

# 공개SW기반 클라우드 컴퓨팅 기술 현황

Trend of Open Source SW-based Cloud Computing Technology

소프트웨어 기술의 미래전망 특집

정의정 (U.J. Jeong) 고성능컴퓨팅SW연구팀 UST연수생  
 강동재 (D.J. Kang) 고성능컴퓨팅SW연구팀 선임연구원  
 정성인 (S.I. Jung) 고성능컴퓨팅SW연구팀 책임연구원

## 목 차

- .....
- I . 서론
  - II . 클라우드 IaaS 구조와 공개SW
  - III . 클라우드 운영관리 기술
  - IV . 다중 클라우드 호환 기술

컴퓨팅 자원을 빌려 쓰는 시대를 여는 클라우드 컴퓨팅은 북미, 유럽뿐만 아니라 아시아 전 세계적으로 확산되고 있는 추세이다. 더구나 SW 기술개발의 새로운 방식으로 부상한 공개SW도 클라우드 컴퓨팅의 빠른 확산에 역할을 하며, 여러 분야의 클라우드 요소 기술 발전에도 기여하고 있다. 따라서 본 고에서는 오픈 프로젝트로 진화되고 있는 클라우드 기술을 살펴본다. 먼저, 클라우드 인프라스트럭처 서비스(IaaS: Infrastructure as a Service)를 제공하기 위한 클라우드 컴퓨팅 구조와 구성 요소 기술을 분석하고, 관련 공개SW 기술들을 분석한다. 여기서 IaaS 클라우드를 운영, 관리하는 기술을 보다 자세히 분석하고 비교한다. 마지막으로 클라우드 간 호환성 문제를 해결하는 공개SW 기술도 살펴본다.

## I. 서론

클라우드 컴퓨팅은 컴퓨팅 자원을 빌려 쓰는 시대를 열고 있다. 서버, 스토리지, SW 등 정보통신기술(ICT)의 자원을 구매하여 소유하지 않고, 필요 시 인터넷을 통해 서비스 형태(as a service)로 이용하는 방식이다. 모바일 시장의 성장과 함께 모바일과 결합하면서 ICT 분야의 새로운 패러다임으로 부상하고 있다. 클라우드 컴퓨팅 용어는 2006년 9월 구글 직원인 Christophe Bisciglia가 유희 컴퓨팅 자원에 대한 활용을 제안하면서 처음으로 사용하였다[1].

공개SW는 클라우드 컴퓨팅 용어가 출현하기 훨씬 이전인 1990년대부터 활성화되어, IT 산업에 주요한 역할을 하는 요인이 되었다. 즉 과거에는 개인 개발자의 취미활동으로 공개SW를 개발하고 활용하였으나, 현재의 공개SW는 기업 개발자가 많이 참여하는 산업표준 기술, 제품화를 위한 기술, 기술 검증 방법으로 활용되고 있다.

공개SW와 클라우드 컴퓨팅 두 기술이 출현한 시기는 다르지만, 클라우드가 빠르게 확산되는 데에 있어 공개SW의 영향도 크다고 생각된다. 클라우드의 핵심기술인 가상화 기술이 오픈 프로젝트로 진행되어, 리눅스 공식버전(mainline kernel)에 추가되었기 때문이다. Xen 기술은 2007년 11월, KVM(Kernel-based Virtual Machine) 기술은 2007년 9월에 리눅스 커널에 포함되어, 공개SW기반의 클라우드 컴퓨팅을 구축할 수 있도록 되었다. 이외에도 클라우드 모니터링, 사용자 인증, 대용량 데이터 관리 등 클라우드 컴퓨팅 분야의 다양한 기술들이 공개되어 활용이 가능한 상황이다.

라이선스 측면에서도 공개SW는 클라우드와의 활용과 부합된다. 기존의 시스템SW 라이선스 정책은 일반적으로 사용자의 수나 구축된 시스템의 규격을

기반으로 결정되었으나, 클라우드 환경은 이러한 라이선스 규칙을 적용하는데 다소 애매한 부분이 있기 때문이다. 그러나 무엇보다도 그 동안의 HW 기술의 발전, 네트워크 기술의 발전과 유틸리티 컴퓨팅 등 컴퓨터 및 SW 기술의 지속적인 발전이 오늘의 클라우드 컴퓨팅을 탄생시켰다.

가트너에 따르면 2010년까지 글로벌 2천 대 기업의 75%에서 사용됐던 공개SW가 2016년에는 99%까지 확대될 것으로 전망한다. 특히 클라우드 컴퓨팅 환경 구축에 공개SW를 활용하는 추세로, 구글, 아마존, IBM 등에서도 이미 공개SW를 활용하고 있다. 또한 클라우드기반 애플리케이션 프레임워크 구축에도 공개SW가 활용되고 있다. 구글 App Engine과 NASA의 Nebula는 공개SW기반의 웹 애플리케이션 프레임워크인 django를 기반으로 서비스를 제공한다. 아마존 EC2(Elastic Computing Cloud)도 django, jboss, rails 등의 공개SW를 사용한다. 따라서 전 세계 많은 기업들은 빠르게 시장의 요구에 대응하고 제품을 업그레이드하기 위해 공개SW 기술을 활용하고 있다.

구글, 아마존, KT, SKC&C, 삼성 등 국내외 민간 기업의 클라우드 기술 확보뿐만 아니라, 각국의 정부 및 공공기관은 클라우드 컴퓨팅 도입과 국가 차원의 육성 전략을 수립하고 있다. 미국의 경우, 그린 IT 실현이라는 목표로 공공업무의 클라우드화를 추진하고 있다. 영국은 공개SW기반의 정부 클라우드를 구축하고, 정부 앱스토어를 통한 자유로운 정보 공유를 유도한다. 일본은 자국 기업으로 전자정부 및 전자자치체 클라우드를 2015년까지 구축하여 새로운 행정 개혁을 추진하고 있다. 중국의 경우, SW 산업 경쟁력 강화를 위해 우시, 북경, 청도, 상해 등 10개 이상의 도시에 클라우드기반 전자정부, 게임, 번역, 전자자료 등의 서비스를 구축하는 시범사업을 2011년에 추

진하였다. 이외, 대만, 러시아, 말레이시아, 프랑스, 덴마크 등 많은 국가에서 클라우드 컴퓨팅을 도입하거나 계획을 추진하고 있다.

따라서 본 고에서는 클라우드 컴퓨팅 확산에 있어 주요한 역할을 하는 공개SW 기술 현황을 살펴본다. 특히 클라우드 인프라스트럭처 서비스(IaaS: Infrastructure as a Service) 단계의 기술과 상호 호환성에 대한 공개SW 기술을 살펴보고자 한다.

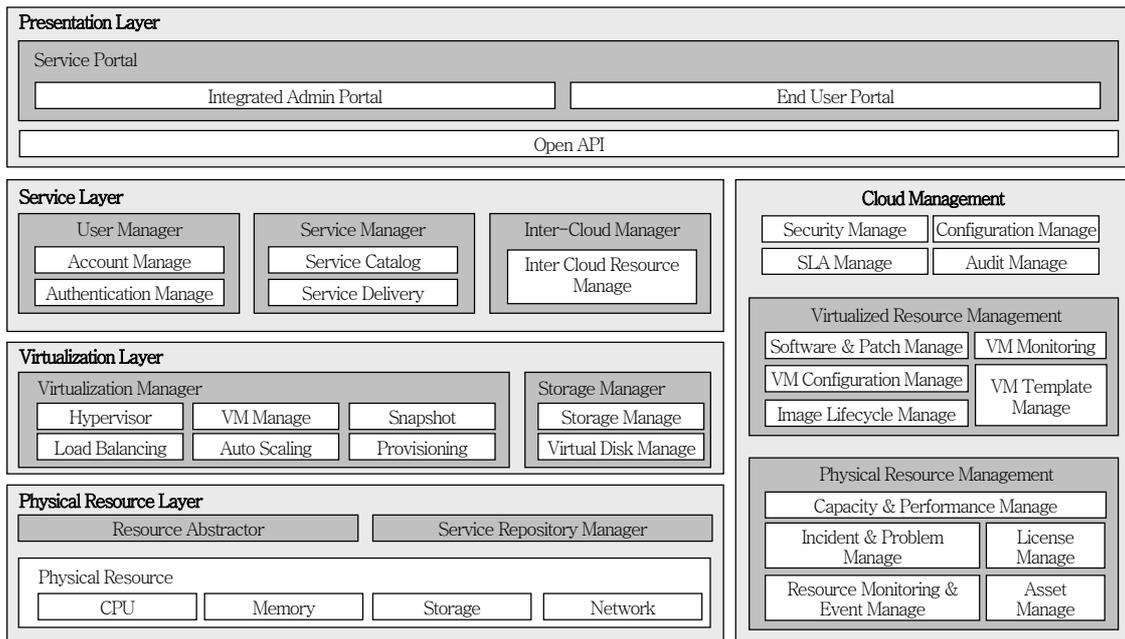
## II. 클라우드 IaaS 구조와 공개SW

### 1. 구조 및 기능

클라우드 IaaS 서비스는 가상의 서버 자원을 제공하고 사용한 만큼 비용을 청구하는 서비스이다. 사용자는 IaaS 서비스를 이용하여 회사 시스템을 운영하고 업무환경을 구축할 수도 있다. 사용할 가상 서버를 만들고 부팅하는데 몇 분 정도의 매우 짧은 시간이 소요되며, 또한 필요에 따라 컴퓨팅 용량을 늘리

거나 줄여야 하는 경우에도 신속하게 사용 용량을 조절할 수 있다. 서비스가 종료되면 할당을 받은 자원은 반환되고 종료된다. 이 모든 서비스는 가상화 기술이라는 핵심기술이 존재하기 때문이다. 이렇게 가상자원으로 제공받을 수 있는 서비스는 CPU, 메모리 등을 활용한 컴퓨팅 서비스, 데이터를 보관하고 관리할 수 있는 스토리지 서비스와 분산 응용 SW 간 통신을 위한 메시지 큐 서비스 등이 존재한다. 이러한 서비스를 제공하는 대표적인 기업은 아마존, 구글, MS, IBM, Rackspace, Niravanix 등 여러 서비스 사업자가 있다.

(그림 1)은 IaaS 서비스를 제공하기 위한 클라우드 스택의 개념적인 구조도이며, 크게 4개 부분으로 구성되어 있다. 첫째, 가상화 층으로 가장 핵심적인 부분이며, 물리적 자원을 가상화하고 가상머신의 동작을 제어, 관리하는 부분이다. 둘째, 사용자 인증, 요금 등 IaaS 서비스의 제반 사항을 관리하는 서비스 관리 층이다. 그리고 셋째 클라우드 관리 층은 물리



(그림 1) 클라우드 인프라 서비스(IaaS) 스택 구조

적인 자원뿐만 아니라, 가상자원 관리를 기반으로 클라우드 인프라를 제어하고 관리한다. 마지막으로 IaaS 서비스의 창구인 인터페이스 층이다. 이 인터페이스 층은 서비스 사업자, 표준단체별로 차이가 존재하여 표준화된 인터페이스가 존재하지 않는 상황이다. 따라서 다중 클라우드 간 IaaS 자원을 공유하고 연계하는 클라우드 연동 및 호환을 위해, 표준화 문제를 해결하는 오픈 프로젝트가 있다. 이 프로젝트의 소개는 IV장에서 한다.

IaaS 서비스 스택을 구성하기 위한 필수적인 기능들은 다음과 같다.

- 가상머신모니터(하이퍼바이저) 기술: 서버 자원을 가상화
- 가상머신모니터 기능 제어 기술: 가상머신모니터 기능을 사용하기 위한 여러 인터페이스 제공
- 스토리지 가상화 기술: 스토리지 가상화 및 대용량 스토리지 구축
- 가상머신 이미지 관리 기술: 가상머신 등록, 저장 관리 및 검색
- 가상머신 시스템 제어 기술: 가상 자원으로 IaaS 클라우드를 구축하고 관리
- 물리시스템 제어 기술: 물리 자원을 제어, 관리
- 관리 자동화 기술: 대규모 IaaS 클라우드 시스템을 관리
- 사용자 인증 기술: 서비스 가입자 관리, 접근 제어
- 서비스 품질 관리 기술: 사용자 서비스의 품질 관리 및 문제 해결
- 미터링 기술: 자원 사용량에 따른 과금
- 가상머신 보안 기술: 가상머신모니터의 보안

기능을 세분화하는 배경은 공개SW로 IaaS 클라우드를 구축할 때 어떤 종류의 공개SW가 있는지를 파악하기 위해서이다. 또 다른 이유는 여러 공개SW

들의 기능, 특징 등이 상이하어, IaaS 클라우드 플랫폼의 기본기능을 정립하여 이들 공개SW의 기능과 수준을 비교, 분석하기 위해서이다. 마지막 배경은 IaaS 클라우드 구축을 위한 클라우드 OE(Operating Environment)를 정립하고자 한다. 기존의 운영체제는 물리적 자원을 제어, 관리한다면, 클라우드 OE는 가상머신을 제어 관리하기 때문에 운영체제와 유사한 개념으로 정리할 수 있다.

분류된 각 세부 기능별 공개SW 현황을 다음 절에서 소개한다.

## 2. 기능별 공개SW

본 절에서는 IaaS 클라우드 플랫폼의 기본 기능에 대한 공개SW를 나열한다. <표 1>은 IaaS 클라우드 플랫폼의 기본 기능에 대한 공개SW 종류이다.

먼저, 가상머신모니터 기술 경우, 3가지의 공개SW가 있다. Xen은 많은 기업에 의해서 개발되어, 개

<표 1> IaaS 클라우드 구축에 활용될 수 있는 공개SW 현황

기본 기능	종류
가상머신모니터 기술	Xen, KVM, VirtualBox
가상머신모니터 기능 제어 기술	Libvirt, Ovirt, virt-manager
스토리지 가상화 기술	LVM, Hadoop, Gfarm, Swift
가상머신 이미지 관리 기술	Glance, Aeolus
가상머신 시스템 제어 기술	OpenStack, CloudStack, Eucalyptus, Nimbus, OpenNebula, ConVirt
물리시스템 제어 기술	Groundwork, Zabbix, Nagios, HInemos, Ganglia
관리 자동화 기술	Chef, Crowbar
사용자 인증 기술	OpenSSO(Open-source Single Sign-On), Higgins, Shibboleth
서비스 품질 관리 기술	Zabbix, Ganglia 등 이용 가능하나, 서비스 문제 해결 기능 부족
미터링 기술	Ovirt 등 이용 가능하나, 자원 모니터링만 가능하고 과금 기능 부족
가상머신 보안 기술	-

발의 공개성도 높고, 안정성도 높다. 기술의 로드맵 등 기술 추가 방침 등도 명확하다. 최근 5년간의 버그 대응률은 69%이다. KVM은 레드햇의 참여와 리눅스 커널버전에 포함되어 안정된 개발이 진행되고 있다. 버그 대응률은 지난 5년간 65%로 기록되고 있다. VirtualBox의 개발 주체는 오라클이며, 개발 로드맵이 공개되어 있지 않다. 1일 다운로드 수가 2만 건을 기록하고 있어 인기가 높다고 할 수 있으며, 총 다운로드 수는 600만 건이다. 버그 대응률도 다른 공개SW 기술과 유사한 64%이다. 가상머신 라이브 이전 등 세부 기능면에서도 세 가지 SW는 유사하다.

가상머신모니터 기능제어 기술 경우, 기능면에서 세 가지 모두 유사하다. oVirt는 주로 KVM 전용, Virt-manager는 주로 KVM, Xen 등에 사용된다. 하지만 HW 감시 기능 또는 스토리지 기능 등에서는 XenServer와 같은 상용SW보다 기능이 떨어진다. 스토리지 가상화 경우, 기능면에서 큰 차이가 없지만, 이용 상황에 따라 큰 차이가 생길 가능성이 있다.

가상머신 이미지 관리 기술은 가상머신 생성 관리 측면의 템플릿 기능과 배치 기능, 이미지 저장공간 효율성을 위한 중복제거 기능으로 세분화할 수 있다. 그러나 이러한 기능을 만족하는 단독적인 공개SW 프로젝트는 존재하지 않는 것으로 보이며, 다른 공개SW 프로젝트에 포함되어 기술개발이 진행되고 있다. 물리시스템 제어 기술은 이미 익숙한 시스템 관리 기술이다. 따라서 많은 공개SW가 존재하지만, Zabbix와 Nagios가 개발의 안정성, 지원성 면에서 다른 공개SW보다 우수하다. Tivoli 등 상용SW와 기본적인 기능에서는 동등한 기능을 가지고 있다. 다만, HW 감시, 보안 관리 등의 부가적인 기능에서는 차이를 보인다. 또한 가상머신 감시 기능은 유상으로만 사용할 수 있다.

관리 자동화 기술은 대규모 클라우드 구축 시, 관

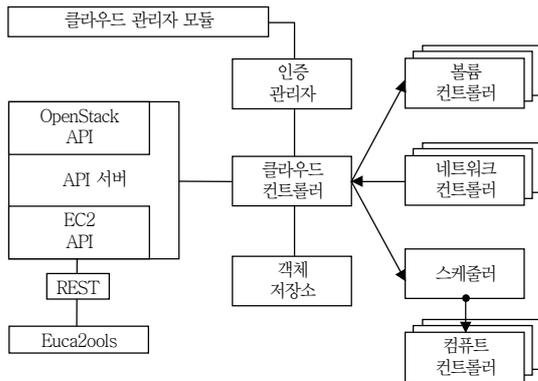
리의 편의성과 자원 운영의 효율성을 고려하는 세부 기능이 요구된다. 즉 클라우드의 부하를 고려한 자동적인 자원 할당과 분배 기능은 필수적이다, 관련되는 공개SW 활동은 활발하지 않는 것으로 판단된다. 사용자 인증 기술 경우 OpenSSO(Open-source Single Sign-On)가 안정성, 기능성, 지원성 등 모든 면에서 다른 공개SW보다 우수하다. 서비스 품질 관리 및 미터링 기술 경우는 단독적인 공개SW가 존재하지 않는다. 즉 서비스 품질 관리와 과금을 위한 자원 사용량 모니터링 기능은 다른 공개SW에 존재하지만, 서비스 품질을 해결, 관리하고 사용량에 따른 과금 기능에 대한 공개SW는 없는 것으로 판단된다.

마지막으로 IaaS 클라우드 구축에 대표 역할을 하는 가상머신 시스템 제어 기술이며, OpenStack, Eucalyptus 등이 해당된다. 앞 절에서 설명했듯이 IaaS 클라우드 스택의 세부 기능을 분류하였지만, 가상머신 시스템 제어 기술이 다른 기술들을 결합하여 클라우드 스택 기능을 제공한다. 그러나, 가상머신 보안 등 앞 절에서 세분화한 기능 모두를 잘 결합하여 제공하지는 못하는 것 같다. 따라서, OpenStack 등의 기술 현황은 다음 장에서 자세히 다룬다.

### III. 클라우드 운영관리 기술

#### 1. OpenStack Nova

OpenStack은 RackSpace와 NASA가 연합하여 만든 오픈 프로젝트이며, IaaS 환경을 구축하고 관리하는 Nova, 클라우드 스토리지 서비스 환경을 구축하고 관리하는 Swift, 가상머신 이미지를 등록하고 관리하는 Glance로 나뉜다. Glance는 아마존의 S3로 구축한 스토리지 클라우드도 서비스 구축이 가능하다[2]. 본 고에서는 IaaS 환경을 관리하는 Nova에



(그림 2) Nova 구조도

대해 중점적으로 기술한다.

Nova는 private, public, hybrid 방식을 모두 지원하며, 파이썬(python)으로 구현되어 있다[2]. Nova는 (그림 2)와 같이 인증 관리자, API(Application Program Interface) 서버, 객체 저장소 및 가상머신을 관리하는 5개의 컴포넌트로 구성되어있다.

인증 관리자는 접속하는 사용자의 공인키와 비밀번호를 확인하고 해당 사용자의 접속 여부와 접근 권한을 결정한다. API 서버는 사용자가 사용할 수 있는 인터페이스를 제공하고 사용자로부터 들어온 요청을 클라우드 컨트롤러로 전송하는 역할을 한다.

가상머신을 관리하는 5개의 컴포넌트는 가상머신 이미지를 저장하고 관리하는 객체 저장소, 가상머신이 사용할 스토리지 공간을 할당하고 관리하는 볼륨 컨트롤러, 가상머신에 IP를 할당하고 프로젝트 단위로 가상머신 네트워크를 연결하는 네트워크 컨트롤러, 가상머신을 생성하고 관리하는 컴퓨터 컨트롤러 및 가상머신이 저장될 물리적 서버를 선택하는 스케줄러로 구성된다. 이 중 볼륨 컨트롤러, 네트워크 컨트롤러, 컴퓨터 컨트롤러는 복수개의 서버 운용이 가능하며 컴퓨터 컨트롤러는 클라우드 컨트롤러와 직접 통신하지 않고, 스케줄러를 통해 데이터를 주고받는다. 이들 컨트롤러는 Xen, KVM 등 가상머신모

니터 기술과 연계되어 동작한다.

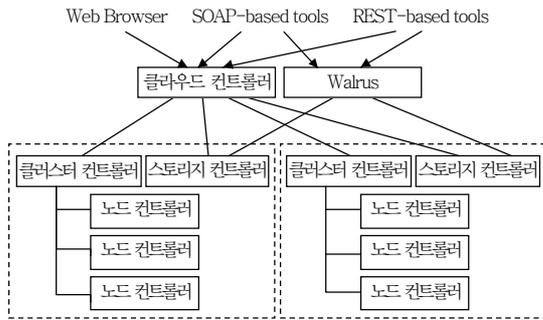
Nova에서 사용자는 전체 관리자, 프로젝트 관리자, 네트워크 관리자, IT 보안 관리자, 개발자의 다섯 가지 역할로 구분되며 기능에 대한 접근 권한이 세부적으로 나누어진다. 또한 사용자의 역할에 따라 보안 그룹으로 사용자를 묶어서 사용자의 접근 권한을 더 쉽게 관리할 수도 있다.

사용자는 REST(Representational State Transfer)기반의 통신을 이용하여 가상머신과 서비스를 관리할 수 있으며 아마존의 EC2 API와 OpenStack에서 개발한 API의 사용이 가능하다. EC2 API를 사용하는 이유는 기존에 다수의 사용자를 보유하고 있는 EC2의 API를 제공함으로써 운용상의 확장성 및 범용성을 지원한다. 추가적으로 Nova는 사용편의성을 제공하기 위하여 CUI(Cummand User Interface)뿐만 아니라 웹 개발 프레임워크인 django를 이용한 웹 GUI(Graphic User Interface) 툴도 제공한다.

## 2. Eucalyptus

Eucalyptus는 Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems의 약자로 private, public, hybrid 방식을 지원하는 클라우드 환경 구축을 위한 공개SW이다. 캘리포니아 산타바바라 대학에서 연구 프로젝트로 시작되었으며 Eucalyptus Inc.에서 상업화한 후에도 오픈 프로젝트로 유지, 개발되고 있다[3].

Eucalyptus는 (그림 3)과 같이 클라우드 컨트롤러, 클러스터 컨트롤러, 노드 컨트롤러, Walrus, 스토리지 컨트롤러로 구성되어 있다. 클라우드 컨트롤러는 전체 시스템과 사용자의 접근을 관리하는 컨트롤러이다. 클라우드에 접근하려면 클라우드 컨트롤러를 거쳐야 하며, 사용자는 웹이나 SOAP(Simple Ob-



(그림 3) Eucalyptus 구조도

ject Access Protocol), REST 기반의 API를 이용하여 클라우드 컨트롤러와 통신할 수 있다.

클러스터 컨트롤러는 Eucalyptus 클라우드를 구성하기 위해 구축한 클러스터를 관리한다. Eucalyptus는 다수의 클러스터로 구축할 수 있으며 각 클러스터별로 클러스터 컨트롤러를 두어 클러스터를 구성하는 서버를 관리한다. 따라서 클러스터 컨트롤러는 클러스터에 연결되어 있는 노드 컨트롤러에 관한 모든 정보를 유지하며, 클러스터 레벨의 스케줄링과 네트워크를 제어한다.

노드 컨트롤러는 각 서버에 동작하면서 서버를 관리하는 컨트롤러이다. 이것은 클러스터를 구성하는 서버의 호스트 운영체제와 하이퍼바이저를 제어하고 클러스터 컨트롤러의 요청에 따라 가상머신을 실행할 서버를 선택하고 실행한다.

Walrus는 bucket 단위의 저장소를 구현한 것으로 아마존의 S3와 호환하는 SOAP와 REST 인터페이스를 통하여 사용할 수 있다.

스토리지 컨트롤러는 아마존의 EBS 방식을 이용한 블록 기반 스토리지이다. 아마존의 S3 인터페이스를 제공하고, 가상머신 이미지를 저장하고 접근하는데 사용된다. 이렇게 아마존의 S3를 제공함으로써 사용자의 편의성을 제공할 수 있으며 더 안정적인 서비스를 제공할 수 있다.

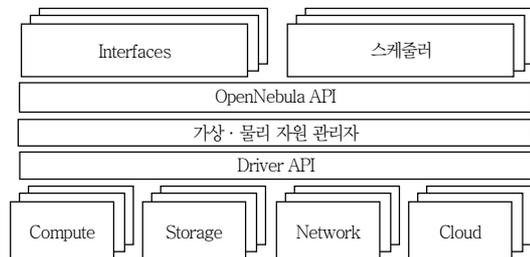
Eucalyptus는 현재 Xen과 KVM을 이용하여 구축할 수 있으며 VMWare도 지원할 수 있도록 개발 중이다. 호스트 운영체제는 CentOS, openSUSE, Debian, Fedora를 지원한다.

사용자는 Eucalyptus 클라우드를 관리하는 관리자와 생성된 가상머신을 관리하는 사용자로 나뉜다. 이들은 Eucalyptus에서 제공하는 Eucal2ools를 이용한 CUI와 웹API를 이용한 GUI 툴을 이용하여 클라우드 컨트롤러에 접근하여 관리한다. 그러나 GUI 툴은 CUI에 비해 사용할 수 있는 기능이 제한적이다.

### 3. OpenNebula

OpenNebula는 IaaS 클라우드 시스템을 구축하기 위한 공개SW로 Universidad Complutense de Madrid의 DSA-research Group에서 처음 개발하였으며 private, public, hybrid 형태의 클라우드 구축이 가능하다[4].

OpenNebula는 (그림 4)와 같이 4개의 관리모듈로 구성되어 있다. 가상머신을 실행하고 관리하는 컴퓨트 모듈, 가상머신의 이미지와 가상머신이 사용할 저장공간을 관리하는 스토리지 모듈, 가상머신의 네트워크를 관리하는 네트워크 모듈, 외부 클라우드와 연결하여 관리하는 클라우드 모듈이 있다. 또한 중간 계층에 있는 가상·물리 자원 관리자는 사용자의 요청을 받아서 전달하는 요청 관리자와 가상머신 관리자, 호스트 관리자, 가상네트워크 관리자가 있어 자



(그림 4) OpenNebula 구조도

원을 효과적으로 사용할 수 있게 한다.

사용자와 클라우드 관리자는 OpenNebula에서 제공하는 인터페이스를 이용하여 클라우드를 사용하고 관리할 수 있다. 이 때 요청이 들어온 작업은 자원 정책과 작업 부하에 따라 스케줄러에 의해 처리된다[5].

OpenNebula에서는 가상머신을 띄울 때 가상머신의 속성을 나열한 가상머신 정의 파일(VMDF: Virtual Machine Definition File)을 함께 생성해야 한다. 그러나 사용자의 요구에 모두 적합하는 파일을 만드는 것은 어렵기 때문에, DNS나 사용자의 SSH key 등 대표적인 속성은 가상머신을 시작할 때 수정할 수 있다[4].

사용자에 대한 기준은 Eucalyptus처럼 OpenNebula 클라우드를 관리하는 관리자와 생성된 가상머신을 사용하고 가상머신에 할당된 네트워크를 관리하는 사용자로 나누며, 사용자에게는 클라우드 제어를 위한 CUI가 제공된다[5]. 또한 아마존의 EC2, OGF(Open Grid Forum)의 OCCI(Open Cloud Computing Interface), vCloud 인터페이스를 모두 지원하여 사용자의 편의성을 높였다[6].

#### 4. CloudStack

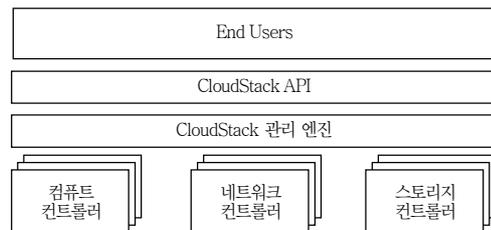
CloudStack은 private, public, hybrid 방식을 지원하는 공개 SW로 Citrix와 Cloud.com에서 개발하였다[4].

CloudStack은 (그림 5)와 같이 3개의 컨트롤러와 사용자를 위한 API, 그리고 중간계층에 관리 엔진을 두어 사용자의 요청에 따라 자원을 효율적으로 관리할 수 있도록 하였다[7].

CloudStack은 사용자를 관리자, 도메인 관리자, 모니터링 관리자, 사용자로 나눈다. 관리자는 전체 클라우드를 관리하는 권한을 가지고, 도메인 관리자는 클라우드에 가상머신을 생성하고 관리하는 등 클라우드 서비스를 관리하는 권한을 가진다. 모니터링 관리자는 가상머신 관리 권한은 없지만 서비스 환경을 모니터링 할 수 있다. 사용자는 도메인 관리자가 제공하는 가상머신 서비스를 이용할 수 있다.

제공하는 API도 사용자의 권한에 따라 클라우드 관리자, 업체별 관리자, 사용자로 나누고 있다. 클라우드 관리자는 클라우드 관리를 위한 모든 API를 사용할 수 있고, 업체별 관리자는 가상머신 서비스와 관련된 API만 사용할 수 있다. 사용자는 가상머신 사용과 관련된 API만 사용할 수 있다.

CloudStack은 AJAX(Asynchronous JavaScript and XML) 기반의 GUI뿐만 아니라, 웹 인터페이스



(그림 5) CloudStack 구조도

〈표 2〉 클라우드 운영관리 SW 비교

	OpenStack	Eucalyptus	OpenNebula	CloudStack
운영체제	Ubuntu, CentOS	Ubuntu, Debian, CentOS, RHEL, Fedora, OpenSUSE	Ubuntu, Debian, OpenSUSE, CentOS, RHEL, Mac OS	RHEL, CentOS, Ubuntu, Fedora
Hypervisor	KVM, Xen 지원	KVM, Xen 지원 VMWare 지원 예정	KVM, Xen, VMWare 지원	KVM, Xen 지원
사용자 요청 방식	REST	SOAP, REST	CUI	REST
개발 언어	Python	C, Java	C/C++, Ruby 등	Java

와 CUI, REST 방식의 통신을 통해 사용자의 요청을 받아서 내부 컴포넌트로 전달한다.

<표 2>는 소개한 4가지 클라우드 운영관리 공개 SW의 특징을 정리한 것이다.

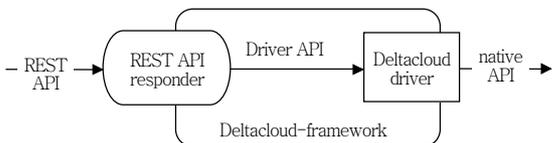
#### IV. 다중 클라우드 호환 기술

IaaS 클라우드 서비스 사업자는 서비스를 위해 자신의 인터페이스(unique API)를 제공하며, 사용자는 이러한 인터페이스를 사용하여 클라우드 응용을 개발하고 사용한다. 하지만 서로 다른 클라우드 인터페이스로, 사용자는 매번 클라우드 사업자에 맞추어 사용방식이나 응용을 변경해야 하는 문제점이 있다. 따라서 이러한 문제점을 해결하기 위한 공개SW를 본장에서 소개한다.

##### 1. Deltacloud

Deltacloud는 클라우드 사업자 간 인터페이스 비호환성의 문제점을 해결하고자 하는 공개SW이다[8]. Deltacloud는 REST 기반의 API를 제공(Deltacloud API)하며, Deltacloud API를 이용하여 개발자는 클라우드 응용을 개발하고, 사용자는 클라우드 사업자의 서비스를 이용할 수 있다. 즉, Deltacloud API 기반의 응용은 클라우드에서 수정없이 사용될 수 있다. Deltacloud 프레임워크는 (그림 6)과 같이 클라우드 사업자의 API (native API)와 Deltacloud API(REST API)를 연결하는 드라이버를 제공한다.

현재 제공되는 드라이버는 EC2 드라이버, Eucalyptus 드라이버, RHEV-M 드라이버, Rackspace 드라이버, OpenNebula 드라이버, IBM Smartcloud 드라이버, RimuHosting 등이다. Deltacloud 프레임워크는 웹서버 방식으로 운용된다. 즉 사용자는 직접 Deltacloud API를 자신의 프로그램에서 사용하는 것이 아니라, Deltacloud API를 지원하는 웹 서버를 이용한다. 루비(ruby)로 구현되어 있으며, Deltcloud 프레임워크를 설치(# gem install Deltacloud-core) 후 웹 서버 구동 시 사용할 클라우드 사업자의 드라이버를 지정(\$ Deltacloud -I sbc)할 수 있다. 또한 다른 클라우드를 사용하기 위해서 Deltacloud 프레임워크 동작 중에도 동적으로 드라이버를 변경(http://localhost:3001/api; driver=ec2; providers=eu-west-1)할 수 있다. Deltacloud API는 HTTP authentication 방식으로 사용자을 인증하며, 별도의 로그인 과정을 거치지 않는다. Deltacloud API는 클라우드 자원을 표현하기 위해서 HW 사양(hardware profiles), 클라우드 위치(realms), 클라우드 이미지(images)와 실제 클라우드 시스템의 현황(instances) API를 제공한다. 이들 API로 사용자는 사용할 클라우드의 세부 사양과 클라우드가 있는 위치, 제공되는 SW 환경 등의 동작사용을 확인할 수 있다. 게다가, instance의 <actions> 기능을 통해 클라우드를 재부팅, 시작 또는 중지시킬 수 있다. 이들 Deltacloud API 기능은 해당 클라우드 사업자의 드라이버를 통해 각 클라우드 native API로 전달되어 동작한다. 따라서, 드라이버에서도 Deltacloud API처럼, hardware\_profiles(), images(), realms(), instances(), create\_instance(), reboot\_instance(), stop\_instance(), destroy\_instance() 등을 제공한다. 현재 Deltacloud 프레임워크는 루비 언어로 제공되지만, 다른 언어도 곧 지원될 예정이다.



(그림 6) Deltacloud의 동작 원리

## 2. Simple Cloud

Simple Cloud API는 PHP에 클라우드 기술을 추가하여 파일 저장소(file storage service), 문서 저장소(document storage service), 그리고 큐 서비스(queue service)를 제공할 수 있도록 한다. Simple cloud API를 지원하는 주요 사업자는 IBM, MS, Rackspace, Nirvanix, GoGrid이며, 클라우드 간 호환성이 있는 클라우드 응용을 구현할 수 있다[9].

참고적으로 Simple Cloud API는 Zend 프레임워크에서 제공된다. Zend 프레임워크는 PHP 개발사인 Zend에서 제공하는 오픈 소스 프레임워크이며, 프레임워크에서 제공하는 컴포넌트 기능으로 웹 응용 서비스를 아주 쉽게 개발할 수 있다. 구글, MS 및 StrikeIron 등 협력기업들이 Zend 프레임워크에 다양한 컴포넌트 기능을 제공하고 있다.

파일 저장소 경우, 제공되는 인터페이스는 Zend\_Cloud\_StorageService\_Adapter로 각 클라우드 사

<표 3> Simplecloud 파일 저장소 인터페이스

세부 인터페이스	기능
fetchItem	해당 위치의 저장소로부터 자료를 가져오기
storeItem	자료를 해당 위치에 저장하기
deleteItem	해당 위치로부터 자료를 제거하기
copyItem	해당 위치 간에 자료를 복사하기
moveItem	해당 위치로 자료를 옮기기
renameItem	해당 위치의 자료이름을 변경하기
listItem	해당 디렉터리 내용을 보여주기
fetchMetadata	해당 위치의 메타데이터 가져오기
storeMetadata	해당 위치에 메타데이터를 저장하기

<표 4> Simplecloud를 지원하는 클라우드 사업자

서비스	지원하는 클라우드 벤더
파일 저장소 API	아마존 S3, 윈도우 Azure Blob 스토리지, Nirvanix
문서 저장소 API	아마존 simpleDB, 윈도우 Azure 테이블 스토리지
큐 서비스 API	아마존 SQS, 윈도우 Azure 큐 스토리지

업자별 인터페이스, 즉 윈도우 Azure 경우, Zend\_Cloud\_StorageService\_Adapter\_WindowAzure가 각각 제공된다. 그리고 <표 3>, <표 4>와 같은 파일 저장소의 기능에 대한 세부 인터페이스를 제공한다.

## 3. Libcloud

Libcloud도 다른 클라우드 서비스 사업자와 상호 동작할 수 있는 방법을 제공한다. 클라우드 사업자 간 호환성을 제공하는 다른 기술과 같이 개발자의 편의성을 제공하는 기술이다[10]. Libcloud는 Cloudkick사에 의해 시작되었으며, 현재 아파치 라이선스 2.0하에 오픈 소스 프로젝트로 성장하고 있다. 참고적으로 Cloudkick사는 고객의 가상 시스템을 관리하고 모니터링 서비스를 하는 전문기업으로 백만대 규모의 고객 시스템을 모니터링하고 있다. 그리고 2010년 12월 Rackspace로 인수되었다.

Libcloud 인터페이스는 클라우드 사업자의 서로 다른 API들을 추상화(abstraction)하여 단일 인터페이스의 파이썬(python) 라이브러리로 제공한다. 세계의 종류의 클라우드 자원에 대한 인터페이스를 제공하는데, 클라우드 서버, 클라우드 스토리지, 부하 분산기이다. 클라우드 서버 경우, 현재 22개의 서비스 사업자를 지원한다. 제공되는 Libcloud의 API는, 클라우드 서버 경우, Deltacloud처럼 클라우드 HW 사양, 위치, 클라우드 이미지 정보 등을 기반으로 노드를 생성, 제거 및 부팅 등의 기능을 제공한다. 관련하여 사용자의 인증 등에 대한 인터페이스도 제공된다. 클라우드 스토리지 경우, API는 파일을 저장할 공간(container)를 생성하거나 제거, 생성된 저장공간에 파일을 생성, 제거 또는 가져오기 등 Simple Cloud와 유사한 기능을 제공하지만, 다른 점은 저장 공간(container)을 생성 관리할 수 있다는 점이다.

〈표 5〉 다중 클라우드 호환 기술 비교

기술	사용언어	지원되는 서비스	지원되는 클라우드
Delta cloud	루비(ruby)	클라우드 서버, 클라우드 스토리지	구글, Linode, Rackspace, EC2, Terremark, Zerigo, Brightbox, Blue Nox Group, AWS
Simple cloud	PHP	파일 저장소	아마존 S3, 윈도우 Azure Blob 스토리지, Nirvanix
		문서 저장소	아마존 simpleDB, 윈도우 Azure 테이블 스토리지
		큐 서비스	아마존 SQS, 윈도우 Azure 큐 스토리지
libcloud	파이선	클라우드 서버, 클라우드 스토리지, 부하분산 서비스	EC2, Gandi, GoGrid, OpenNebula, Rackspace, RimuHosting, Slicehost, VoxCloud, vCloud, CloudSigma 등 22개

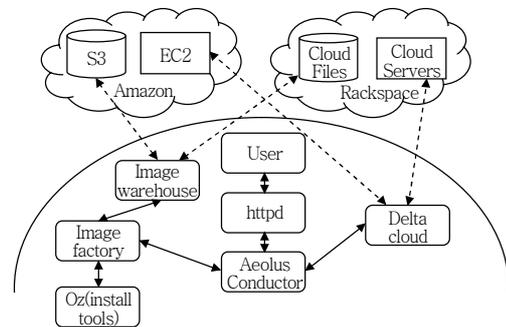
Deltacloud, Simple cloud과는 달리, Libcloud는 부하 분산기의 기능을 제공한다. 부하 분산기를 생성, 제거하거나, 생성된 서버를 부하 분산기에 연결, 제거하는 기능 등을 제공한다.

〈표 5〉에서는 Deltacloud, Simple Cloud, Libcloud를 비교하는 표이다.

#### 4. Aeolus

Aeolus는 가상머신이 수행될 private 클라우드 또는 public 클라우드 등 적절한 클라우드를 선택하는 클라우드 브로커 기술이다[11]. 다중 클라우드 환경이기 때문에 서로 다른 클라우드 간 인터페이스 문제는 Deltacloud 기술을 이용한다. 참고적으로 초기의 Deltacloud 프로젝트는 클라우드 공통 API 기술과 클라우드 간 브로커 기술로 구성되었지만, 두 기술은 분리되어 다른 프로젝트로, Deltacloud 프로젝트와 Aeolus 프로젝트, 운영되고 있다. 따라서 Aeolus에서는 Deltacloud 프로젝트 결과물인 REST 기반 클라우드 API 기술을 활용하고 있다. Aeolus 기술의 활용분야로는 임의의 클라우드 장애 대응, 클라우드 서비스 사업자간의 가격 경쟁 환경, 지리적인 문제로 인한 인접 클라우드 자원 사용, 클라우드 서비스의 부하 분산 등의 다중 클라우드 환경에서 활용될 수 있다.

Aeolus 기술은 레드햇 RHEV(Red Hat Enter-



(그림 7) Aeolus 운영 환경

prise Virtualization), Eucalyptus, VMware vSphere 가상화 기술과 연계되어 동작되며, 지원되는 public 클라우드는 현재 아마존 EC2와 Rackspace이다. Aeolus 기술은 리눅스 RHEL6(Red Hat Enterprise Linux 6), 페도라 14와 윈도우 2008에서 동작한다. (그림 7)은 Aeolus 기술의 구성요소와 다중 클라우드 환경에서 가상머신을 어떻게 배치, 관리하는지를 보여준다.

사용자의 가상머신 이미지는 클라우드 이미지 빌더인 imagefactory와 OS 설치 기능을 가진 oz 구성 요소에 의해서 생성이 되며, image warehouse는 생성된 가상머신을 각 클라우드 서비스 사업자에 맞게 수정하여 복제, 관리 등을 수행한다. 즉 사용자는 하나의 가상머신을 생성하였지만, aeolus의 이미지 관리 기능에 의해 다중의 클라우드 사업자별로 가상머신이 생성되어 존재하게 된다. 가상머신 생성 후, 사용자가 가상머신을 활성화하면, Aeolus conductor

는 다중 클라우드의 모니터링 데이터를 기반으로 사용자의 요청사항에 적절한 클라우드 사업자를 선택하고 가상머신을 활성화시켜 서비스가 제공되도록 한다. 사용자는 Deltacloud API를 사용하여 가상머신을 제어하기 때문에, 가상머신이 어느 클라우드에서 동작되더라도 가상머신의 제어, 관리의 문제가 없다. 게다가 사용 중에 클라우드 장애가 발생되면 다른 클라우드로 전환도 가능하다. 하지만 실시간 가상머신 이동을 지원하지 않으면 서비스 연속성을 보장할 수 없다.

● 용 어 해 설 ●

가상머신: 물리서버 위에 생성하는 가상의 컴퓨터  
하이퍼바이저: 물리서버 위에 가상머신을 만들기 위한 소프트웨어.  
Private Cloud: 사내 등 제한된 조직에 클라우드 서비스를 제공하는 방식  
Public Cloud: 웹 서비스를 통해 불특정 사용자에게 클라우드 서비스를 제공하는 방식  
Hybrid Cloud: 두 개 이상의 클라우드를 결합한 방식

약어 정리

AJAX	Asynchronous JavaScript and XML
API	Application Program Interface
CUI	Cummand User Interface
EC2	Elastic Computing Cloud
GUI	Graphic User Interface
IaaS	Infrastructure as a Service
KVM	Kernel-based Virtual Machine
OCCI	Open Cloud Computing Interface
OE	Operating Environment
OGF	Open Grid Forum
OpenSSO	Open-source Single Sign-On

REST	Representational State Transfer
RHEL6	Red Hat Enterprise Linux 6
RHEV	Red Hat Enterprise Virtualization
SOAP	Simple Object Access Protocol
VMDF	Virtual Machine Definition File

참고 문헌

- [1] 민옥기 외, “훤히 보이는 클라우드 컴퓨팅.” 서울, 전자신문사, 2009. 10.
- [2] OpenStack LLC, OpenStack Compute Administration Manual. <http://docs.openstack.org>, June 8, 2011.
- [3] Eucalyptus Systems Inc., Eucalyptus EE 2.0 Administrator Guide, 2010. <http://www.eucalyptus.com/resources/documentation>
- [4] F. Gomez-Folgar et al., “An Open-source Cloud Management Platform Comparison,” *Cloud comput. Edu. Study Dev.*, Russia, 2011.
- [5] OpenNebula Project Leads, OpenNebula 2.2 Documentation, Mar. 2011. <http://opennebula.org/documentation:documentation>
- [6] Ignacio M. Llorente, The OpenNebula Standard-based Open-source Toolkit Build Cloud Infrastructures, 2009. <http://www.rediris.es/jt/jt2009/ponencias/Clausura1.pdf>
- [7] Cloud.com Inc., CloudStack Introduction, 2011. [http://www.cloud.com/index.php?option=com\\_k2&view=item&layout=item&id=114&Itemid=346](http://www.cloud.com/index.php?option=com_k2&view=item&layout=item&id=114&Itemid=346)
- [8] Deltacloud Core, Deltacloud Documentation. <http://incubator.apache.org/deltacloud/documentation.html>
- [9] Zend Technologies, Simple Cloud API. <http://simplecloudapi.org/>
- [10] Apache Software Foundation, Apache Libcloud. <http://libcloud.apache.org/>
- [11] RedHat. Inc., About Aeolus. <http://www.aeolus-project.org/>