

성능분석을 통한 안정화된 클라우드 컴퓨팅 기반 교육 실습 시스템 구축

윤준원*, 송의성**

요약

이미 잘 알려져 있는 클라우드 컴퓨팅은 사용자가 요구하는 컴퓨팅 자원을 최적화하여 유연하고 확장성 있게 지원할 수 있어 다양한 분산컴퓨팅 분야에 적용되고 있다. 클라우드 컴퓨팅은 가상화 기술을 통해 실제적인 구현 및 서비스가 가능하며 여러 가상화 기술과 제품들이 개발, 출시되고 있다.

본 논문에서는 클라우드 컴퓨팅의 가상화 기반으로 교육 실습 환경을 구축하고, 각 실습환경의 성능을 분석하였다. 클라우드 컴퓨팅 기반의 가상화 실습 환경은 최적화된 자원을 제공할 수 있으며, 실습 제공자는 자원 관리의 편리성, 결과에 대한 손쉬운 관리, 실습환경의 성능 및 상태 등의 수월하게 관리하고 파악 할 수 있다. 또한, 다양한 실습환경을 구축하는데 있어 요구되는 시스템 환경을 유연성 있고 신속하게 제공할 수 있어 시스템의 활용성 또한 높아지게 된다. 나아가 가상화 실습환경의 성능 분석을 통하여 구동될 수 있는 교육 어플리케이션의 성능을 미리 평가하고 가늠할 수 있다.

키워드 : 클라우드 컴퓨팅, 가상화, 성능분석, 교육실습

Build the Teaching Practice System based on Cloud Computing for Stabilization through Performance Evaluation

JunWeon Yoon*, Ui-Sung Song**

Abstract

Cloud computing is already well known paradigm that a support computing resource flexible and scalable to users as the want in distributed computing environment. Actually, cloud computing can be implemented and provided by virtualization technology. Also, various products are released or under development.

In this paper, we built the teaching practice system using cloud computing and evaluated practical environment which constructed over a virtual machine. Virtualization-based cloud computing provides optimized computing resources, as well as easy to manage practical resource and result. Therefore, we can save the time for configuration of practice environment. In the view of faculty, they can easily handle the practice result. Also, those practice condition reuse comfortably and apply to various configuration simply. And then we can increase capabilities and availabilities of limited resources. Additionally, we measure the performance requirements for educational applications through evaluation of virtual-based teaching practical system in advance.

Keywords : Cloud Computing, Virtualization, Performance Evaluation, Practical Education

1. 서론

※ 교신저자(Corresponding Author): Ui-Sung Song
접수일:2014년 09월 13일, 수정일:2014년 10월 21일
완료일:2014년 10월 23일

* KISTI 국가슈퍼컴퓨팅연구소 선임연구원
Tel: +82-42-869-0581, Fax: +82-42-869-0569
email: jwyoong@kisti.re.kr

** 부산교육대학교 컴퓨터교육과
ussong@bnue.ac.kr

클라우드 컴퓨팅은 존재하는 자원들을 요구사항과 목적에 맞게 최적화하여 사용자에게 제공할 수 있는 장점을 가지고 있어 최근 학계, 산업

이 논문은 2014년도 부산교육대학교 교육연구원의 지원을 받아 연구되었음

분야 등의 다양한 솔루션을 제공하기 두루 적용되고 있다. 클라우드 컴퓨팅은 가상화라는 기술을 기반으로 실제적인 구현 및 서비스가 가능하며 사용자가 요구하는 컴퓨팅 환경을 유연하게 구성할 수 있다[1]. 최근 x86 서버에 멀티코어가 일반화되고 집적도가 증가하면서 성능은 점차 증가하였고, 더불어 가상화를 제공하기 위한 오픈소스 또는 상용 프로그램들의 다양하게 출시 개발되고 있다[2]. 더욱이 안정성 및 성능에 대한 효용성이 검증됨에 따라 클라우드 컴퓨팅 활용이 점차 두각을 나타내고 있다. 클라우드 컴퓨팅의 대표적인 장점은 분산컴퓨팅 환경에서 사용자가 요구하는 자원을 사용자의 요구사항에 맞게 최적화하여 유연하고 확장성 있게 지원할 수 있다는 점이다[3].

본 논문에서는 이런 특징을 활용하여 대학 내에서 수행되는 실습환경을 클라우드 컴퓨팅의 가상화 기술을 이용하여 구성함으로써 학생들이 수행해야 할 시스템 환경을 최적화하여 제공할 수 있다. 나아가 구축된 가상화 기반 실습환경(Virtual Machine)의 성능을 분석하여 제공함으로써 사용자가 실습 환경을 이용할 시 요구되는 자원의 요구 성능을 보장하고 안정적인 환경을 제공할 수 있다. 실습 제공자는 자원 관리의 편리성, 결과에 대한 손쉬운 관리, 실습환경의 성능 및 상태 등의 수월하게 관리하고 파악 할 수 있는 장점을 가진다. 또한 다양한 실습환경을 구축하는데 있어 요구되는 시스템 환경을 유연성 있고 신속하게 제공할 수 있어 시스템의 활용성 또한 높아지게 된다. 나아가 가상화 실습환경의 성능 분석을 통해 구동될 수 있는 교육 어플리케이션의 성능을 평가하고 가능할 수 있다[4].

2. 관련 연구

본 논문에서는 클라우드 컴퓨팅 기반에서 가상화를 통해 최적화 된 실습 환경을 구축하기 위하여 오픈 소스 기반의 SW를 활용하여 구축하였다. 리눅스 기반 OS상에서 가상 머신들을 생성하고, 가상머신(VM: Virtual Machine)을 관리하는 오픈소스 가령, Eucalyptus, OpenNebula, Nimbus, OpenStack, Abiquo 등과 같은 IaaS (Infrastructure as a Services) 기반의 SW를 설

치하였다, 또한 가상머신 위에서 운영하게 될 다양한 실습환경을 자원의 요구사항(SW, 환경설정)에 따라 이미지로 만들어 복제 사용함으로써 시스템 설정 및 관리가 용이해진다. 이런 환경을 가상 어플라이언스(Virtual Appliance)고 하며 가상화 환경이 생성되면서 바로 사용자가 직접 사용할 수 있도록 OS 뿐만 아니라 소프트웨어 스택(Software Stack)이 구성된다. 이로써 다양한 실습환경에 맞게 가상 어플라이언스를 생성하여 제공함으로써 자원 활용성 및 재사용성을 최대화할 수 있다[5].

클라우드 컴퓨팅의 핵심 기술은 가상화 자원을 사용하여 문제를 해결하는 데 있으며. 이를 어떻게 구성하고 관리하는가가 가장 큰 이슈 중에 하나이다. 클라우드 컴퓨팅은 다수의 상이한 컴퓨팅 자원들을 가상화하여 사용자에게 온디맨드 방식으로 제공하는 기술로 자원 효율성의 극대화 와 관리비용 최소화라는 장점으로 인터넷 환경 최신 트렌드로 부상하고 있다. 다음은 클라우드의 핵심 기술을 확산시키는데 주도적인 역할을 수행했던 공개SW, 즉 오픈소스 기반 IaaS (Infrastructure as a Service) 솔루션 들이다.

▪ Nimbus

Nimbus는 오픈소스 클라우드 툴킷으로 사용자에게 IaaS 클라우드 서비스를 제공한다. 이 프로젝트는 시카고대-아르곤(UChicago Argonne LLC)이 운영하는 아르곤 국립연구소 Kath Keahy 연구팀에 의해서 개발되었으며, 과학자에게 가상머신을 사용하여 방대한 데이터를 처리하는 오픈 소스 클라우드 컴퓨팅 인프라이다. 또한 Nimbus는 Amazon의 EC2와 유사하게 사용자가 가상머신을 사용하도록 하는 클라우드 컴퓨팅 구조이고, 여러 개의 가상머신들을 가상 클러스터로 결합하여 사용할 수 있는 Nimbus Context Broker와 같은 사용자 레벨의 툴을 제공한다[6].

▪ Eucalyptus

Eucalyptus(Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems)는 리눅스 기반의 인프라스트럭처 상에서의 IaaS 클라우드 컴퓨팅 서비스를 제공하는 오픈소스 기반 소프트웨어 플랫폼으로 2007년 캘리포니아 대학의 Computing Science 학부에서 연구 프로젝트로 시작하였다. 2007년 VGrADS(Virtual Grid

Application Development Software) 프로젝트에서 LEAD(Linked Environment for Atmospheric Discovery) 기후 예측 시스템을 개발 중 LEAD에서 생성되는 대규모의 워크플로를 수행하기 위해 클라우드 환경의 필요성이 제기되었다. 이에 아마존 AWS가 고려되었으며 VGrADS와 AWS 연동을 위한 로컬 클라우드 플랫폼으로 Eucalyptus가 개발되었다. Eucalyptus는 Hypervisor를 위해 KVM 및 Xen을 지원하며 클러스터 관리를 위한 Rocks 클러스터 배포판을 포함하고 있다[7][8].

▪ **OpenNebula**

OpenNebula는 Universidad Complutense de Madrid의 DSA-research Group에서 시작된 오픈 소스 기반의 IaaS Management 툴킷으로 물리적 자원들 안에서 가상머신을 동적으로 구성하고 재할당하는 오픈 소스 가상 인프라스트럭처 엔진이다. private, public, hybrid cloud를 구축할 수 있으며 Amazon AWS, OGC OCCI, VMware vCloud와 연동이 가능한 공통 인터페이스를 제공한다. Open Nebula는 어떤 네트워킹과 스토리지 솔루션과 통합될 수 있고, 그래서 현존하는 어떠한 데이터센터와 통합되도록 설계되었다. OpenNebula는 Xen, KVM/Linux 및 VMware를 지원하며 libvirt와 같은 요소를 사용하여 관리 및 검사 작업을 수행한다[9].

▪ **OpenStack**

OpenStack은 Public과 Private 클라우드를 구축하기 위한 오픈 소스 software로 서버, 스토리지, 네트워크, 가상화 기술들과 같은 리소스들을 모으고, 이들을 제어하고 운영하기 위한 Cloud Operating System이다. 또한, OpenStack은 오픈 소스 기반으로 클라우드를 구축하고 운용하고자 하는 오픈소스 개발자, 회사, 사용자들로 이루어진 커뮤니티이다. 오픈스택은 크게 3가지 컴포넌트로 구성된다.

- Swift: Amazon S3와 같은 object/blob 저장소.
- Glance: VM 이미지, 스토리지와 같은 클라우드 자원을 검색하기 위한 컴포넌트
- Nova: Amazon EC2와 비슷하다. 컴퓨팅자원을 할당해서 VM을 만들기 위한 컴포넌트

이들은 각각 독립된 프로젝트로 이들이 서로 협력해서 클라우드 인프라를 구축한다. 각 컴포넌트들은 매우 느슨하게 연결 되어 있어서, 확장

개발이 용이하다는 장점을 가진다[10].

▪ **Amazon AWS**

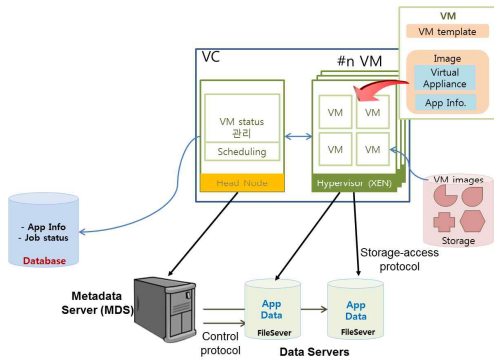
Amazon Web Services(AWS)는 아마존에서 제공하는 웹서비스를 통칭하며, 웹 서비스 인터페이스를 통해 사용자가 원하는 파워의 규모에 맞게 자원과 시스템 환경을 구성할 수 있다. AWS 서비스로는 EC2, EBS, S3, SQS, Mturk, Alexa, A2S, FPS, DevPay, SimpleDB, CloudFront 등이 다양한 솔루션을 제공하고 있으며 지속적으로 새로운 서비스들을 추가하고 있다. 아마존 AWS 플랫폼 구조는 Xen기반의 EC2 호스트, EC2 메니저, Amazon Simple Storage, S3 관리자로 구성되어 있으며, SOAP과 REST 기반의 AWS 인터페이스를 제공한다[11].

위와 같이 다양한 컴퓨팅 환경에서, 클라우드 컴퓨팅이 가져온 변화는 독립된 컴퓨팅 자원을 활용하기 보다는, 이미 존재하는 자원들을 상황과 목적에 맞게 최적화하여 이용할 수 있다는 것이다. 또한 일시적으로 특정 목적에 최적화된 계산 자원을 유연하게 설계하여 제공할 수 있어 다양한 실습 환경에 맞는 컴퓨팅 환경을 유동적으로 제공하여 지원할 수 있게 된다.

3. 가상화 기반 실습환경 구축

클라우드 기반의 교육 실습 환경을 구축하기 위해 실제 실습이 구동되는 가상머신을 관리할 필요성이 있다 이를 요구되는 도구로서 관련 연구에서 다루었던 오픈소스 기반의 IaaS 도구인 OpenNebula를 사용하여 가상머신들을 제어하였다. 전체적인 가상 자원의 생성, 초기화, 동작, 대기, 회수 등을 일괄적으로 수행할 수 있다. 또한 다양한 교육용 실습환경이 가상머신을 통해서 구현되기 위해서는 이미지 형태의 가상 어플라이언스들(Virtual Appliance)을 생성하여 저장하게 된다. 가상 어플라이언스는 시스템의 요구사항을 담은 일종의 컨테이너로 의존성이 있는 어플리케이션, 라이브러리, 서비스, 관련 데이터 그리고 운영체제까지 통합된 일종의 소프트웨어 스택이다. 실습환경이 구성되면 가상 환경으로 제공 되는 실습 데이터와 이를 가공한 산출물 데이터를 관리하기 위한 공용의 파일시스템이 필요하다.

(그림 1) 가상화 기반 실습 환경 구성도

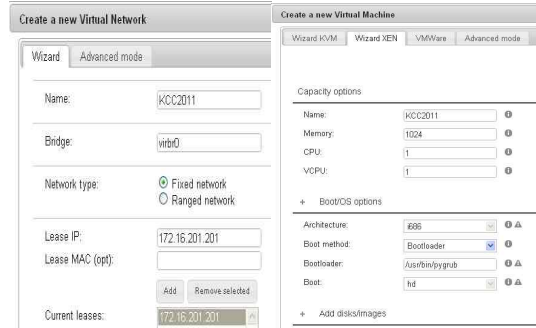


(Figure 1) Practice System based Virtualization

(그림 1)은 가상화 기반의 실습환경구성도를 보여주고 있다. 가상 어플라이언스를 구성하기 위한 하이퍼바이저 즉, 가상머신(VM)으로 Xen을 사용하였다. 대표적인 가상머신으로써는 Xen, KVM, VMware 등이 있다. Xen은 반가상화(Para Virtualization)라는 게스트 운영체제 일부를 수정하는 방식으로 동작하며 게스트 운영체제는 특권 명령을 실행하려면 '하이퍼콜(Hypercall)'로 하이퍼바이저에게 서비스를 요청하는 해야 한다. 운영체제에서 어플리케이션이 커널에게 시스템 콜(System Call)로 서비스를 요청하는 방식과 동일하다.

두 번째로 실습환경이 구동될 복수개의 가상환경(VM)을 관리하기 위해서는 IaaS기반의 오픈소스가 요구된다. 관련연구에서 논하였듯이 대표적인 오픈소스 IaaS 소프트웨어로서는 Nimbus, OpenNebula, OpenStack, Eucalyptus 등이 있으며, 본 실험에는 OpenNebula를 설치하여 VM을 관리하였다. 이 툴을 사용하여 VM들을 동작, 종료, 대기, 리셋 시킬 수 있으며 장애가 발생할 시 마이그레이션 기능도 제공된다. VM은 리눅스 기반(CentOS) 이미지로 생성하였으며 (Figure 2)와 같이 오픈소스 기반의 IaaS 도구인 OpenNebula를 이용하여 VM을 생성하였다.

(그림 2) OpenNebula 가상 네트워크 할당 및 VM 생성 화면



(Figure 2) Set up of virtual network and VM creation on OpenNebula

4. 실습환경 성능분석

4.1 성능 분석을 위한 벤치마크

구축된 가상화 기반의 실습 환경에서 실습 사용자의 어플리케이션이 정상적으로 제시시간에 이상 없이 수행되기 위해서는 안정적인 성능을 보장하여야 한다. 실습환경 관리를 위해서는 부품 교체, 수리 심지어 시스템 자체가 변경되었을 시에도 일정한 성능을 낼 수 있는지 검증을 해야 한다. 이를 위해 다양한 벤치마크(benchmark) 도구가 사용된다. 본 연구에서는 HPL (High-Performance Linpack Benchmark)을 이용하여 실습 시스템의 실제 성능을 측정하였다[12]. HPL은 분산-메모리 컴퓨터에서 각각의 시스템이 성능을 측정하는데 사용한다. HPL은 mpi를 통해서 Linpack(여러 프로세서가 하나의 커다란 메모리 공간에 접근하여 계산을 수행)을 병렬로 처리할 수 있도록 하는 기능 또한 제공한다. HPL에서 Rmax는 LINPACK 벤치마크를 통해 실제 측정된 최대 Flops 값이며, Rpeak는 시스템에 장착된 프로세서수, 클럭주파수, 성능 factor 값 등을 고려하여 이론적으로 산출된 최대 Flops 값이다. 벤치마크에서 실제 측정된 수치인 Rmax 값을 기준으로 결정된다. 보통 실제 성능 Rmax 값은 이론 성능인 Rpeak 값에 비해 낮게 측정되는데 OS와 계산노드가 사용하는 리소스 사용량이 존재하기 때문이다. 이런 성능차이는 CPU 특성이나 시스템 아키텍처 등의 영향을 받는다. 클라우드 컴퓨팅의 가상화 환경에서는 각각의 가상 머신들을 관리하기 위한 하이퍼바이저

(Hypervisor)의 리소스 또한 요구가 되어 자체 시스템(bare machine)의 성능보다 더 낮게 측정된다.

4.2 성능분석을 위한 실습환경 구축

본 실험을 위해 Intel core i7-920(2.66GHz) 프로세서에 12GB 메모리를 가지는 시스템(Bare Machine)에서 가상 머신 환경을 변경(코어 개수, 메모리 크기)하면서 HPL을 이용하여 성능을 측정하였다. 실제로 2009년 말부터 인텔에서는 CPU 아키텍처에 대한 이론치 최대성능(Theoretical maximum Gflops) 공식적으로 제공하지 않고 있다. 하지만 아래와 같은 공식을 이용하여 최대 이론치 성능을 측정할 수 있다.

- Theoretical Performance(Rpeak) = Number of cores x Floating point instructions per clock cycle x Clock frequency
- i7 920 @ 2.66GHz : 4 x 4 x 2.66 GHz = 42.56 GFlops (1 core =10.64GFlops)

Intel Core i7에서는 하이퍼스레딩(Hyper-Threading)라는 기술이 적용되었는데 프로세서 리소스를 보다 효율적으로 사용하기 위해 코어당 두 개의 스레드(thread)를 수행할 수 있도록 성능을 높여주는 기능이다. 이론치 성능 측정에서는 하이퍼스레딩 기술을 비활성화 하여 측정하였다.

LINPACK은 선형 시스템의 해를 구하는데 널리 사용하는 package로 Blas 같은 라이브러리를 이용하여 Ax=B 형태의 주어진 연립 방정식의 해를 구함으로써 컴퓨터의 성능을 측정하는데 해를 구하는 연산이 대부분 부동 소수점(Floating-point)로 구성된다. 슈퍼컴퓨터 성능 측정 소프트웨어 도구로도 잘 알려진 HPL은 Top 500에서 컴퓨터 선정에 사용되며, 성능 측정을 위해서는 MPI가 설치되어야 하며 BLAS, CBLAS, ATLAS와 같은 수학 라이브러리와 컴파일러 등의 설치가 선행되어야 한다. 사용자의 실습 환경인 가상 머신의 성능 측정을 위해 <표 1>과 같이 소프트웨어 스택을 구축하였다.

<표 1> 실습환경 소프트웨어 스택

운영체제(OS)	CentOS6.5 (Kernel 2.6.32-431)
컴파일러	gcc(version 4.4.7)
MPI라이브러리	Openmpi (version 1.8.1)
수학라이브러리	GotoBLAS2
벤치마크 툴	HPL (version 2.1)

<Table 1> Software stack of Practical System

HPL benchmark는 MPI를 기반으로 동작하며 정밀한 수학 계산을 위한 library package를 필요로 한다. 수학 library package로는 위의 표와 같이 BLAS library를 생성하는 프로그램인 GotoBLAS2[13]를 사용하였다. 실험을 위해 설치된 이 소프트웨어 패키지는 n차 선형방정식 Ax=B 의 n by n+1 계수행렬(coefficent matrix)을 LU 인수분해(LU factorization)로 계산하여 풀게 된다.

<표 2> HPL 벤치마크 옵션

옵션	설 명	옵션	설 명
N	Problem size	Pmap	Process mapping
NB	Blocking factor	threshold	for matrix validity test
P	Rows in process grid	Ndiv	Panels in recursion
Q	Columns in process grid	Nbmin	Recursion stopping criteria
Depth	Lookahead depth	Swap	Swap algorithm
Bcasts	Panel broadcasting method	L1, U	to store triangle of panel
Pfacts	Panel factorization method	Align	Memory alignment

<Table 2> List of HPL benchmark option

<표 2>와 같이 HPL 벤치마크에서는 다양한 옵션을 줄 수 있는데 성능 측정 결과에 큰 영향을 주는 것은 문제 크기 N과 프로세스 격자 P, Q 그리고 블록사이즈 NB 값이다[14]. 노드 수와 프로세스 수를 증가시켜 가며 다양한 실험을 수행하였다. 문제의 크기 N은 가상 머신에 할당된 메모리의 크기에 따라 좌우되며 시스템에서 기본적으로 사용하는 메모리 사용량이 있기 때문에 일반적으로 전체 메모리의 대략 80% 정도를

잡아서 테스트하게 된다. 메모리의 크기는 배경 밀도(double precision) 경우 8Byte 단위로 처리가 가능하므로 아래와 같이 계산되며 실험에 사용된 문제의 크기 또한 아래의 식을 근거로 수행되었다[13].

$$Ns \approx 0.8 \sqrt{\frac{Total\ Memory\ Size\ (bytes)}{8\ (double\ precision)}}$$

4.3 성능분석 및 실험 결과

본 실험을 위하여 성능이 다른 두 개의 가상화 기반 실험환경을 구축하였다. <표 3>은 각각의 실험환경(가상머신)의 성능을 나타내고 있으며 여기에는 이론성능과 HPL을 수행하는데 필요한 문제의 크기 기준을 4.2절에서 언급한 수식을 이용하여 나타내었다.

<표 3> 실험환경(가상머신) 성능 요약

실험환경 (가상머신)	CPU	Memory (MB)	이론성능 (GFlops)	HPL Problem Size(N)
vm1	1	1,024	10.64	9,268
vm2	2	2,048	21.28	13,107

<Table 3> Summary of the Practical System(VM)

두 가상머신에서 HPL 벤치마크를 수행했을 때 top 명령어를 통해 (그림 3)과 같이 CPU개수, 메모리 크기, CPU 사용률, 수행중인 HPL 프로세스 등을 확인할 수 있다.

(그림 3) top 명령어를 통한 HPL 수행 내역

```
[root@vm1 Intel64bmt]# top
top - 01:15:19 up 47 min, 3 users, load average: 1.00, 1.00, 0.87
Tasks: 140 total, 2 running, 138 sleeping, 0 stopped, 0 zombie
Cpu(s): 99.0%us, 1.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%st, 0.0%si, 0.0%rt
Mem: 1019328k total, 539276k used, 480052k free, 23396k buffers
Swap: 2064376k total, 0k used, 2064376k free, 163260k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
 2310 peter    20   0 638m 147m 3676 R 99.6 14.8  36:28.83 xhpl

[root@vm2 ~]# top
top - 09:14:12 up 2 min, 2 users, load average: 1.02, 0.38, 0.14
Tasks: 132 total, 3 running, 129 sleeping, 0 stopped, 0 zombie
Cpu(s): 100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%st, 0.0%si, 0.0%rt
Mem: 1921468k total, 1181028k used, 740440k free, 21440k buffers
Swap: 2064376k total, 0k used, 2064376k free, 123464k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
 7955 root     20   0 667m 420m 3784 R 100.1 22.4  0:31.94 xhpl
 7954 root     20   0 668m 420m 3820 R 99.8 22.4  0:31.79 xhpl
```

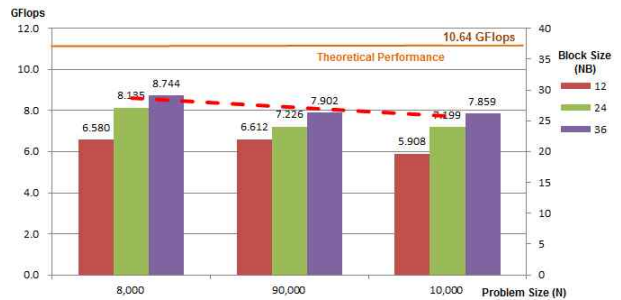
(Figure 3) HPL execution status from top command

<표 4> 실험환경 벤치마크 수행결과(GFlops)

실험환경	N \ NB	12	24	36
VM1	8,000	6.580	8.135	8.744
	9,000	6.612	7.226	7.902
	10,000	5.908	7.199	7.859
VM2	10,000	11.990	14.490	15.720
	12,000	11.710	14.490	16.760
	13,000	11.560	15.430	16.760
	14,000	12.350	15.520	17.260

<Table 4> Result of HPL benchmark(GFlops)

(그림 4) VM1에서 HPL 성능 분석



(Figure 4) HPL evaluation on the VM1

(그림 5) VM1에서 HPL 성능 분석



(Figure 5) HPL evaluation on the VM2

위 실험을 통해 실험 환경에서 얻을 수 있는 최대의 성능을 얻기 위한 문제의 사이즈(N)을 유추할 수 있었고 실제 벤치마크실험을 통해 이론 성능 대비 최대 80% 이상 실측 성능을 얻을 수 있었다. 가상화 특징상 리눅스 커널과 하이퍼바이저에서 사용하는 리소스를 감안하면 매우 양호한 결과이다. 가상화 기반 실험환경이 다양한 자원의 특성을 가지고 생성되더라도 성능측적을 위한 문제의 크기를 유추하여 성능을 평가함으로써 가상머신의 문제점을 파악하고 진단할 수 있다.

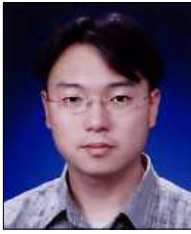
5. 결론

본 논문에서는 클라우드 컴퓨팅의 가상화 속성을 이용하여 다양한 교육 실습 환경에 접목함으로써 사용자가 요구하는 환경을 유연하고 신속하게 제공할 수 있음을 살펴보았다. 빈번하게 실습 환경 제공을 위해 구축해야할 여러 시스템 환경 또한 가상 어플라이언스(Virtual Appliance)로 이미지화하여 설정함으로써 관리의 부하를 축소시킬 뿐만 아니라 실습 환경의 규모에 대해서도 유연하게 대처할 수 있다. 나아가 다양한 가상화 기반의 실습환경이 구축되면서 발생할 수 있는 성능의 문제점을 실제로 측정하고 파악함으로써 안정적인 실습환경을 구축할 수 있음을 살펴볼 수 있었다. 더욱이 가상 어플라이언스를 생성하여 실습환경에서 구동될 때 발생할 수 있는 자원의 요구사항들을 미연에 측정하여 최적화된 실습환경을 제공할 수 있다는 장점은 클라우드 컴퓨팅의 개념에 더욱 부합된다. 한편, 본 실험을 통해서 가상화 실습 환경의 성능을 이론 대비 약 80%까지 얻을 수 있었고, 최대한의 성능을 얻기 위한 문제 즉, 시스템 부하의 크기를 유추 하는 과정을 알아낼 수 있었다.

향후, 다양한 벤치마크 도구를 적용하여 가상 실습환경의 성능을 비교, 분석하여 보는 연구가 필요하며, 계산성능 뿐만 아니라 파일시스템의 I/O, 네트워크의 대역폭 등 여러 각도에서 실습 환경에 대한 성능 평가가 필요하겠다.

References

- [1] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. In Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC, NCM '09, pages 44-51, 2009.
- [2] J. Diaz, G. v. Laszewski, F. Wang, and G. Fox, "Abstract Image Management and Universal Image Registration for Cloud and HPC Infrastructures " IEEE Cloud. 2012.
- [3] Daniel J. Abadi, "Data Management in the Cloud: Limitations and Opportunities", In IEEE DE Bulletin, vol 32(1), pp 3-12, Feb 2009.
- [4] G.Y. LEE, H.K. Choi, "Capacity Analysis of University Cloud Computer for Integrating Academic Affairs Business", Journal of Digital Contents Society Vol. 15 No. 3 June, pp. 413-423, 2014.
- [5] B.Sotomayor, R. S. Montero, I. M. Llorente, and IFoster, "Virtual Infrastructure Management in Private and Hybrid Clouds," IEEE Internet Computing, vol. 13, pp. 14-22, Sep-Oct 2009.
- [6] Nimbus Project Web Page. <http://www.nimbusproject.org>
- [7] Eucalyptus Web Pages (Open Source). <http://open.eucalyptus.com>
- [8] Nurmi, D.Wolski, R. Grzegorzcyk, C. Obertelli, "The Eucalyptus Open-Source Cloud-Computing System ", CCGRID '09. 9th IEEE/ACM International Symposium, Cluster Computing and the Grid, 2009. pp.124 - 131, 2009.
- [9] OpenNebula Web Page. <http://www.opennebula.org>
- [10] OpenStack Home Page. <http://www.openstack.org>
- [11] Amazon Elastic Compute Cloud (EC2). <http://aws.amazon.com/ec2>
- [12] PETITET, Antoine. HPL-a portable implementation of the high-performance Linpack benchmark for distributed-memory computers. <http://www.netlib.org/benchmark/hpl/>, 2004.
- [13] Goto, Kazushige, and K. Milfeld. "GotoBLAS2.", <https://www.tacc.utexas.edu/tacc-projects/gotoblas2>, 2011.
- [14] Dunlop, Dominic, Sébastien Varrette, and Pascal Bovvry. "On the use of a genetic algorithm in high performance computer benchmark tuning", Parallel Processing and Applied Mathematics. Springer Berlin Heidelberg pp. 102-114, 2010



윤 준 원

2004년 : 고려대학교 대학원 컴퓨터
학과(이학석사)
2011년 : 고려대학교 대학원 컴퓨터
학과 박사 수료
2005년~현재 : KISTI 국가슈퍼컴퓨
팅연구소 선임연구원

관심분야 : 그리드 컴퓨팅, 분산 컴퓨팅, 결합포용시스
템, 클라우드 컴퓨팅, 슈퍼컴퓨팅



송 의 성

1997년 : 고려대학교 컴퓨터학과(학
사)
1999년: 고려대학교 대학원 컴퓨터
학과(이학석사)
2005년: 고려대학교 대학원 컴퓨터
학과(이학박사)

2006년~현재: 부산교육대학교 컴퓨터교육과 교수
관심분야: 컴퓨터교육, 교육용로봇교육, 컴퓨터네트워크,
스마트러닝