

## 가상 컴퓨팅 랩 환경에서 노드 전원관리 스케줄러 설계 및 구현

서경석<sup>1</sup> · 이봉환<sup>2\*</sup>

### Design and Implementation of a Node Power Scheduler in Virtual Computing Lab Environment

Kyung-Seok Seo<sup>1</sup> · Bong-Hwan Lee<sup>2\*</sup>

<sup>1</sup> Igloo Security Inc., Seoul, Korea

<sup>2</sup> Department of Information and Communications Engineering, Daejeon University, Daejeon 300-716, Korea

#### 요 약

기존의 PC 기반의 데스크탑 환경은 보안, 이동성, 업그레이드 비용 감소 등의 장점으로 인하여 서버 기반의 가상 데스크탑 환경으로 바뀌고 있다. 본 논문에서는 오픈소스 기반의 클라우드 컴퓨팅 플랫폼과 하이퍼바이저를 이용하여 컴퓨터 실습실에 적용 가능한 가상 컴퓨팅 랩 서비스 시스템을 설계하고 구현하였다. 또한, 서버 팜에 있는 노드들의 전력 소모를 줄이기 위한 전원 스케줄러를 제안하였으며, 이 스케줄러 탑재 시 기존시스템에 비하여 전력 소모량을 대폭 줄일 수 있는 실험 결과를 제시하였다.

#### ABSTRACT

The existing PC-based desktop environment is being changed to a server-based virtual desktop environment due to various advantages such as security, mobility, and upgrade cost reduction. In this paper, a virtual computing lab service system which is applicable to the existing computer lab is designed and implemented using both an open source-based cloud computing platform and hypervisor. In addition, a node power scheduler is proposed in order to reduce power consumption in a server farm. The experimental results show that the power scheduler reduces power consumption considerably over the server farm without the power scheduler.

**키워드** : 클라우드 컴퓨팅, 가상 컴퓨팅 랩, 하이퍼바이저, 전원 스케줄러

**Key word** : Cloud Computing, Virtual Computing Lab, Hypervisor, Power Scheduler

접수일자 : 2013. 01. 21 심사완료일자 : 2013. 07. 28 게재확정일자 : 2013. 08. 05

\* **Corresponding Author** Bong-Hwan Lee(E-mail:blee@dju.kr, Tel:+82-42-280-2553)

Department of Information and Communications Engineering, Daejeon University, Daejeon 300-716, Korea

**Open Access** <http://dx.doi.org/10.6109/jkiice.2013.17.8.1827>

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

클라우드 컴퓨팅[1]은 2006년 구글의 크리스토프 비시글리아라는 직원에 의해 구글 CEO 에릭 슈미츠와의 회의에서 제안된 것으로 인터넷을 통해 사용자가 소프트웨어, 서버, 스토리지 등의 IT 자원을 필요한 만큼 빌려서 혹은 구매해서 사용하고 비용을 지불하는 서비스 모델을 말한다. 클라우드 컴퓨팅은 구현을 하기 위한 하드웨어 인프라스트럭처의 모델에 따라 세 가지로 분류된다[2][3]. 첫 번째는 Public Cloud로 공용의 클라우드 컴퓨팅을 이용하여 클라우드 서비스를 구축하는 것으로 초기 구축비용이 적고 접근하기 쉽지만 서비스 업체의 클라우드가 장애를 일으키면 사용자는 이에 대한 대비를 할 수 없다는 문제점이 있다. 그러나 Public Cloud의 경우 사용자의 모든 데이터가 외부 클라우드에 저장되므로 데이터 보안상의 문제를 야기할 수 있다. 두 번째는 Private Cloud로 기업 또는 서비스를 구축하려는 기관에서 클라우드 컴퓨팅 장비를 직접 구축하여 사용하는 것으로 사용자 데이터 및 기관의 데이터가 구축한 기관에서 관리하기 때문에 보안이 Public Cloud 보다 안전하다. 하지만 초기비용이 많이 든다는 단점이 있다.

마지막으로 Hybrid Cloud는 Public Cloud와 Private Cloud를 혼합하여 구축한 클라우드이다. 이 경우 중요한 데이터는 Private Cloud에 저장하게 되고 기타 서비스를 위한 컴퓨팅 자원은 Public Cloud에서 이용하기 때문에 데이터의 안정성이 높고 Private Cloud 보다 적은 비용으로 클라우드 컴퓨팅을 구축하여 서비스를 제공할 수 있다.

본 논문에서는 오픈소스 기반 클라우드 컴퓨팅을 구축하기 위해 OpenNebula와 KVM(Kernal-based Virtual Machine), Hadoop의 HDFS(Hadoop Distributed File System)를 이용하여 가상 컴퓨팅 랩 시스템을 구현하고 가상 노드의 전원관리 스케줄러를 제안하여 그 성능을 비교 분석하였다.

## II. 관련 연구

클라우드 컴퓨팅 IaaS 구조는 그림 1과 같이 가상화를 이용한 클라우드에서 서비스를 관리하는 Head Node

와 가상화하여 자원을 나눠주는 End Node로 클러스터링 되어 있으며, 컴퓨팅 파워를 확장할 경우에는 End Node를 추가하면 된다. Head Node에는 End Node를 클러스터링 시키기 위한 클라우드 플랫폼 솔루션이 설치되어 있다. End Node에는 하이퍼바이저를 이용하여 가상머신을 생성하고 이를 통해 자원을 임대한다.

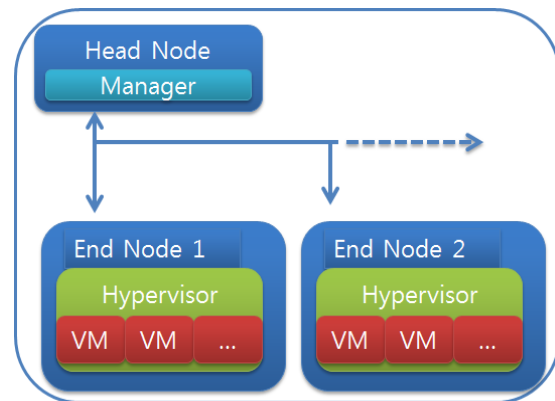


그림 1. 클라우드 컴퓨팅 구성도  
Fig. 1 Cloud Computing Architecture

OpenNebula[4][5]는 Universidad Complutense de Madrid의 DSA research Group에서 처음 개발한 공개 소프트웨어이며, 기관별로 하드웨어를 직접 구축하는 Private 클라우드를 구축할 수 있고 외부 클라우드 서비스 즉, Amazon의 AWS 등과 같은 인프라스트럭처를 임대하여 Public 클라우드를 구축할 수 있다. OpenNebula는 그림 2와 같이 여러 컴포넌트로 구성되어 있다. Interfaces 모듈에서는 OpenNebula 컴포넌트를 제어할 수 있는 Command Line Interface (CLI)와 클라우드를 웹에서 제어할 수 있는 Sunstone GUI를 제공한다. 스케줄러는 End Node에 가상머신을 할당하기 위한 최적 노드를 선택하여 가상머신을 생성한다. OpenNebula API는 OpenNebula의 모든 제어 즉, 가상머신 생성, 가상머신 이미지 관리, 가상네트워크 관리, 사용자 관리, End Node 관리 등의 모든 기능을 위해 Java와 같은 프로그래밍 API를 제공한다. 자원관리자는 가상머신, 이미지, 가상네트워크 등을 관리하며 자원 모니터링을 한다. Drivers는 KVM, Xen, VMware 등 하이퍼바이저에 대한 템플릿을 파싱해주어 다양한 하이퍼바이저를 사용할 수 있게 한다.

Compute는 End Node에 대한 관리와 모니터링에 대한 커스터마이징을 가능하게 한다. Network는 가상네트워크 자원들을 설정하고 Storage는 파일 관리시스템을 설정하며 Shared File System과 Non-Shared File System 등이 지원된다. Shared File System은 NFS와 같은 공유파일 시스템을 이용한 파일 관리이고 Non-Shared File System은 End Node의 로컬 파일시스템을 이용한 파일 관리이다. Cloud는 OCCL, AWS 등의 외부 Public 클라우드를 클러스터링 해준다.

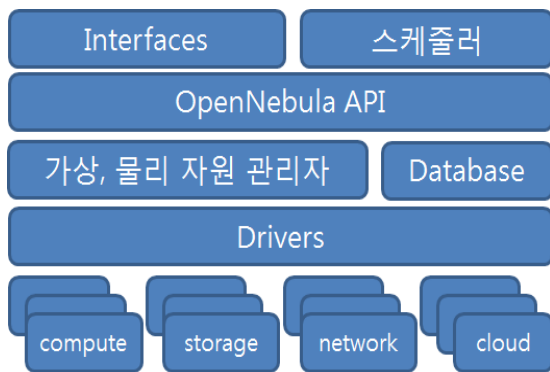


그림 2. 간략한 OpenNebula 구조  
Fig. 2 Simple Architecture of the OpenNebula

하이퍼바이저[4]는 서버 가상화를 이용하여 시스템에서 OS를 포함한 가상화 계층을 나타내며 VMM (Virtual Machine Monitor)으로 불리기도 한다. 하이퍼바이저는 다양한 OS를 실행 가능하게 해주는 논리적 플랫폼을 말한다.

가상화는 전가상화(Full Virtualization)와 반가상화(Para Virtualization)로 분류되며 전가상화는 이진코드 변환 기법(binary code translation)을 사용하는 VMM 위에서 서로 다른 Guest OS를 갖는 다수의 가상머신을 실행하는 구조로 Guest OS를 수정할 필요가 없지만 하드웨어 에뮬레이션으로 인한 성능 저하가 단점이다. 반가상화는 VMM 위에 서로 다른 Guest OS를 갖는 다수의 가상머신 구조이며 전가상화의 하드웨어 에뮬레이션으로 인한 성능 저하의 단점을 보완하였으나 하드웨어 API를 직접 Guest OS에 반영하기 위해 Guest OS를 수정해야 한다.

HDFS (Hadoop Distributed File System)[6][7][8]은 분산 컴퓨팅을 위한 오픈 소스 기반의 Nutch(lucene)의 서브프로젝트인 Hadoop에서 분산 컴퓨팅 프레임워크를 지원하기 위해 개발된 분산 파일시스템이다. 원래는 Google Search Engine Platform을 범용적인 플랫폼으로 개발하기 위해 시작되었으며, 그림 3과 같이 Nutch는 구글 검색엔진 구조에 대응된다. 현재 Amazon, IBM, Yahoo 등과 같은 글로벌 IT 벤더들의 클라우드 컴퓨팅 플랫폼의 기반이 되는 분산 파일 시스템으로 가장 널리 활용되고 있다. HDFS는 GFS (Google File System)를 본보기로 삼아 개발된 분산 파일시스템으로 플랫폼 간의 이식성을 보장하기 위해 자바를 사용하여 구현되었으며, HDFS를 이용한 개발에 쉬운 접근을 위해 JAVA 용 라이브러리가 제공된다.

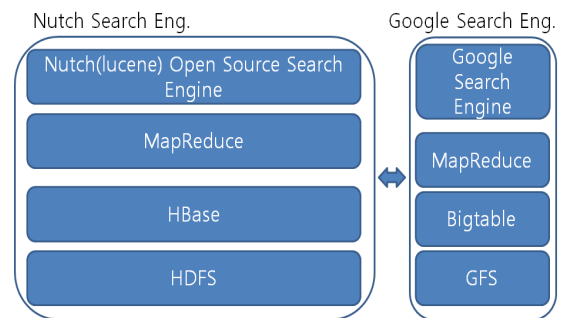


그림 3. Nutch 검색엔진과 구글 검색엔진 구조 비교  
Fig. 3 Comparison of the Nutch and Google Search Engine

HDFS를 이용한 하나의 분산 파일시스템을 구축하기 위해서는 그림 4와 같은 하나의 Name Node와 블록 파일이 저장되는 다수의 Data Node가 필요하다. Name Node에서는 파일시스템의 네임스페이스(Directory, Filename, Location 등)를 관리하면서 사용자의 파일 접근 요청을 처리한다. HDFS에서는 파일들은 기본적으로 64MB의 블록 단위로 저장된다. 데이터는 Data Node에 저장되고 Data Node의 장애 상황을 대비하기 위해 복사본을 생성하여 기본적으로 총 3개의 같은 데이터 블록이 생기게 되며, 복사본들은 부하 분산을 위해 여러 Data Node에 나누어 저장된다.

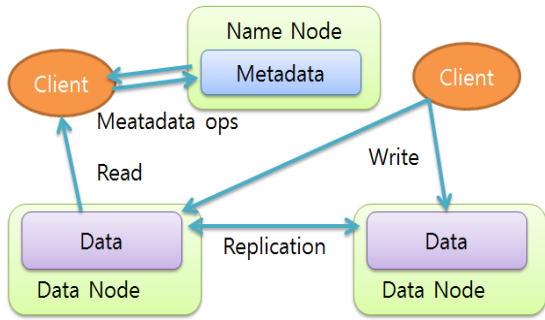


그림 4. 하둡 분산 파일시스템 구조  
 Fig. 4 Architecture of the Hadoop Distributed File System

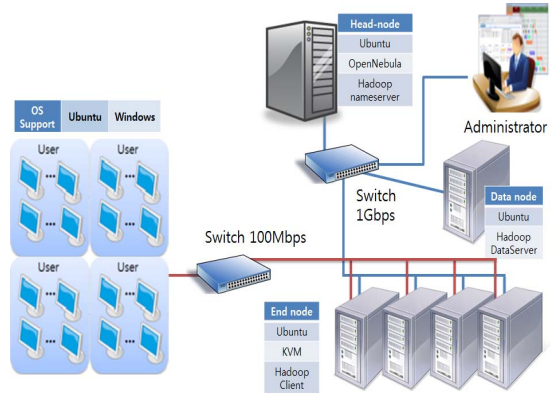


그림 5. 가상 컴퓨팅 랩 구조  
 Fig. 5 Architecture of the Virtual Computing Lab.

### III. 가상 컴퓨팅 랩 서비스 시스템 설계 및 구현

가상 컴퓨팅 랩 서비스[9][10][11]는 서버가 가상화를 이용하여 가상머신을 생성하여 사용자에게 가상의 데스크 탑 서비스를 제공하여 사용자에게 일정한 품질의 데스크탑을 제공하면서 관리자에게는 하나의 가상머신 이미지를 이용하여 배포하기 때문에 유지 보수 시간과 비용을 줄여준다.

가상 컴퓨팅 랩 서비스를 이용하여 강의에 대한 시간표 작성과 가상의 강의에 필요한 가상머신을 제공하는 VDI (Virtual Desktop Interface) 서비스를 위해 그림 5와 같은 구조로 설계하였다. Head Node는 End Node와 Data Node를 제어하기 위한 관리자 노드로 분산파일 시스템을 제어하여 HDFS의 Name Node 역할을 하며 분산된 파일 블럭들의 메타 데이터들을 저장하고 클라이언트 요청에 대해 응답을 해준다. End Node는 하이퍼바이저와 HDFS를 접속하기 위한 클라이언트를 내장하여 가상머신 생성과 삭제 등의 관리를 제어하는 노드이다. Data Node는 HDFS의 블럭단위 파일이 저장되며 클라이언트 요청에 대한 응답을 한다.

그림 5와 같은 클라우드 컴퓨팅 랩을 설계한 후 VDI 서비스를 제공하기 위해 기존 컴퓨터 실습실의 구성요소(표 1)를 분석하여 그림 6과 같은 서비스를 구현하였다.

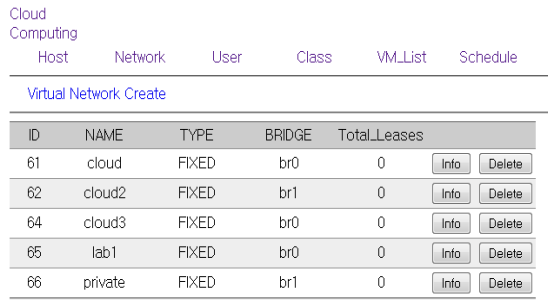
그림 6의 웹서비스를 이용하여 관리자가 실습실에 필요한 정보와 운영 일정을 입력하면 자동적으로 서비스가 제공된다. 클라우드 구축 노드 사양은 표 2와 같고 1대의 Head Node, 4대의 End Node 및 2대의 Data Node를 이용하면 4대의 End Node에서는 총 30대의 가상머신이 생성되며 30명이 수강하는 실습 과목을 운영할 수 있다. Head Node, End Node 및 Data Node에 대한 하드웨어 사양은 표 2와 같다.

표 1. 기존 실습실 구성 요소  
 Table. 1 Existing Lab Components

구성요소	용도
실습실	실습강의 공간
강의 시간표	실습강의 시간표
강의 정보	실습강의에 대한 특성 및 종류
운영체제	실습강의에 사용되는 운영체제
S/W	실습강의에 사용되는 응용프로그램
사용자 데이터	실습에서 발생하는 사용자 데이터
사용자 단말	실습강의용 단말
네트워크 정보	가상네트워크 이름 및 가상머신에 할당되는 IP 주소
수강생 정보	실습학생 정보

표 2. 전체 노드 사양  
Table. 2 Specifications of Cluster Nodes

	CPU	RAM	HDD	OS
Head Node	Quad Core 2.80GHz	16GB	1TB	ubuntu 12.04.1
End Node	Quad Core 2.80GHz	16GB	1TB	ubuntu 12.04
Data Node	Quad Core 3.10GHz	4GB	4TB	ubuntu 12.04



Daejeon University. Computer Networks & Security Lab

그림 6. 가상 컴퓨팅 랩 서비스 페이지  
Fig. 6 Screenshot of Virtual Computing Lab

#### IV. 가상 컴퓨팅 랩 환경에서 노드 전원관리 스케줄러 설계 및 구현

가상 컴퓨팅 랩 환경에서 서버팜에 있는 노드들의 전력을 절감하기 위해서는 시스템을 사용하지 않을 때에는 강제로 노드를 종료시키는 방법이 효과적이다. 본 논문에서 제안하는 노드 전원 관리 스케줄러는 Head Node, End Node 및 Data Node로 구성된 클라우드 서비스 환경에서 모든 노드를 관리하지 않고 미션크리티컬한 Head Node와 Data Node를 제외한 End Node들의 전원을 관리하여 에너지 비용을 줄이기 위한 것이다. 가상 컴퓨팅 랩 환경에서 End Node의 전원관리를 위해

Head Node에 노드 전원관리 스케줄러를 탑재한다. 스케줄러의 동작 흐름도는 그림 7과 같다.

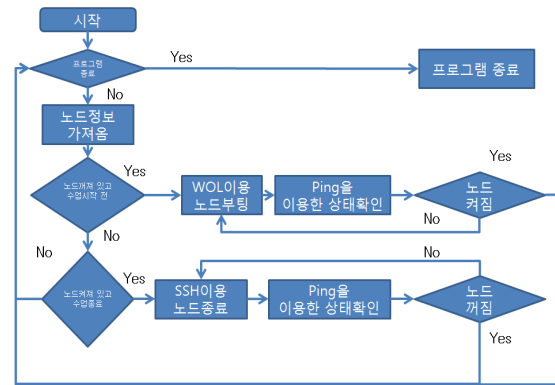


그림 7. 전원관리 스케줄러 흐름도  
Fig. 7 Flow chart of Power Scheduler

스케줄러는 노드의 원격 부팅 기능과 노드 전원을 원격으로 종료시키는 기능 등 두 가지 기능을 가지고 있고 이 두 가지 기능은 DB 검색, SSH[12], WOL[13][14], Ping 등 네 가지 모듈로 이용하여 구현하였다. 첫 번째 기능은 스케줄러를 통해 End Node 컴퓨터를 부팅시키는 기능이다.

컴퓨터를 부팅시키기 위해서는 먼저 수업 관련 정보와 노드의 상태 정보를 이용해서 부팅 여부를 결정하게 된다. 스케줄러는 DB에서 강의 시간표를 불러와서 WOL 기술을 이용하여 모든 End Node들에게 Magic 패킷을 전송하여 부팅시키게 된다. 부팅된 End Node들의 상태를 알아보기 위해 Ping을 전송하여 End Node의 부팅 상태를 파악하게 되는데 Ping이 정상적으로 정보를 가져오면 End Node는 정상 부팅이 되는 것이고 그렇지 않으면 부팅에 실패한 것이기 때문에 다시 WOL 작업을 반복한다.

ID	NAME	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	AMEM	STAT
9	192.168.1.10	0	800	800	800	15.7G	15.4G	15.7G	on
10	192.168.1.5	0	800	800	800	15.7G	15.4G	15.7G	on
11	192.168.1.8	0	800	800	800	15.7G	15.4G	15.7G	on
12	192.168.1.9	0	800	800	800	15.7G	15.3G	15.7G	on

그림 8. End Node가 사용 가능한 상태  
Fig. 8 Enabled End Node State

모든 End Node들이 정상 가동되면 OpenNebula API 를 이용하여 그림 8처럼 End Node를 사용 가능 상태로 변경하게 된다.

두 번째 기능은 스케줄러를 통해 End Node들을 원격으로 종료시켜 전력을 감소시키는 기능이다. 스케줄러는 DB에서 강의시간 테이블 가운데 해당 요일의 마지막 수업 종료시간을 받아와서 현재 시간과 비교하여 같으면 End Node를 종료시키기 위해서 SSH로 원격 명령을 내린다. SSH를 이용하여 종료 명령을 내리고 난 뒤에는 Ping을 이용하여 End Node들의 상태를 확인하고 OpenNebula API를 이용하여 모든 End Node들이 정상 가동되면 OpenNebula API를 이용하여 End Node를 사용 불가능 상태로 변경한다.

### V. 실험 및 성능 분석

제안한 스케줄러의 성능 분석을 위한 시나리오를 제시하고 그 결과를 분석함으로써 제안한 시스템의 유용성을 평가하였다. 이를 위해 전력 스케줄러를 사용하는 경우와 사용하지 않는 경우에 동일한 가상 컴퓨팅 랩 서비스 환경과 동일한 수강 시간표를 이용해서 사용 전력을 측정하였다. 실험을 위해서 그림 9와 같은 가상의 강의 시간표를 사용하였다.

	MON	TUE	WED	THU	FRI
9					
10	ubuntu(1) Info				windows(1) Info
11			windows(1) Info		
12		windows(1) Info		ubuntu(1) Info	
13					
14	windows(1) Info				ubuntu(1) Info
15			ubuntu(1) Info		
16		ubuntu(1) Info		windows(1) Info	
17					

그림 9. 실험을 위한 강의 시간표  
Fig. 9 Time Table for Performance Evaluation

노드 전원관리 스케줄러를 미적용한 시스템 실험은 그림 9의 강의시간표에 따라 1주일간 매일 24시에 사용 전력을 측정하고 3회 반복한 측정결과를 표 3에 나타내었다. 표 3에서 나타나듯이 각 요일의 전력 사용량은 큰 변화가 없었다. 한편, 노드 전원관리 스케줄러를 적용한 시스템 실험에서도 그림 9의 동일한 강의시간표에 따라 1주일간 매일 24시에 사용전력을 측정하고 3회 반복한 측정결과를 표 4에 나타내었다.

표 3. 전원 스케줄러 미사용 시 사용전력량  
Table. 3 Power Consumption w/o Power Scheduler

요일 \ 횟수	1회	2회	3회
월요일	10,848 W	10,838 W	10,872 W
화요일	10,834 W	10,821 W	10,813 W
수요일	10,851 W	10,837 W	10,841 W
목요일	10,844 W	10,851 W	10,828 W
금요일	10,829 W	10,810 W	10,831 W
토요일	10,608 W	10,612 W	10,609 W
일요일	10,597 W	10,601 W	10,618 W

전원 스케줄러를 가동시킨 경우 주중에도 전력이 적게 소모되지만 강의가 없는 토요일, 일요일과 같은 주말에는 더 많은 소비 전력이 줄게 됨을 알 수 있다.

표 4. 전원 스케줄러 사용 시 사용전력량  
Table. 4 Power Consumption w/ Power Scheduler

요일 \ 횟수	1회	2회	3회
월요일	7,056 W	7,049 W	7,063 W
화요일	7,061 W	7,032 W	7,041 W
수요일	7,049 W	7,012 W	7,039 W
목요일	7,057 W	7,032 W	7,051 W
금요일	7,080 W	7,039 W	7,048 W
토요일	4,920 W	4,941 W	4,938 W
일요일	4,919 W	4,918 W	4,931 W

전원 스케줄러를 사용하지 않은 경우와 사용한 경우 요일별 평균 전력 소모량을 바탕으로 일주일간 누적 전력 사용량을 그래프로 나타내면 그림 10과 같다.

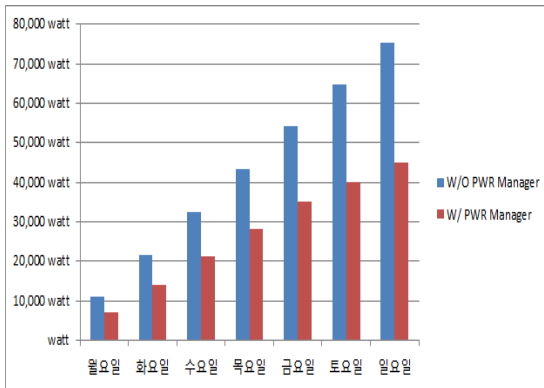


그림 10. 일주일간 누적 사용 전력  
Fig. 10 Cumulative Power Consumption per Week

그림 10의 누적 평균 사용전력 그래프를 보면 가상 컴퓨팅 랩 환경에서 노드 전원관리 스케줄러를 이용할 수록 전력 소비가 큰 폭으로 차이나는 것을 실험으로 통해 확인하였다. 1주일간 평균 사용전력의 누적 값을 보면 노드 전원관리 스케줄러를 미적용한 시스템은 1주일간 평균 75,395W를 사용하게 되고 전원관리 스케줄러를 적용한 시스템은 1주일간 평균 45,089W를 사용하여 1주일간 평균 사용전력 차이는 30,306W이다. 즉, 노드 전원관리 스케줄러를 사용하였을 때 소비전력이 40% 정도 절감할 수 있음을 실험적으로 보였다.

## VI. 결 론

PC 기반의 데스크탑 환경은 보안, 이동성, 업그레이드 비용 감소 등의 장점으로 인하여 서버 기반의 VDI 환경으로 바뀌고 있다. 본 논문에서는 오픈소스 기반의 클라우드 컴퓨팅 플랫폼과 하이퍼바이저를 이용하여 기존의 컴퓨터 실습실을 대체할 수 있는 가상 컴퓨팅 랩 서비스 시스템을 설계하고 구현하였다. 또한, 서버 팜에 있는 노드들의 전력 소모를 줄이기 위한 전력 관리 스케줄러를 제안하였으며, 이 스케줄러 탑재 시 전력 소모량을 비교 분석하였다.

30개의 가상머신을 생성하는 4대의 End Node에 제안한 전력 관리 스케줄러를 적용한 경우 일주일에 사용하는 전력은 기존 시스템에 비해 40% 가량 절감됨을 실험을 통해 확인하였다.

## REFERENCES

- [ 1 ] Mladen A. Vouk, "Cloud Computing - Issues, Research and Implementations", Journal of Computing and Information Technology - CIT 16, pp. 235-246, 2008. 4.
- [ 2 ] NIST, "NIST Definition of Cloud Computing", <http://www.nist.gov/itl/cloud/>, 2011. 2.
- [ 3 ] Jung Seung-Ho, Shin Shin-Ae, Kwon Young-il, Park Ji-hye, "A Study on Cloud Service in e-Government Division for Green IT", The Korea Society of Management Information Systems, pp.684-690, 2009.
- [ 4 ] O. G. Mim, H. Y. Kim, and G. H. Nam, "Trends in Technology of Cloud Computing", Electronics and Telecommunications Trends, 24(4), ETRI pp. 1-13, 2009. 8.
- [ 5 ] OpenNebula, <http://www.opennebula.org>, 2012. 11.
- [ 6 ] Dhruba Borthakur, "The Hadoop Distributed File System: Architecture and Design", Technical Report, <http://Hadoop.apache.org/core>, pp. 1-14, 2007.
- [ 7 ] Tom White, "Hadoop: The Definitive Guide", O'REILLY, pp. 1-528, 2009. 7.
- [ 8 ] Hadoop, "What is Apache Hadoop", <http://hadoop.apache.org/index.html>
- [ 9 ] Marianne C. Murphy, Marty McClland, "Computer Lab to Go: A "Cloud" Computing Implementation", Proc. ISECON 2008, v25.
- [ 10 ] Epelbaum, S. S, "Virtual Computing Lab Experiments in Telecommunications for Distance Learning", In Proc. of the Information Systems Education Conference, v.17, Philadelphia, 2000.
- [ 11 ] Vouk, M. "Cloud Computing", In Proc. of the Information Systems Education Conference, v.17, Pittsburgh, 2007.
- [ 12 ] Secure Shell, "SSH Intro", [https://wiki.archlinux.org/index.php/Secure\\_Shell/](https://wiki.archlinux.org/index.php/Secure_Shell/), 2012. 11.
- [ 13 ] Wake On Lan, "Wake On Lan intro", <https://wiki.archlinux.org/index.php/Wake-on-LAN/>, 2012. 11.
- [ 14 ] Wake On Lan, "IBM Announces Universal Management - Industry's Most Comprehensive Tools to Lower Total Cost of Ownership", [http://www-03.ibm.com/press/us/en/press\\_release/2705.wss](http://www-03.ibm.com/press/us/en/press_release/2705.wss), 2012. 11.



**서경석(Kyung-Seok Seo)**

2011년 대전대학교 컴퓨터 공학과 졸업(학사)  
2013년 대전대학교 정보통신공학과 졸업(석사)  
현재 ㈜이글루시큐리티 연구원  
※관심분야 : 클라우드 컴퓨팅, 빅데이터, 네트워크 등



**이봉환(Bong-Hwan Lee)**

1985년 서강대학교 전자공학과 졸업(학사)  
1987년 연세대학교 대학원 전자공학과 졸업(석사)  
1993년 Texas A&M 대학교 대학원 전기 및 컴퓨터공학과 졸업(박사)  
현재 대전대학교 정보통신공학과 교수  
※관심분야 : 클라우드컴퓨팅, 빅데이터, 네트워크보안 등