 처리 시 우수하다.

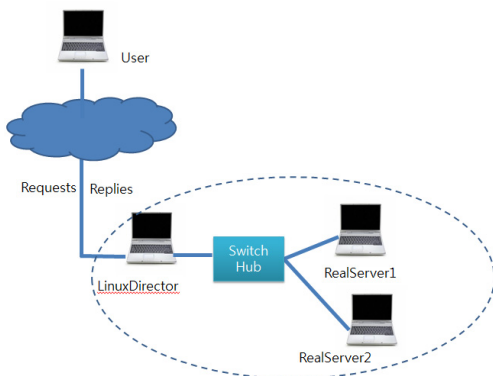


그림 5. NAT 방식의 구조

3) IP Tunneling 방식

IP Tunneling 방식은 그림 6과 같이

Tunneling을 이용하여 보안성이 높으며 일반적으로 사용된다.

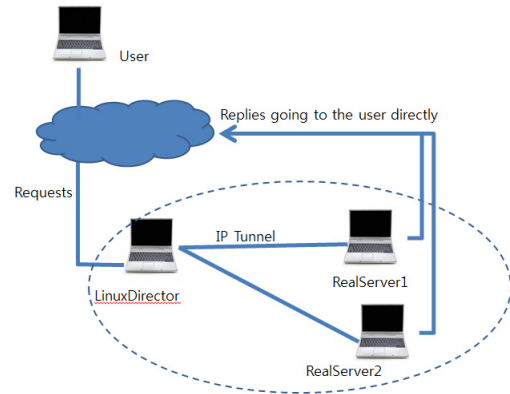


그림 6. IP Tunneling 방식의 구조

III. 클라우드 환경에서 LVS 구현

가. 클라우드 환경 구축

본 논문에서는 그림 7와 같이 OpenNebula, Hadoop, KVM을 이용하여 클라우드 플랫폼을 구축하였다. OpenNebula는 하이퍼바이저 및 컴퓨팅 자원 관리를 하고 KVM은 가상머신 생성을 하여 컴퓨팅 자원을 분배하게 된다. 또 Hadoop은 가상머신 이미지 또는 데이터 이미지 파일을 전송 및 백업을 한다.

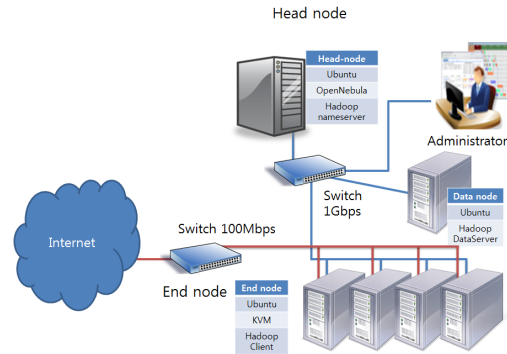


그림 7. 구현한 클라우드 플랫폼 구성도

나. 클라우드 환경에서 LVS구현

LVS는 1개의 Linux Director 가상 머신과 29대의 Real Server로 구현하였다. 가상머신의 운영환경은 표 1과 같다. 로드밸런싱 구조는 구축한 클라우드 컴퓨팅 네트워크 환경이 사설망인 NAT를 사용하기 때문에 그림 5와 같은 NAT 방식의 로드 밸런싱을 채택하였다. 로드밸런싱 설정은 표 2와 같이 Director Server에 LVS 솔루션인 Ipvssadm을 설치하고 Director Server로 들어오는 패킷을 처리서버 또는 Real Server에 포워딩하기 위해 표

2에 두 번째 줄 파일을 수정하였고 LVS 스케줄러는 라운드-로빈으로 설정하였다.

또 표 2의 마지막 줄은 Real Server를 추가하는 작업으로 29대의 Real Server를 추가하였다. 29대의 Real Server에는 Tomcat을 설치하여 웹서비스를 구축하였다.

표 1. 가상머신 운영 환경

OS	Ubutu 10.04LTS
Memory	1024

표 2. Linux Director 구성 환경

```
# apt-get install ipvsadm
# vi /etc/sysctl.conf
// net.ipv4.ip_forward = 1 변경
# sysctl -p
# ipvsadm -A -t 192.168.1.xxx:80 -s rr
# ipvsadm -a -t 192.168.1.xxx:80 -r 192.168.1.yyy:80 -m
```

#### IV. 실험 결과

본 실험에서는 일반 웹호스팅에서 LVS 환경을 구축하여 웹서비스를 로드 밸런싱하는 시간과 클라우드 컴퓨팅에서 로드 밸런싱하는 시간을 비교하였다. 실험 환경은 표 3과 같이 메모리는 1024MB를 할당하고 운영체제는 없는 초기화 상태에서 LVS를 구축하는데 걸리는 시간을 비교하였다.

표 4은 초기화 상태 일반 PC와 VM을 설치하고 LVS를 구현하는데 걸린 시간을 측정하여 비교한 것이다.

표 3. 실험 환경

	PC	VM
Memory	1024 MB	1024MB
OS	null	null

표 4. LVS 구현 시간 비교

환경 회수	PC	VM
1	1h	1h
2	2h	1h 10min
3	3h	1h 15min
4	4h	1h 20min
5	5h	1h 25min
...	...	...

#### V. 결 론

본 논문에서는 오픈 소스 기반의 클라우드 컴퓨팅 환경에서의 LVS 로드 밸런싱과 일반 웹 호스팅 환경에서의 LVS 로드 밸런싱 시간을 비교 분석하였다. 처음 1대의 노드를 설치하여 LVS를 구현 할 시에는 시간적 차이가 별로 발생하지 않았지만 2대 이상의 노드를 설정할 때에는 상당한 시간 차이를 보인다. 즉, 일반 PC에서는 첫 번째 노드와 같이 모든 노드를 처음부터 설치해야 하기 때문에 시간이 많이 걸리지만 클라우드 환경에서는 가상머신을 생성할 때 가상머신 이미지 파일을 복사한 뒤 배포하기 때문에 새로 운영체제 및 기타 설정을 해줄 필요가 없게 되어 LVS 구축시간을 크게 단축시킬 수 있다.

#### Acknowledgement

본 연구는 교육과학기술부 및 한국연구재단의 지역혁신인력양성사업의 일환으로 수행된 연구결과임.

#### 참고문헌

- [1] 송길현, "그린 데이터센터 구축 사례에 기반한 그린IT 도입 방안에 관한 연구", Information Systems Review, Vol.11, No.2, 2009.8.
- [2] 이주영, "클라우드 컴퓨팅 특징 및 사업자별 제공 서비스 현황", 방송통신정책, Vol.22, No.6, pp.1~22, 2010.
- [3] <http://www.opennebula.org>
- [4] <http://www.linux-kvm.org>
- [5] <http://hadoop.apache.org>
- [6] <http://www.linuxvirtualserver.org>