
클라우드 컴퓨팅 환경에서 가상머신 할당기법 및 임대 서비스 구현

황인찬* · 이봉환**

Implementation of Virtual Machine Allocation Scheme and Lease Service in Cloud
Computing Environments

In-Chan Hwang* · Bong-Hwan Lee**

이 논문은 2009년도 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임
(No. 2009-0076877)

요 약

오픈 소스 클라우드 컴퓨팅 플랫폼인 OpenNebula를 이용한 클라우드 컴퓨팅 환경에서 가상머신 임대 서비스를 구현하고 클라우드 자원 관리와 서비스 사용의 편의성을 위하여 웹기반 클라우드 사용자 인터페이스를 구현하였다. OpenNebula의 가상머신 할당 기법은 가상화 소프트웨어의 CPU 할당 스케줄러를 고려하지 않아 성능 저하의 요인이 되고 있다. 이러한 문제점을 해결하기 위하여 클러스터 노드의 유휴 CPU 자원의 우선순위와 Xen의 Credit 스케줄러를 고려하여 OpenNebula의 가상머신 할당 스케줄러의 성능을 개선하였다. 실험 결과 제안한 가상머신 할당 기법은 기존 방식에 비하여 수용 가능한 가상머신 수와 CPU 자원 할당량에서 향상된 결과를 보였다.

ABSTRACT

A virtual machine lease service in the cloud computing environment has been implemented using the open source cloud computing platform, OpenNebula. In addition, a web-based cloud user interface is developed for both convenient resource management and efficient service access. The present virtual machine allocation scheme adopted in OpenNebula has performance reduction problem because of not considering CPU allocation scheduler of the virtualization software. In order to address this problem we have considered both the priority of the idle CPU resources of the cluster and credit scheduler of Xen, which resulted in performance improvement of the OpenNebula virtual machine scheduler. The experimental results showed that the proposed allocation scheme provided more virtual machine creations and more CPU resource allocations for cloud service.

키워드

클라우드 컴퓨팅, 가상머신, 임대서비스, OpenNebula, Xen

Key word

Cloud Computing, Virtual Machine, Lease Service, OpenNebula, Xen

* 대전대학교 정보통신공학과
** 대전대학교 (교신저자, blee@dju.kr)

접수일자 : 2010. 01. 08
심사완료일자 : 2010. 02. 09

I. 서 론

클라우드 컴퓨팅은 서로 다른 물리적 위치에 존재하는 다양한 도메인, 다양한 종류의 컴퓨팅 및 스토리지 자원을 통합하여 가상화된 고성능 컴퓨팅 자원 집합체를 구축하고 다수의 고객들에게 높은 수준의 확장성을 가진 IT 자원들을 온-디맨드 (On-demand) 방식으로 제공하여 자원 효율성 극대화과 관리의 최소화라는 장점을 가지는 새로운 컴퓨팅 패러다임이다[1]. 클라우드 컴퓨팅은 구체적으로 제공하는 서비스의 유형에 따라 SaaS (Service as a Service)[2], PaaS (Platform as a Service)[3] 및 IaaS (Infrastructure as a Service)[4] 등 세 가지로 구분할 수 있으며, 본 논문에서 구현한 가상머신 임대 서비스는 IaaS에 속한다[5]. IaaS는 저장 장치 또는 컴퓨팅 능력을 인터넷을 통한 서비스 형태로 제공하는 것으로 아마존 EC2 [6,7], Nimbus[8], OpenNebula[9,10], Eucalyptus[11] 등이 있으며, 본 논문에서는 오픈 소스 클라우드 컴퓨팅 플랫폼인 OpenNebula를 활용하여 클라우드 컴퓨팅 환경을 구축하고, 가상머신 임대 서비스의 관리·운영에 편의성 제고를 위하여 웹기반 클라우드 사용자 인터페이스를 구현하였다. 또한 OpenNebula의 스케줄러 성능 개선과 시스템의 활용성 향상을 위하여 클러스터 노드의 유휴 CPU 자원 우선순위와 Xen의 Credit 스케줄러를 고려한 가상머신 할당 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대하여 기술하고, 3장에서는 클라우드 컴퓨팅 구축 및 가상머신 임대 서비스 구현에 대해 기술하며, 4장에서는 본 연구에서 제안하는 가상머신 할당 기법에 대해 설명한다. 5장에서는 시스템 활용률에 대한 실험을 통하여 성능을 분석하고 마지막으로 6장에서는 결론 및 향후 연구내용에 대해 기술한다.

II. 관련 연구

클라우드 컴퓨팅을 구성하는 물리적인 컴퓨팅 자원들에 가상화 기술을 적용하여 다수의 사용자에게 서버, 스토리지, 네트워크 등의 자원을 제공하기 위한 가상화 기술은 하나의 물리적인 컴퓨터에서 여러 종류의

운영 체제를 사용할 수 있게 하기 위하여 각 운영체제가 하드웨어를 공유할 수 있도록 기능을 제공하는 소프트웨어 기술로 그림 1과 같이 다수의 서비스 플랫폼을 하나의 플랫폼으로 통합하는 것을 가능하게 한다 [12]. 기존 데이터센터 환경에서는 개별적으로 운영되는 서버들의 컴퓨팅 자원 활용률이 하드웨어 성능에 비해 크게 떨어져 자원낭비가 발생하였다. 그러나 서버통합 환경에서 공간 및 컴퓨팅 자원의 낭비를 방지할 수 있는 가상화 기술은 데이터센터 환경에서 주목 받고 있다.

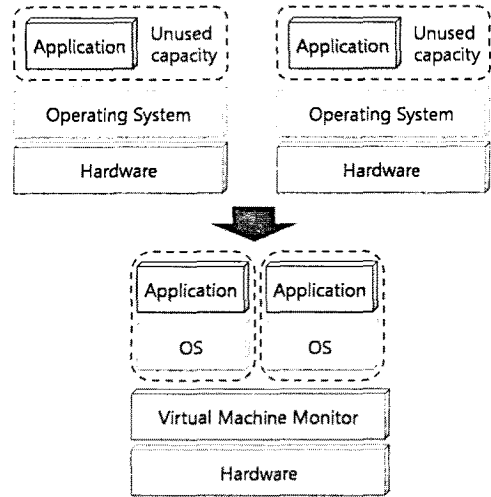


그림 1. 가상화 소프트웨어를 사용한 서버 통합 환경
Fig.1. Server Integration Environment using Virtualization Software

가상화 기술은 현재 여러 벤더에서 연구 및 상품화를 진행하고 있다. 그 중 시스템 가상화 도구로 보편적으로 사용되는 것으로는 VMware [13]와 Xen [14]을 들 수 있으며, 가상화의 형태에 따라 전가상화(Full Virtualization)와 반가상화(Para Virtualization)로 나뉜다. 전가상화는 하드웨어 전체를 모두 가상화하는 방법으로 VMware의 가상화 방법이 이에 속한다. 반가상화는 운영체제를 가상화 지원이 가능할 수 있도록 수정하여 시스템 부팅 시 적용하는 방법으로서 CPU와 메모리 등 일부 장치만을 가상화하며, Xen의 가상화 방법이 이에 속한다. 본 논문에서 물리적인 자원의 가상화를 위하여 사용한 가상화 도구는 Xen이며, Xen의 구조는

VMM(Virtual Machine Monitor)으로도 불리는 하이퍼바이저(Hypervisor)와 호스트 OS(Dom-0), 게스트 OS(Dom-U)로 구성된다. 하이퍼바이저는 하드웨어 상에서 CPU와 메모리의 자원을 게스트 OS에 할당해주는 역할을 한다. 호스트 OS는 도메인-0로 불리며, CPU와 메모리 외에 게스트 OS가 디바이스를 접근하기 위해서는 이를 거쳐야 한다. 또한 시스템 부팅 시 먼저 시작하여 게스트 도메인의 생성 및 제거 등 관리 작업을 수행하며, 게스트 OS는 가상머신을 의미한다. 그림 2는 Xen의 구조를 나타낸다.

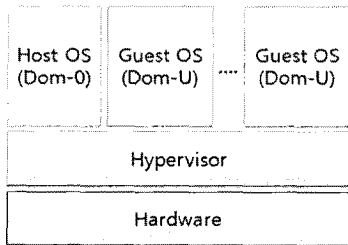


그림 2. Xen의 구조
Fig.2. Xen Architecture

최신 버전의 Xen 하이퍼바이저에는 생성된 각 도메인에 CPU 자원 할당을 결정하는 스케줄러로서 기본적으로 Credit 스케줄러 [15]를 사용하고 있다. Credit 스케줄러는 다중 프로세스 환경에서 로드 밸런싱을 제공하며, 각 도메인에 배분 비율에 따라 CPU를 할당해주는 비례 공유 스케줄링 방법이다. 각 도메인은 Weight 값과 Cap 값을 가지고 있으며, Weight 값에 의해서 모든 도메인에 CPU 자원을 비례하여 할당해주고 Cap 값은 CPU 자원 이용률의 상한을 두고자 할 경우에만 사용된다. 따라서 Cap 값이 설정되어 있으면 도메인은 그 이상의 CPU 대역폭을 할당 받을 수 없게 된다.

III. 클라우드 컴퓨팅 플랫폼 구축 및 가상머신 임대 서비스 구현

가상머신은 실제 컴퓨터와 같이 프로그램들을 수행할 수 있는 소프트웨어 기반 머신의 구현으로 운영체제

의 완전한 실행을 지원할 수 있는 시스템을 말한다. 가상머신의 사용은 하나의 컴퓨터 안에 독립적인 여러 개의 운영체제 환경을 구현할 수 있고 자원을 분할하여 할당하고 이용할 수 있는 이점을 제공한다.

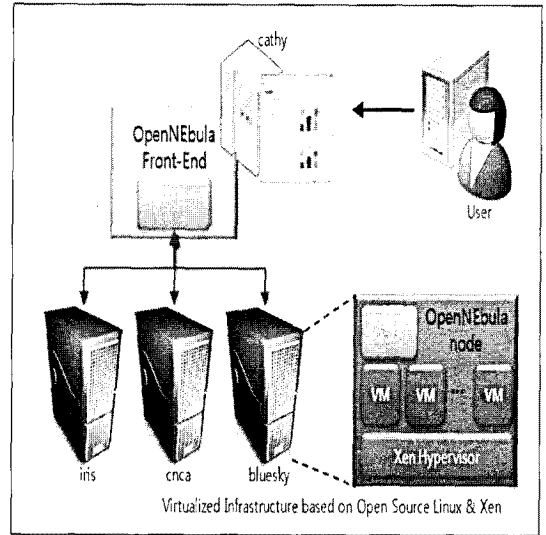


그림 3. 클라우드 컴퓨팅 플랫폼
Fig.3. Cloud Computing Platform

본 장에서는 가상머신 임대 서비스 구현을 위하여 구축한 시스템 구성도와 웹기반 클라우드 사용자 인터페이스에 대하여 설명한다. 그림 3은 구축한 클라우드 컴퓨팅 플랫폼의 전체 구성도이다.

전체 시스템은 저가의 PC 사양 컴퓨터를 사용하여 클라우드 컴퓨팅 환경을 구축하는 것을 목표로 하며, 본 연구에서는 소규모 사설 클라우드 컴퓨팅 환경을 구축하였다.

표 1. 전체 시스템 사양
Table. 1. System Specifications

	CPU	RAM	HDD	OS
Front-End	Dual Core 2.13GHz	2.0GB	230GB	ubuntu9.04 i686
Cluster Node 1, 2, 3	Core 2 Duo 2.80GHz	8.0GB	460GB	ubuntu8.04 i686

전체 시스템은 표1과 같이 총 4대의 서버로 구성되며, 1대의 Front-End와 3대의 클러스터 노드로 구축하였다. 시스템에는 OpenNebula(이하 ONE)를 사용하였고 Front-End에는 ONE의 데몬이 실행되어 각각의 클러스터 노드와 생성된 가상머신의 모니터링 기능, 가상머신 설치를 위한 이미지 파일 관리 기능 등을 수행한다. 각각의 클러스터 노드는 실제 가상머신이 생성될 노드로서 Xen을 설치하여 가상화하였다.

가상머신 임대 서비스 제공을 위한 클라우드 컴퓨팅 운영을 위해서는 먼저 시스템에 가상 네트워크와 가상머신 이미지 파일이 준비되어 있어야 한다. 가상 네트워크는 하나의 C 클래스 IP 네트워크를 범위로 정의하였으며, 가상머신 이미지 파일은 Ubuntu 9.04 버전의 977MByte 저장 공간을 사용하였다. 이후 사용자가 웹기반 클라우드 사용자 인터페이스를 통하여 가상머신 생성을 요청하면 요구사항을 템플릿 파일로 작성하여 데이터베이스에 저장한다. 이후 스케줄러는 사용자의 요구사항에 적합한 클러스터 노드를 선정하고 해당 클러스터 노드에 가상머신 이미지 파일과 요청서를 전송한다. 이를 이용하여 Xen은 가상머신을 생성하게 되고 사용자는 생성된 가상머신에 SSH 또는 VNC(Virtual Network Computing)을 통하여 접속한 뒤 사용할 수 있다. 이러한 과정을 그림 4에 나타내었다.

ONE에서 기본적으로 제공하는 인터페이스인 CLI(Command Line Interface)는 명령어 입력 방식으로써 클러스터 노드와 가상머신을 관리하는 방법으로 Front-End에 접속을 위하여 SSH 통신을 위한 별도의 툴을 필요로 한다. 따라서 가상머신 생성 시 마다 템플릿을 새로 작성해야하는 번거로움이 있다. 이러한 관리 방법의 불편함을 해소하기 위하여 웹기반 클라우드 사용자 인터페이스를 구현하였다. 이는 실시간으로 클러스터 노드와 가상머신의 상세정보를 모니터링 할 수 있게 하며, 가상머신 생성과 관리의 편의를 제공한다. 그림 5는 총 5대의 가상머신이 사용자에게 의해 생성된 결과를 보여주는 웹기반 클라우드 사용자 인터페이스의 메인 화면이다.

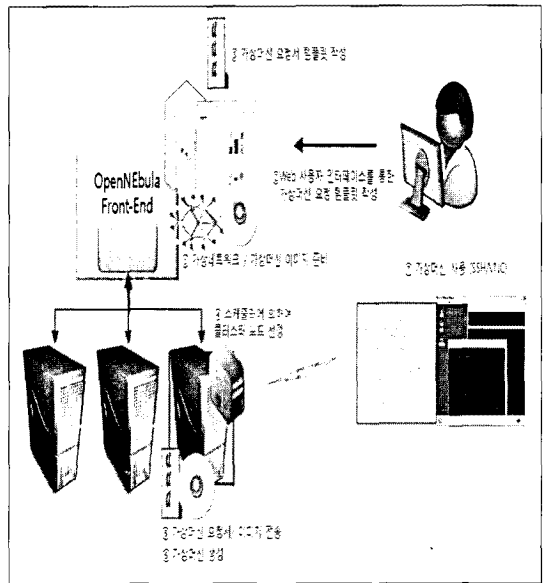


그림 4. 가상머신 임대 서비스 개념도
Fig.4. Conceptual Block Diagram of Virtual Machine Lease Service

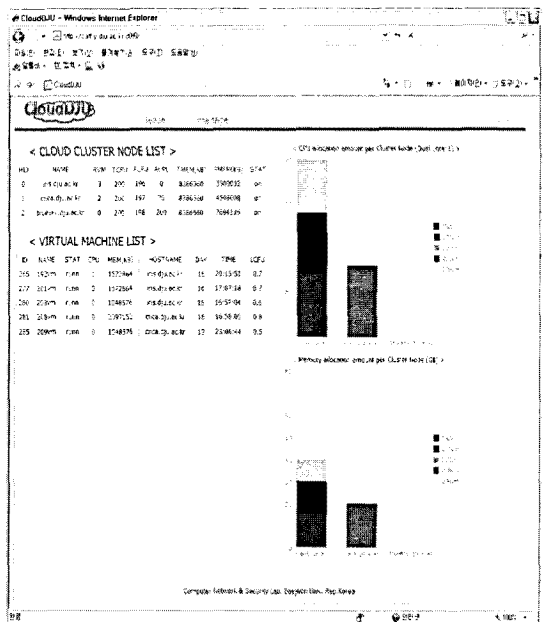


그림 5. 웹기반 클라우드 컴퓨팅 사용자 인터페이스
Fig.5. Web-based Cloud Computing User Interface

IV. 제안하는 가상머신 할당 기법

클라우드 자원을 관리·유지하는 입장에서 클라우드 운영자의 경제적 이익과 밀접한 관련을 갖는 자원 관리 기술은 최소한의 비용으로 클라우드 사용자의 요구사항에 따라 자원을 동적인 방법을 통하여 효과적으로 제공할 수 있어야 한다. 이에 본 논문에서는 Xen의 Credit 스케줄러와 클라우드 클러스터 노드의 CPU 활용률을 고려하여 사용자가 요청한 CPU의 요구사항을 만족시키며 전체 시스템이 수용할 수 있는 가상머신 수와 CPU 자원 할당량을 증가시킬 수 있는 가상머신 할당 방법을 제안한다.

Xen의 Credit 스케줄러는 CPU 자원에 대하여 비례공유 방법을 이용하기 때문에 실제로 요청된 가상머신의 CPU 자원 할당량(LCPU)과 사용할 수 있는 가상머신의 CPU 자원 사용량과는 차이가 있다. 생성된 가상머신은 최초 요청된 LCPU는 보장하면서 이후 LCPU 보다 큰 작업이 요구될 경우 클러스터 노드의 유휴 CPU 자원을 비례 공유하여 사용 가능하다. 이와 같은 유연한 CPU 스케줄링 방법은 가상머신을 사용하는 사용자의 요구사항을 보장하고 보다 질 높은 서비스를 제공할 수 있다. 그러나 클러스터 노드의 측면에서 LCPU 이상의 작업을 수행하는 것은 새로운 가상머신에게 할당 가능한 CPU 자원 할당량(ACPU)이 충분함에도 불구하고 새로운 가상머신을 생성하지 못하는 경우가 발생할 수 있다. 본 논문에서는 이러한 경우 해당 클러스터 노드의 모든 가상머신에 Cap을 설정하여 유휴 CPU 자원을 확보하는 방법으로 해결하고자 한다.

가상머신의 생성 과정은 두 단계로 구분할 수 있다. 그림 6은 사용자의 가상머신 요구사항에 적합한 클러스터 노드를 선정하는 단계이며, 그림 7은 선정된 클러스터 노드에 가상머신을 생성하는 단계이다.

먼저 사용자에 의해 가상머신의 생성이 요청되면 클러스터 노드들을 ACPU가 작은 순으로 정렬한다. 정렬된 클러스터 노드 리스트의 최상위부터 클러스터 노드를 선택하여 클러스터 노드가 가상머신에게 할당 가능한 남아있는 CPU 자원 할당량(ACPU)과 가상머신 생성에 요구된 CPU 자원 할당량(LCPU)을 비교하여 ACPU가 부족하다면 다음 순위의 클러스터 노드를 비교하고 그렇지 않다면 실제 클러스터 노드의 유휴 CPU 자원 할

당량(FCPU)과 LCPU를 다시 비교한다.

이 단계에서 FCPU가 부족하다면 기존에 생성되어 있던 가상머신이 요구사항 이상의 CPU 자원을 점유하고 있는 것이므로 각각의 가상머신에 대하여 LCPU 만큼의 Cap을 설정하여 FCPU를 확보한다.

다음 ACPU와 FCPU가 부족하지 않다면 가상머신의 요구사항을 데이터베이스에 저장하고 대기한다. 이후에 스케줄러는 데이터베이스에서 대기 중(PENDING)인 가상머신을 생성하기 위하여 ONE의 가상머신 Life-cycle 관리 기능을 수행하며, 가상머신을 생성하는 과정에서 가상머신에 Cap이 설정되어 있다면 해제한다.

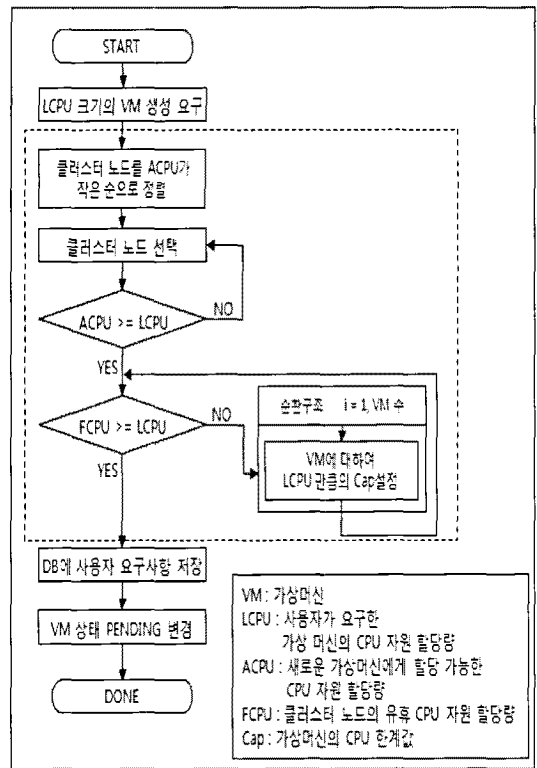


그림 6. 클러스터 노드 선정과 FCPU 확보 과정
Fig.6. Selection of Cluster Node and FCPU Acquisition Process

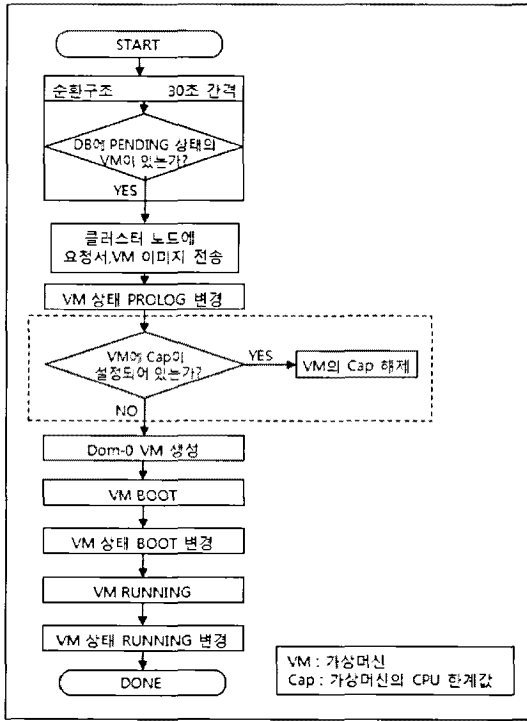


그림 7. Cap 해제와 가상머신 생성 과정
Fig.7. Cap Release and Virtual Machine Creation Process

이와 같이 정상 시에는 Cap 설정을 피하여 가상머신이 가능한 최대의 CPU 자원을 사용하게 하고 필요시 클러스터 노드의 FCPU 확보를 위해 일시적으로 Cap을 설정한 후 해제하는 방법을 사용하면 가상머신은 요구 사항 이상으로 CPU 성능을 이용할 수 있게 된다. 또한 클러스터 노드는 CPU 활용률을 높일 수 있으며, 전체 시스템이 수용할 수 있는 가상머신 수와 CPU 자원 할당량을 증가시킬 수 있다.

V. 시스템 활용률에 대한 실험 및 성능 분석

본 장에서는 구축한 클라우드 컴퓨팅 환경에서 가상머신 임대 서비스를 제공함에 있어 시스템이 수용할 수 있는 가상머신의 수와 CPU 자원 할당량을 증가시키기

위하여 제안하는 가상머신 할당 방법과 ONE의 스케줄러에 의한 가상머신 할당 방법을 비교하여 실험하고 그 결과를 분석함으로써 제안하는 시스템의 유용성을 평가하였다.

실험을 위하여 요구된 가상머신의 CPU 성능은 랜덤하게 생성하며, 요청된 CPU 자원의 합은 전체 클러스터 노드의 CPU 자원 할당량인 6.0으로 한다. 또한 요청된 가상머신의 메모리 할당량은 512MByte로 동일하다.

생성된 가상머신은 CPU 위주의 작업을 수행하며, 각각 idle 상태, 0.2, 0.4, 0.6, 0.8, 1.0의 CPU 자원 사용량을 요구하는 6단계로 나누어 실험하고 각각 30번씩 총 180번을 실험한다. 성능 분석으로는 시스템이 수용한 가상머신의 개수, 사용자의 요구사항인 LCPU를 기준으로 CPU 자원 할당량 단계 별 평균, 전체 평균을 계산하고 이를 시스템 활용률로 평가한다.

ONE의 가상머신 할당 스케줄러는 사용자의 요청에 의하여 가상머신이 생성될 클러스터 노드를 선정함에 있어서 각각의 클러스터 노드의 유휴 공간만을 고려하며, 동적인 클러스터 노드 간 우선순위를 부여하지 않고 가상머신을 할당하기 때문에 가상머신이 할당될 클러스터 노드의 선정이 효과적이지 못하다. 또한 Xen의 Credit 스케줄러의 Cap 값을 고려하지 못하기 때문에 LCPU 이상의 작업을 수행중인 가상머신이 존재 할 경우에는 ACPU가 충분함에도 불구하고 더 이상 가상머신을 생성할 수 없다.

랜덤하게 가상머신을 요청하였을 경우 idle 상태에서 하나의 클러스터 노드에는 3.37대의 가상머신이 생성되었다. 또한 ONE의 스케줄러를 이용하여 가상머신을 할당하고 가상머신이 요청한 작업이 CPU 자원 사용량 0.8 이상일 경우에는 하나의 클러스터 노드에서 2.79대의 가상머신 밖에 수용할 수 없게 되었다. 이 결과는 가상머신이 생성된 후에 큰 작업을 수행할수록 전체 시스템이 수용할 수 있는 가상머신의 수와 CPU 자원 할당량이 감소하게 됨을 알 수 있다.

표 2는 각 단계별 30번의 실험과 전체 180번의 실험에 대한 시스템이 수용한 가상머신의 수와 CPU 자원 할당량에 대한 평균을 ONE 스케줄러와 비교한 결과이다.

표 2. 각 단계별 시스템이 수용한 평균 가상머신의 수 및 CPU 자원 할당량

Table 2. Allocated Average Number of Virtual Machines and CPU Resources

구분	ONE 스케줄러를 이용한 가상머신 할당 방법		제안하는 가상머신 할당 방법	
	수용한 VM 수	수용한 CPU 자원 할당량	수용한 VM 수	수용한 CPU 자원 할당량
idle	10.10	5.49	10.37	5.67
0.2	10.53	5.48	10.63	5.59
0.4	9.97	5.38	10.13	5.57
0.6	9.27	4.93	10.27	5.54
0.8	8.37	4.24	10.37	5.60
1.0	6.0	3.12	11.27	5.71
평균	9.04	4.77	10.51	5.61

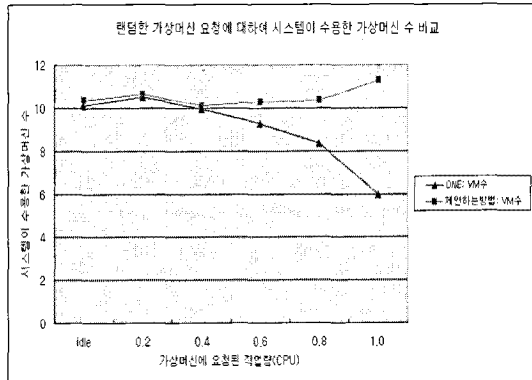


그림 8. 단계별 시스템이 수용한 가상머신 수 비교
Fig.8. Comparison of Allocated Virtual Machines

그림 8은 ONE의 스케줄러와 제안하는 가상머신 할당방법을 통한 실험 결과 각 단계별로 시스템이 수용한 평균 가상머신 수를 나타낸 것이며, 그림 9는 CPU 자원 할당량이다. 그림 10은 180번의 전체 실험에 대하여 시스템이 수용한 평균 가상머신 수이며, 그림 11은 평균 CPU 자원 할당량이다. 시스템이 수용한 평균 가상머신 수는 16.23%, 평균 CPU 자원할당량은 17.6%의 증가를 보였다.

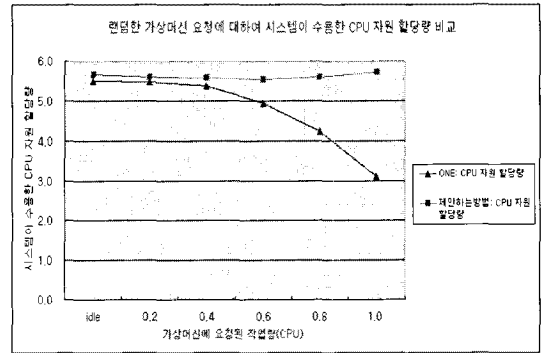


그림 9. 단계별 시스템이 수용한 CPU 자원 할당량 비교

Fig.9. Comparison of Allocated CPU Resources

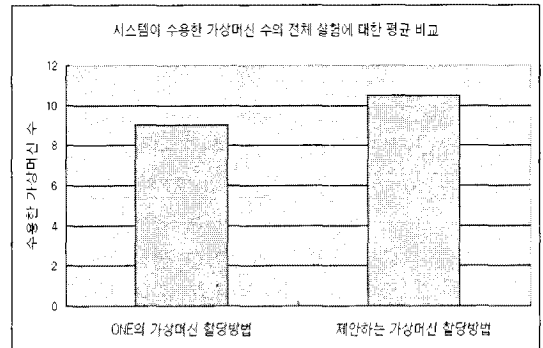


그림 10. 전체 실험에 대하여 수용한 가상머신 수의 평균 비교

Fig.10. Comparison of Overall Virtual Machines

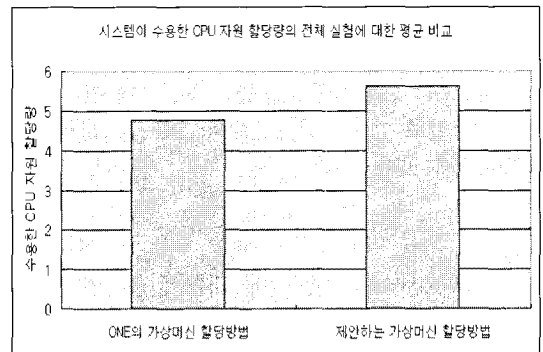


그림 11. 전체 실험에 대하여 수용한 CPU 자원 할당량 평균 비교

Fig.11. Comparison of Overall Allocated CPU Resources

그러나 제안하는 가상머신 할당 방법은 클라우드 컴퓨팅 서비스 제공자와 사용자간에 LCPU 크기의 CPU 자원 할당을 보장하는 것을 최우선으로 하고 있으므로 가상머신이 LCPU 이상의 작업 요청 시에 LCPU 이상의 CPU 작업량에 대한 보장은 항상 하고 있지 않다. LCPU 이상의 CPU 작업량을 보장할 수 있는 경우는 가상머신이 할당 되어 있는 해당 클러스터 노드에 유휴 CPU 자원이 있을 경우이다. 따라서 성능 향상 결과는 가상머신 자체의 성능 향상이 아니라 클라우드 컴퓨팅의 목적인 더 많은 수의 서버나 개인 컴퓨터를 통합 가능하여 IT 자원의 효율성을 향상 시키는 측면과 클라우드 서비스 제공자의 경제적 이익 증대와 관련이 있다.

VI. 결 론

본 논문에서는 오픈 소스 클라우드 컴퓨팅 플랫폼인 OpenNebula를 이용하여 클라우드 컴퓨팅 환경을 구축하고, 가상머신 임대 서비스의 관리·운영에 편의성 제고를 위하여 웹기반 클라우드 사용자 인터페이스를 구현하였다. 또한 OpenNebula의 스케줄러 성능 개선과 시스템의 활용성 향상을 위하여 클러스터 노드의 유휴 CPU 자원 우선순위와 Xen의 Credit 스케줄러를 고려한 가상머신 할당 방법을 제안하고 그 성능을 비교 분석하였다. 랜덤하게 가상머신을 요청하는 실험을 통해 제안하는 가상머신 할당 기법이 전체 시스템이 수용할 수 있는 가상머신 수와 CPU 자원 할당량에서 향상된 성능을 보였다.

향후 연구내용으로 소규모 단일 플랫폼을 벗어나 타 플랫폼과 연동 가능한 대규모 클라우드 컴퓨팅 환경 구축 및 가상머신 할당 기법의 성능 평가에 대한 연구가 필요하다.

참고문헌

- [1] 김창수, 김학영, 남궁한, "클라우드 서비스를 위한 대규모 클러스터 관리 기술 개발," 전자통신동향분석, Vol. 24, No. 4, pp. 89-98, 2009. 8.
- [2] Dave Thomas, "Enabling Application Agility - Software as a Service, Cloud Computing and Dynamic Languages," Journal of Object Technology, Vol. 7, No. 4, 2008. 5.
- [3] 민옥기, 김학영, 남궁한, "클라우드 컴퓨팅 기술 동향," 전자통신동향분석, Vol. 24, No. 4, pp. 1-13, 2009. 8.
- [4] Michael Armbrust et al, "Above the Clouds: A Berkeley View of Cloud Computing," <http://radlab.cs.berkeley.edu>, 2009. 2.
- [5] M. D. Dikaiakos, D. Katsaros, G. Pallis, A. Vakali, and P. Mehra, "Cloud Computing," IEEE Internet Computing, Vol. 12, No. 5, 2009. 9.
- [6] Amazon, "Amazon Web Services: Overview of Security Process," <http://aws.amazon.com>, 2008. 9.
- [7] 한재선, "클라우드 컴퓨팅 플랫폼과 오픈 플랫폼 기술," 정보처리학회지, Vol. 16, No. 2, pp. 39-50, 2009. 3.
- [8] K. Keahey, T. Freeman, "Science Clouds: Early Experiences in Cloud Computing for Scientific Applications," Cloud Computing and Its Applications 2008 (CCA-08), Chicago, 2008. 10.
- [9] OpenNebula, <http://www.opennebula.org>
- [10] B. Sotomayor, R. S. Montero, I. M. Llorente and I. Foster, "Capacity Leasing in Cloud Systems using the OpenNebula Engine," Workshop on Cloud Computing and its Applications 2008 (CCA08), Chicago, 2008. 10.
- [11] Daniel Nurmi, Rich Wolski, Chris Grzegorzcyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov, "The Eucalyptus Open-source Cloud-computing System," Cloud Computing and Its Applications, 2008. 10.
- [12] M. Rosenblum, T. Garfinkel, "Virtual machine monitors: current technology and future trends," Computer (IEEE Computer Society), Vol. 38, Issue, pp. 39-47, 2005. 5.
- [13] VMware, <http://www.vmware.com>
- [14] A. Whitaker, M. Shaw, S. D. Gribble, "Lightweight Virtual Machines for Distributed and Networked Application," Technical Report University of Washington, 2002.

[15] Credit scheduler, <http://wiki.xensource.com/xenwiki/CreditScheduler>

저자소개



황인찬(In-Chan Hwang)

2008년 대전대학교 정보통신공학과
졸업(공학사)
2010년 대전대학교 정보통신시스템
공학 졸업(석사)

※ 관심분야: 클라우드컴퓨팅, BRMS 등



이봉환(Bong-Hwan Lee)

1985년 서강대학교 전자공학과 졸업
(공학사)
1987년 연세대학교 전자공학과 졸업
(석사)

1993년 Texas A&M 대학교 전기 및 컴퓨터공학과 졸업
(박사)

1995~현재 대전대학교 정보통신공학과 교수

※ 관심분야: 클라우드컴퓨팅, 유비쿼터스 헬스케어,
네트워크 보안 등