

# 클라우드 컴퓨팅의 기술 동향과 가상화 기반 관리 기술\*

## (Technology Trends and Virtualization based Management Techniques for Cloud Computing)

심영철  
홍익대학교 정보컴퓨터공학부  
[shim@cs.hongik.ac.kr](mailto:shim@cs.hongik.ac.kr)

### 요약

최근 클라우드 컴퓨팅에 대한 관심이 높아지고 관련 서비스와 제품이 많이 출시되고 있으나 이의 정확한 정의, 특성 및 서비스에 대해서는 많은 혼란이 있는 상황이다. 본 논문에서는 먼저 클라우드 컴퓨팅에 대한 소개를 한 후 이의 중요한 기반 기술이면서 많이 혼동이 되고 있는 그리드 컴퓨팅과 여러 기술적 관점에서 비교 설명한다. 다음 클라우드 컴퓨팅 시스템의 운용 및 관리에 가장 중요한 기술 중의 하나인 가상화에 대해 설명하고 분산 시스템의 관리를 위한 표준화 기관인 DMTF의 CIM과 가상화 표준을 사용하여 클라우드 컴퓨팅 시스템을 관리하는 기술에 대해 기술한다.

**Keywords:** Cloud Computing, Virtualization, Cloud Management

## 1. 서론

최근 클라우드 컴퓨팅에 대한 관심이 높아짐에 따라 많은 클라우드 컴퓨팅 서비스와 제품들이 나오고 있으며 각 기관에서의 표준화 활동이 시작되고 있는 상황이다. 그러나 클라우드 컴퓨팅은 그리드 컴퓨팅, 유틸리티 컴퓨팅, SaaS(Software as a Service) 등의 많은 컴퓨팅 기술을 바탕으로 발전된 기술이므로 클라우드 컴퓨팅의 정의와 특징 및 서비스 등에 대해 많은 혼란이 있으며 특히 그리드 컴퓨팅과는 더욱 많은 혼란이 있는 상황이다. 그러므로 본 논문에서는 먼저 클라우드 컴퓨팅의 정의, 특징, 서비스, 아키텍처 등에 대해 설명하고 다음 그리드 컴퓨팅과의 차이점에 대해 여러 가지 기술적인 관점에서 비교 설명하고자 한다.

클라우드 컴퓨팅의 운용 및 관리에 가장 기본이 되는 기술 중의 하나가 가상화(Virtualization)이다. 이질적인 분산시스템의 관리를 용이하게 하기 위한 표준을 개발하는 기관 중의 하나가 DMTF인데 이 기관에서는 관리 대상 자원의 객체 지향 모델링을 위하여 CIM(Common Information Modeling) 스키마를 개발하고 또 이를 바탕으로 하여 가상화 표준을 개발하고 있다. 본 논문에서는 먼저 가상화의 정의 및 종류에 대해 설명하고 다음 DMTF의 CIM과 가상화 표준을 사용하여 클라우드 컴퓨팅 시스템을 관리하는 방안에 대해 기술한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 클라우드 컴퓨팅의 정의와 기술동향을 포함한 소개를 하고 3장에서 여러 가지 기술적인 측면에서 그리드 컴퓨팅과 비교 설명한다. 4장에서는 먼저 가상화에 대해 설명하고 DMTF의 CIM과 가상화를 사용하여 클라우드 컴퓨팅 시스템을 관리하는 방안에 대해 기술한 후 5장의 결론으로 맺는다.

## 2. 클라우드 컴퓨팅의 소개

본 장에서는 클라우드 컴퓨팅의 정의와 특징, 서비스, 발전과정, 아키텍처에 대해 설명하고 클라우드 컴퓨팅의 국내외 기술 동향과 표준화 동향에 대해 기술한다.

---

\* 본 연구는 지식경제부의 정보통신표준기술력향상사업의 일환으로 수행하였음.

[2009-P1-21-08J21, 차세대 컴퓨팅 분산 환경 이기중 시스템 자원 관리 표준개발]

## 2.1. 클라우드 컴퓨팅의 정의, 특징과 서비스

클라우드 컴퓨팅(Cloud Computing)은 인터넷을 통하여 제공되는 어플리케이션들과 이 서비스들을 제공하는 데이터센터 내의 모든 하드웨어와 소프트웨어를 포함한다. 이 데이터센터 내의 하드웨어와 소프트웨어를 합쳐서 클라우드(Cloud)라 부른다. 가트너 자료에 의하면 클라우드 컴퓨팅은 ‘인터넷 기술을 활용하여 다수의 고객들에게 높은 수준의 확장성을 가진 IT 자원들을 서비스로 제공하는 컴퓨팅이다’ 라고 정의되어 있다. 즉, 클라우드 컴퓨팅 환경에서 사용자들은 인터넷이 연결된 단말을 통해 대용량의 컴퓨터 집합에 접속하여 어플리케이션, 스토리지, OS, 보안 등 필요한 IT 자원을 원하는 시점에 필요로 하는 만큼 골라서 사용하게 되며, 사용량에 기반하여 대가를 지불한다. 클라우드 컴퓨팅은 SaaS(Software as a Service)와 유틸리티 컴퓨팅(Utility Computing)의 합으로 생각할 수 있으며 그림 1은 이러한 개념 하에서 클라우드 컴퓨팅 내 구성체층의 사용자 또는 제공자로서의 사람들의 역할을 보여주고 있다[1].

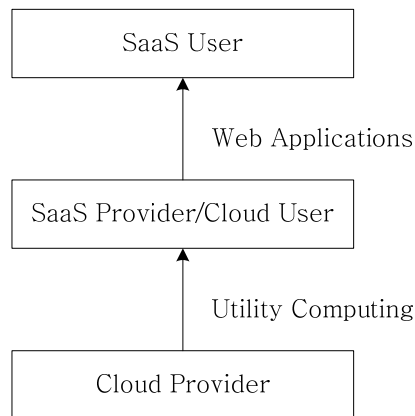


그림 1. 클라우드컴퓨팅의 사용자와 제공자

클라우드 컴퓨팅에는 다음과 같은 장점이 있다.

- 사용자가 자신의 필요에 따라 무한정의 컴퓨팅 자원을 사용할 수 있다는 환상(Illusion)을 제공한다. 그러므로 사용자는 하드웨어와 소프트웨어 시스템을 제공하는 계획을 미리 세울 필요가 없다.
- 사용자는 작은 시스템으로부터 시작할 수 있고 시스템 자원에 대한 요구가 증가함에 따라 시스템 자원을 증가시키면 된다.
- 필요에 따라 짧은 시간을 단위로 (예를 들어 프로세서를 시간 당 또는 스토리지를 날짜 당) 사용하고 비용을 지불하면 되고 필요가 사라지면 자원을 더 사용하지 않을 수 있다.

클라우드는 가상화된 컴퓨팅 자원들의 풀(Pool)로서 다음을 행할 수 있다.

- 배치(Batch) 유형의 백엔드 잡이나 대화식 잡과 같은 다양한 종류의 워크로드(Workload)를 수행할 수 있다.
- 가상 기계나 물리적 기계를 신속히 제공함으로써 워크로드가 신속히 배치되고 또 크기가 변동될 수 있도록 한다.
- 중복되어 있으며 스스로 복구가 가능하고 확장성이 높은 프로그래밍 모델을 제공함으로써 워크로드가 피할 수 없는 많은 하드웨어/소프트웨어 고장으로부터 복구될 수 있도록 한다.
- 컴퓨팅 자원을 실시간으로 모니터링함으로써 필요에 따라 자원의 할당을 재구성할 수 있다.

클라우드는 크게 공용 클라우드(Public Cloud)와 개인 클라우드(Private Cloud)로 나눌 수 있다. 공용 클라우드는 일반 사용자들이 사용량에 따라 사용료를 지불할 수 있도록 하는 클라우드이며 판매되는 서비스는 유틸리티 컴퓨팅이다. 개인 클라우드는 특정 기관의 내부 데이터센터로서 일반 사용자들의 접근이 허용되지 않는다.

클라우드 컴퓨팅 시스템은 다음과 같이 크게 세 가지 다른 레벨에서의 서비스를 제공한다[2].

- Infrastructure as a Service (IaaS)

- Platform as a Service (PaaS)
- Software as a Service (SaaS)

IaaS는 하드웨어와 소프트웨어, 장비 등을 제공하여 사용자가 이를 사용하여 사용량에 따라 요금을 지불하면서 소프트웨어를 작동시킬 수 있도록 하는 환경을 제공한다. 기반구조는 실행되는 어플리케이션이 필요로 하는 자원의 요구량에 따라 확장 또는 축소될 수 있다. 이러한 서비스의 대표적인 예로는 Amazon EC2(Elastic Cloud Computing)과 S3(Simple Storage Service)가 있는데 이들 시스템은 일반 사용자가 계산과 저장의 기반구조를 사용하고 사용량에 따라 요금을 낼 수 있도록 한다[3].

PaaS는 사용자가 자신이 원하는 기능의 어플리케이션을 구축, 테스트, 설치할 수 있도록 하는 높은 수준의 통합 환경을 제공한다. 일반적으로 개발자는 개발할 수 있는 소프트웨어의 유형에 제한을 받게 된다. 이의 예에는 Google App Engine이 있는데 이 시스템은 확장 및 축소가 가능한 시스템 위에서 웹 어플리케이션을 구축할 수 있도록 한다.

SaaS는 사용자가 인터넷을 통하여 특정한 소프트웨어를 사용하고 이 소프트웨어의 사용량에 따라 요금을 낼 수 있도록 하는 서비스이다. 이의 예로는 Amazon Flexible Payments Service와 Google Maps 등이 있다.

## 2.2. 클라우드 컴퓨팅의 발전

그림 2는 지난 20여년 동안 클라우드컴퓨팅이 발전되어 온 과정을 보여주고 있다[4]. 클라우드컴퓨팅으로의 트렌드는 1980년대 말 그리드컴퓨팅의 개념으로부터 시작되었다. 그리드컴퓨팅은 여러 개의 컴퓨터를 병렬로 사용하여 특정한 문제를 풀거나 특정한 어플리케이션을 수행하는 것을 목표로 하였다. 보통 그리드는 서버 클러스터로 구성되며 이 위에서 매우 큰 태스크가 여러 개의 작은 부태스크들로 나뉘어져 병렬로 수행되었다. 반면 클라우드 컴퓨팅은 컴퓨팅 자원을 포함하는 많은 자원을 사용하여 사용자에게 서비스를 제공하는 것을 목표로 한다. 다시 설명하면 클라우드 컴퓨팅에서는 서버, 스토리지, 네트워크, 어플리케이션, 프로세스 등과 같은 컴퓨팅 자원과 비즈니스 자원을 포함하는 기반구조에서 선택하고 동적으로 구성한 후 워크로드가 이 위에서 수행될 수 있도록 한다. 클라우드에서는 그리드 환경뿐 아니라 비 그리드 환경(예를 들어 Web 2.0 어플리케이션을 수행하는 3 계층의 웹 아키텍처)도 지원할 수 있다.

Grid Computing	Utility Computing	Software as a Service	Cloud Computing
<ul style="list-style-type: none"> <li>- Solving large problems with parallel computing</li> <li>- Made mainstream by Globus Alliance</li> </ul>	<ul style="list-style-type: none"> <li>- Offering computing resources as a metered service</li> <li>- Introduced in late 1990s</li> </ul>	<ul style="list-style-type: none"> <li>- Network-based subscriptions to applications</li> <li>- Gained momentum in 2001</li> </ul>	<ul style="list-style-type: none"> <li>- Next generation internet computing</li> <li>- Next generation data centers</li> </ul>

그림 2. 클라우드 컴퓨팅의 발전

1990년대에 가상화의 개념이 가상 서버 개념으로부터 시작하여 가상 어플리케이션과 같은 개념으로까지 발전하였다. 유틸리티 컴퓨팅은 클러스터를 컴퓨팅을 위한 가상 플랫폼으로 제공하며 계량화된 비즈니스 모델을 제공하였다. 최근 SaaS에서는 가상화의 개념을 어플리케이션으로까지 확장시켜서 사용한 자원량에 따라 대금을 청구하는 것이 아니라 사용자에게 제공된 어플리케이션의 가치에 따라 대금을 청구하는 것이 가능하게 되었다.

클라우드컴퓨팅의 개념은 그리드, 유틸리티, SaaS 개념들로부터 발전하였다. 클라우드컴퓨팅에서 사용자는 언제, 어디서나 자신의 어플리케이션에 접근할 수 있다. 이 어플리케이션들은 확장성이 매우 높은 데이터센터 내에 위치하고 이 데이터센터는 동적으로 제공될 수 있으며 많은 사용자들에 의해 공유될 수 있다. 바로 이 동적 관리가 가능한 기반구조가 클라우드 컴퓨팅의 장점이다.

## 2.3. 클라우드 아키텍처

클라우드 아키텍처의 정의에는 여러 버전이 있을 수 있으나 그림 3은 4 계층으로 구성된 구조를 보여주고 있다[2]. 최하 계층인 패브릭(Fabric) 계층은 계산 자원, 저장 자원, 네트워크 등의 기본 자원을 포함한다. 통합된 자원(Unified Resource) 계층은 가상화 등을 통하여 추상화되고 캡슐화된 자원을 포함하며 상위 계층이나 종단 사용자에게 가상 컴퓨터/클러스터, 논리적 파일 시스템, 데이터베이스 시스템 등과 같은 통합된 자원을 제공한다. 플랫폼(Platform) 계층은 특화된 도구, 미들웨어 서비스 등을 통합된 자원 위에 추가하여 개발 및 배치 플랫폼을 제공한다. 이의 예로는 웹 호스팅 환경, 스케줄링 서비스 등이 있다. 마지막으로 어플리케이션 계층은 클라우드 내에서 수행되는 어플리케이션들을 포함한다.

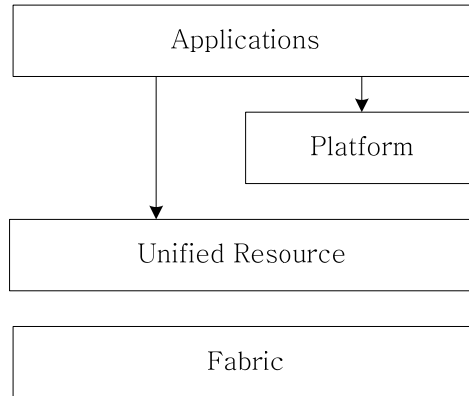


그림 3. 클라우드의 4 계층 아키텍처

그림 4는 앞의 4 계층 구조에 기반하여 IBM에서 정의된 동적 기반구조(Dynamic Infrastructure)를 지원하는 데이터센터의 아키텍처 프레임워크를 보여주고 있으며 이 그림에서의 각 계층은 그림 3에서의 계층과 동일하다[4]. 최하 계층의 물리적 하드웨어들은 가상화되어 유연하고 적응성이 높은 플랫폼을 제공함으로써 자원의 이용률은 높일 수 있다. 동적 기반구조에서 가장 중요한 요소는 중간에 있는 가상화 계층과 관리계층이다. 이 두 계층은 데이터센터 내의 자원들이 효율적으로 관리되어 신속하게 제공/배치/구성될 수 있도록 한다. 그리고 동적 기반구조는 그림에서 보는 바와 같이 다양한 종류의 워크로드를 동시에 처리할 수 있도록 한다.

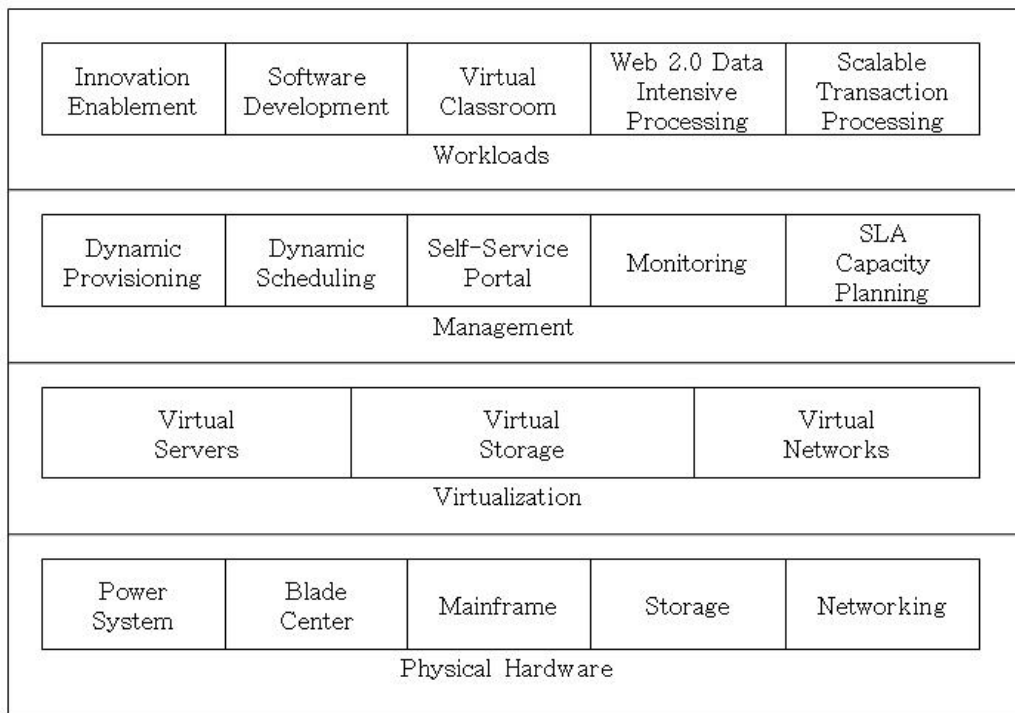


그림 4. 동적 기반구조를 지원하는 IBM 데이터센터의 아키텍처 프레임워크

## 2.4. 클라우드 컴퓨팅의 국내외 기술 동향

현재 Amazon, Google, Microsoft, IBM, 3Tera 등과 같은 많은 회사들이 클라우드 서비스를 제공하고 있다.

Amazon Web Service(AWS)는 Amazon.com이 인터넷을 통하여 제공하는 원격 컴퓨팅 서비스들의 집합이다. 2002년 시작된 이 서비스는 말단 사용자에게 직접 제공되지 않고 클라우드컴퓨팅 서비스 개발자들이 사용할 수 있는 기능들을 제공한다. 현재 330,000의 개발자들이 AWS를 사용하고 있다고 한다. AWS는 HTTP, REST와 SOAP 프로토콜을 사용하여 접근할 수 있으며 사용량에 따라 비용이 청구된다. AWS가 제공하는 서비스 중에는 다음이 있다[3].

- Amazon Elastic Compute Cloud (EC2): Zen을 사용하여 확장성이 있고 가상화된 개인 서버를 제공
- Amazon Elastic Block Store (EBS): EC2에 지속성이 있는 블록 레벨의 스토리지 볼륨을 제공
- Amazon Simple Storage Service(S3): 어플리케이션에 웹 기반 스토리지를 제공

Google App Engine은 Google의 기반구조 위에 웹 어플리케이션을 구축 및 실행할 수 있도록 하는 플랫폼으로 2008년 4월에 베타 버전이 출시되었다. Google App Engine은 많은 수의 서버들을 사용하여 어플리케이션을 수행하거나 데이터를 저장할 수 있도록 하며 사용자의 요청을 안정적으로 처리할 수 있도록 서버의 수를 자동적으로 조정할 수 있도록 한다[5].

Microsoft의 Azure Service Platform은 어플리케이션들이 Microsoft 데이터센터에서 수행될 수 있도록 하는 어플리케이션 플랫폼이다. Azure는 Windows Azure라고 불리는 클라우드 운영체제를 제공하는데 이 운영체제는 어플리케이션을 위한 런타임 환경을 제공하고 어플리케이션을 개발, 관리, 실행할 수 있도록 하는 서비스를 제공한다[6].

이외에도 IBM은 Linux, Hadoop, Xen과 PowerVM에 기반한 서비스인 Blue Cloud를 발표하였고 RightScale과 3Tera는 여러 개의 다른 데이터센터들의 기계들을 어려움 없이 사용할 수 있도록 하는 관리도구를 제공하고 있다[7-9].

Rackspace사는 Cloud Sites, Cloud Files, Cloud Servers 를 포함하는 Mosso라는 이름의 클라우드컴퓨팅 플랫폼을 발표하였고 ServePath사는 GoGrid라는 이름의 유사한 플랫폼을 발표하였다[10,11].

국내에는 최근 클라우드컴퓨팅산업포럼, 한국클라우드컴퓨팅연구조합, 한국클라우드서비스협의회 등과 같은 많은 단체들이 조직되어 활동을 시작하였지만 아직 초기 수준에 머물러 있고 많은 단체들의 기능과 참여업체들이 중복되어 있어서 아직은 혼란한 상태이다.

## 2.5. 클라우드 컴퓨팅의 국내외 표준화 동향

2008년 중반 미국의 몇 개 대학 연구소들을 중심으로 새로 설립된 OCC(Open Cloud Consortium)는 클라우드컴퓨팅의 공개 표준에 관하여 연구하고 있다[12]. OCC의 활동 영역은 다음과 같다.

- 클라우드컴퓨팅의 표준 개발과 클라우드 사이의 상호운용성을 위한 프레임워크 개발을 지원
- 클라우드컴퓨팅을 위한 벤치마크를 개발
- 클라우드컴퓨팅을 위한 참조 구현(Reference Implementation)을 지원
- Open Cloud Testbed라 불리는 클라우드컴퓨팅을 위한 테스트베드를 관리
- 클라우드컴퓨팅에 관련된 워크샵과 다른 행사들을 후원

OCC는 현재 다음과 같은 네 개의 워킹그룹으로 구성되어 있다.

- **대용량 데이터 클라우드를 위한 표준과 상호운용성에 관한 워킹그룹:** Google 기술 보고서에 의해 많이 알려진 클라우드의 구조는 분산 파일 시스템을 제공하는 저장(Storage) 클라우드와 MapReduce를 지원하는 계산(Compute) 클라우드와 테이블 서비스를 지원하는 데이터 클라우드로 구성된다. 개방형 소스 Hadoop 시스템은 이 구조를 따르고 있다. 이 워킹그룹은 이러한 유형의 클라우드에 대용량 데이터 클라우드(Large Data Cloud)라는 이름을 사용한다. 이 워킹그룹의 목표는 대용량 데이터 클라우드들 사이의 상호운용을 위한 표준을 개발하는 것이다. 현재 Hadoop, Thrift, Sector와 Pig 사이의 상호운용을 평가하고 있으며 (Malstone이라고 불리우는) 사용자 요구 기반 계산 용량을 제공하는 클라우드를 위한 벤치마크를 개발하고 있다.

- **표준 클라우드 성능 측정과 평가 시스템에 관한 워킹그룹:** 다양한 클라우드 컴퓨팅 제공자들의 성능, 보안, 품질을 비교할 수 있는 용이한 방법을 제공할 수 있도록 하기 위하여 사용 케이스(Use Case)를 다듬고, 요구사항을 수집하여 벤치마크를 개발하는 것을 목표로 한다. 이 워킹그룹은 CCIF(Cloud Computing Interoperability Forum)과 협조 관계에 있다.
- **클라우드 사이의 정보 공유와 보안에 관련된 워킹그룹:** 클라우드 사이의 정보 공유를 위한 표준과 표준 기반 구조에 초점을 두고 있다. 특히 다른 기관에 속하여 다른 권한과 정책을 가진 클라우드 사이의 정보 공유에 관심을 가진다. 이 그룹은 보안 구조에도 관심을 가지고 있다.
- **개방형 클라우드 테스트베드 워킹그룹:** 개방형 클라우드 테스트베드를 관리하고 운영한다. 개방형 클라우드 테스트베드는 네트워크 연결을 위하여 Cisco C-Wave와 UIC Teraflow Network를 사용한다. 이 두 네트워크는 National Lambda Rail에 의해 제공되는 파장을 사용한다.

현재 OCC 참여기관은 다음과 같다.

- Member: Aerospace, CISCO, MIT Lincoln Labs, Northwestern University , Open Data Group, Sector Project, University of Illinois at Chicago , Yahoo
- Contributing Members: Calit2, Johns Hopkins University, National Lambda Rail, University of Chicago,

OGF(Open Grid Forum)이 중 IaaS로서의 클라우드에 인터페이스할 수 있는 현실적인 솔루션을 제공하는 것을 목표로 하여 OCCI-WG(**Open Cloud Computing Interface WG**)을 구성하였다[13]. OCCI-WG은 클라우드컴퓨팅 기반구조의 원격 관리를 위한 API 명세를 개발하는 것을 목표로 한다. 원격관리 기능에는 배치, 자율적 스케일링(Autonomic Scaling), 모니터링 등의 공통 태스크들을 포함된다. 현재 OCCI-WG은 다음의 API들을 고려하여 연구를 진행하고 있다.

- Amazon EC2 API
- ElasticHosts API
- FlexiScale API
- GoGrid API
- Sun Cloud API

DMTF(Distributed Management Task Force) 개방형 클라우드 표준 인큐베이터(Open Cloud Standards Incubator)를 구성하고 엔터프라이즈컴퓨팅과 클라우드컴퓨팅의 상호운용이 가능한 관리 기능을 통합할 수 있도록 하는 명세들을 개발하고 한다[14]. 이를 위하여 CIM(Common Information Model), OVF(Open Virtualization Format), WBEM 프로토콜과 같은 기존의 DMTF 표준들의 확장이 필요할 수 있다고 보고 있다. DMTF 활동의 범위는 IaaS로서의 클라우드의 자원 관리에 중점을 두고 있으나 SLA, QoS, 이용률(Utilization), 제공(Provisioning), 회계(Accounting)와 과금(Billing)과 같은 PaaS 영역의 이슈들에도 관심을 보이고 있다. 이 인큐베이터에서는 다음과 같은 결과물들을 예상하고 있다. 결과물 뒤의 내용은 결과물이 나올 것이라고 예상되는 일시이다.

- 클라우드 분류 방법: 3Q09
- 클라우드 상호운용성 white paper: 3Q09
- 정보의 명세(Informational Specifications)
  - 클라우드에 사용하기 위해 필요한 OVF의 변경: 1Q10
  - 클라우드 자원 관리를 위한 프로파일들: 4Q09
  - 기타 다른 DMTF 표준의 변경: 1Q10
- 클라우드 자원 관리를 위해 필요한 신뢰(Trust)에 대한 요구사항: 3Q09

최근 국내에서는 TTA가 차세대 서버컴퓨팅 관련 표준화 대상항목에 시스템 자원 가상화와 클라우드컴퓨팅의 내용을 포함하여 발표하였다. 시스템 자원 가상화에 관련된 표준은 DMTF의 표준을 기초로 하여 ETRI와 공동 연구를 진행하고 있으며 클라우드컴퓨팅에 관련해서는 ‘모바일 클라우드 서비스를 위한 그리드 기술 표준’과 ‘클라우드 플랫폼 인터페이스 표준’ 등과 같은 표준을 개발하려 하고 있다.

### 3. 클라우드 컴퓨팅과 그리드 컴퓨팅의 비교 분석

그리드 컴퓨팅은 동적이고 여러 기관으로 구성된 가상 조직에서 자원 공유와 잘 조정된 문제 해결을 가능하도록 하는 것을 목표로 한다. Ian Foster는 무엇이 그리드이고 무엇이 그리드가 아닌지를 정의하는데 도움이 되는 3가지 중요한 점검 사항을 제안하였다[15].

- 중앙 집중식으로 제어되지 않는 자원들을 조정
- 표준화되고, 개방형이며, 다목적의 프로토콜과 인터페이스를 사용
- 간단하지 않은 서비스 품질을 제공

클라우드와 그리드는 컴퓨터를 한 기관이 전부 구매하고 운용하는 개념에서 제 삼자에 의해 운용되는 개념으로 전환시킴으로써 컴퓨팅 비용을 감소시키고, 신뢰성을 증가시키며, 유연성을 증가시킨다는 공동 목표를 가지고 있으므로 이 둘은 상이점들도 있지만 공통점도 많이 가지고 있다. 그림 5는 클라우드와 일부 중복되는 다른 도메인들 사이의 관계를 보여주고 있다. 웹 2.0은 서비스 지향적 어플리케이션의 스펙트럼을 전부 포함하고 있으며 이 안에서 클라우드 컴퓨팅은 스케일이 큰 쪽을 차지하고 있다. 슈퍼컴퓨터와 클러스터 컴퓨팅은 비서비스 어플리케이션 부분에 초점을 맞추고 있다. 그리드 컴퓨팅은 모든 분야와 겹치는데 슈퍼컴퓨터나 클라우드보다는 스케일 면에서 작은 편이다[15].

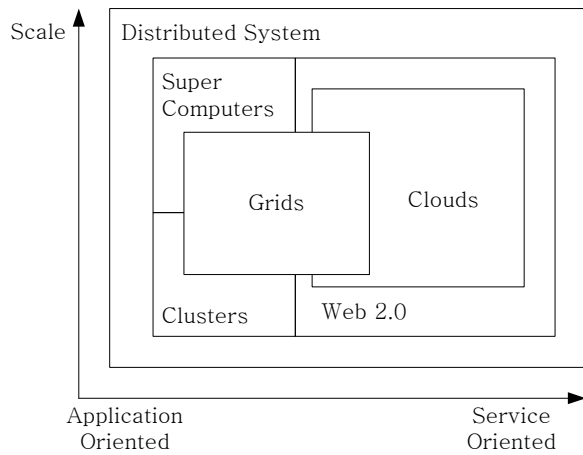


그림 5. 그리드와 클라우드의 개괄

클라우드와 그리드는 공통 목표를 지향하고 있지만 구체적인 내용과 사용되는 기술 면에서는 많은 차이점이 있으므로 본 장에서는 이 둘 사이의 차이점에 대해 비즈니스 모델, 아키텍처, 계산 모델, 가상화, 어플리케이션 모델과 보안 모델 등의 측면에서 설명한다[2].

**비즈니스 모델:** 클라우드 기반 비즈니스 모델에서는 사용자가 자원 사용량에 기반하여 제공자에게 비용을 지불하며 규모의 경제성(Economy of Scale)에 기반하여 사용자의 경비를 낮추고 제공자의 이윤을 증가시키는 것을 목표로 한다. 예를 들어 Amazon의 EC2는 사용하는 자원 유형 별로 사용 시간 당 경비를 부과하며 S3는 사용된 저장공간의 GB-Month 단위로 부과한다.

그리드의 비즈니스 모델은 프로젝트 지향적으로 프로젝트를 제안한 사용자는 특정한 서비스 단위(예를 들어 CPU 시간)를 할당 받는다. 예를 들어 TeraGrid에서는 사용하고자 하는 계산 용량을 증가시키기 위해서 이에 상응하게 복잡한 제안서를 제출하여야 한다. 사용자들이 그리드 커뮤니티에 가입하는 동기는 자신의 기관에 속한 자원을 다른 기관들이 사용할 수 있도록 허락하는 대신 다른 기관의 자원을 할 수 있게 된다는 것이다.

**아키텍처:** 그리드는 지역적으로 분산되어 있는 여러 기관들의 계산, 저장, 네트워크 자원들을 연합하여 대형의 계산 문제를 푸는 것을 목표로 하였기 때문에 구성 자원들이 이질적이고 동적일 수밖에 없었다. 그러므로 그리드에서는 기존의 자원들의 하드웨어, 운영체제, 지역적 자원 관리, 보안 기반구조를 그대로 가지고 이들 자원들을 통합하는 것에 초점을 맞추었고 상호운용성과 보안이 일차적으로 중요한 문제가 되었다. 그리드 프로토콜 아키텍처는 그림 6과 같이 5 계층으로 구성된다. 패브릭 계층에서는 계산, 저장, 네트워크, 코드 저장소와 같은 다른 자원들에 대한 접근을 제공한다. 연결(Connectivity) 계층은 핵심 통신과 인증 프로토콜을 정의하며 자원(Resource) 계층은 개별 자원에 대한 공유 오퍼레이션에 대한 광고, 발견, 협상, 모니터링, 과금과 지불 등을 위한 프로토콜을 정의한다. 공유(Collective) 계층은 자원들 모임

사이의 상호운용성을 지원하고 어플리케이션 계층은 앞의 계층들 위에 구축되는 모든 사용자 어플리케이션을 포함한다. 그리드의 구조의 각 계층에서는 그리드를 위한 특정 프로토콜들이 개발되어 사용되고 있다.

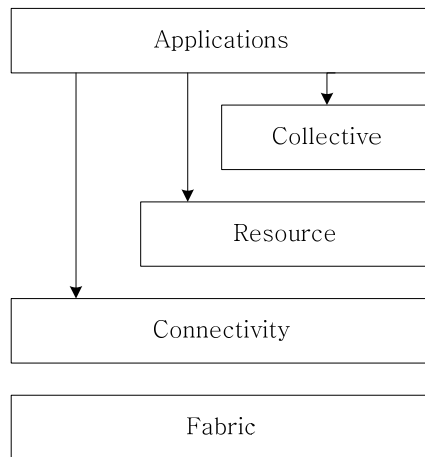


그림 6. 그리드 프로토콜 아키텍처

클라우드는 보통 표준 프로토콜들에 의해 접근될 수 있는 계산과 저장 장치들의 대형 풀을 의미한다. 클라우드는 기존에 존재하는 프로토콜들 위에 구축되며 그리드 기술 위에 클라우드를 구현하는 것도 가능하다. 클라우드의 아키텍처는 그림 3에서 살펴 본 바와 같이 기본적으로 4계층으로 구성되어 있다. 클라우드는 IaaS, PaaS, SaaS와 같은 여러 종류의 서비스를 제공하고 있지만 이들 서비스들에 대한 인터페이스 표준은 아직 마련되어 있지 않다. 그러므로 현재의 클라우드들 사이에는 상호운용성 문제가 존재한다. 클라우드가 더욱 발전하고 여러 개의 클라우드를 필요로 하는 더욱 복잡한 어플리케이션이나 서비스가 나타나게 되면 상호운용성 문제를 해결할 수 있도록 하는 표준 인터페이스를 채택하는 것에 대한 요구가 커질 것이다.

**계산 모델:** 대부분의 그리드는 배치 스케줄링 기반 계산 모델을 사용한다. 그리드에서는 PBS나 Condor와 같은 지역적 자원 관리자가 그리드 사이트의 계산 자원을 관리하고 사용자는 배치 잡을 제출하여 자원들을 일정 시간 동안 사용하고자 한다는 요구를 한다. 대부분의 그리드는 대화식(Interactive) 어플리케이션을 지원하지 못한다.

클라우드 컴퓨팅 계산 모델은 클라우드 내의 자원들이 모든 사용자들에 의해 동시에 공유는 모델을 제공한다. 그러므로 클라우드 상에서는 지연에 매우 민감한 어플리케이션을 수행하는 것이 수월하다.

**가상화:** 가상화는 거의 모든 클라우드를 위한 필수불가결한 요소가 되었다. 클라우드는 여러 개의 사용자 어플리케이션을 수행하여야 하며 사용자에게는 모든 어플리케이션이 동시에 수행되고 있고 또 클라우드 내의 가용 자원을 모두 사용할 수 있는 것과 같이 보여져야 한다. 가상화는 추상화(Abstraction)를 통하여 기반의 패브릭이 자원의 풀(Pool)로 통합 되어지고 이 위에 자원 오버레이(예를 들어 데이터 저장 서비스나 웹 호스팅 환경)가 구축될 수 있도록 한다. 또 각 어플리케이션이 캡슐화(Encapsulation)될 수 있도록 함으로써 어플리케이션들이 구성, 배치, 시작, 이동, 일시 정지, 재개, 종료될 수 있도록 하며 더 나은 보안, 관리용이성, 고립성(Isolation)을 제공될 수 있도록 한다.

반면 그리드는 클라우드와 같은 정도로 가상화에 의존하고 있지는 않는데 이것은 각 기관들이 자신들의 자원에 대한 완전한 제어를 유지하고 있기 때문이다. 그러나 그리드에서도 가상화를 사용하기 위하여 많은 노력이 기울여지고 있다.

**어플리케이션 모델:** 그리드는 고성능 컴퓨팅(High Performance Computing: HPC)으로부터 고처리율 컴퓨팅(High Throughput Computing: HTC)에 이르는 다양한 종류의 어플리케이션을 지원한다. HPC 어플리케이션은 지연 시간이 매우 낮은 인터컨넥션 네트워크(Interconnection Network)를 가진 기계 상에서 밀결합(Tightly Coupled) 병렬 잡으로 효율적으로 처리되므로 광대역 네트워크 그리드 상에서는 수행되지 않는다. 이러한 어플리케이션들은 보통 MPI(Message Passing Interface)를 사용한다. 그리고 그리드는 소결합(Loosely Coupled) 어플리케이션들을 수행하는데도 성공적으로 활용되고 있는데 이 어플리케이션들은 워크플로우 시스템(Workflow System) 등을 사용하여 관리되고 수행된다.

원칙적으로 클라우드 컴퓨팅도 같은 종류의 어플리케이션을 수행할 수는 있으나 매우 낮은 네트워크 지연 시간을 필요로 하는 HPC 어플리케이션에는 적절하지 않다. 클라우드 컴퓨팅은 일반적으로 소결합 어플리케이션이나, 트랜잭션 지향 어플리케이션, 그리고 대화식 어플리케이션 등에 적절하다.

**보안 모델:** 자원들이 이질적이고 동적이며 각 그리드 사이트는 자신의 관리 도메인을 가지고 운용 자치권



을 가진다는 가정 위에 그리드가 구축된다. 그러므로 보안은 그리드에서 매우 중요한 문제가 되고 그리드 기반 구조에 포함되어 설계된다. 중요한 이슈는 SSO(Single Sign On)으로서 사용자가 한번만 로그인함으로써 여러 그리드 사이트를 접근할 수 있도록 하는 것이다. 이를 위하여 사용하는 메커니즘에는 크리덴셜 위임(Credential Delegation)이 포함된다.

대부분의 클라우드는 한 기관에 속하는 데이터 센터들로 구성되므로 데이터 센터 내의 하드웨어와 소프트웨어 구성, 지원 플랫폼 등은 그리드에 비하여 일반적으로 동일한 종류이다. 현재 클라우드의 보안 모델은 그리드에 비하여 매우 간단하고 보안성이 약하다. 클라우드 기반구조는 보편적으로 사용자를 위한 계좌 정보를 생성 및 관리하기 위하여 웹 폼을 사용하고 있으며 패스워드를 리셋하고 새 패스워드를 받기 위하여 이메일을 사용한다. 새 사용자는 신용카드와 이메일 주소만을 가지고 신속하고 쉽게 클라우드를 사용할 수 있다. 반면 그리드는 보안에 대하여 매우 엄격하다. 예를 들어 패스워드와 같이 중요한 정보는 팩스나 우편을 통하여 전달되지 절대로 이메일을 통하여 전달되지 않는다.

## 4. 가상화와 클라우드 컴퓨팅의 관리

본 장에서는 클라우드 컴퓨팅 운용 및 관리를 위해 사용되는 기술 중 가장 중요한 요소인 가상화에 대해 설명하고 DMTF의 CIM과 가상화 표준을 사용한 클라우드 컴퓨팅 시스템의 관리에 대해 기술한다.

### 4.1. 가상화의 정의, 특징과 분류

가상화는 하드웨어와 운영체제와 어플리케이션 사이에 소프트웨어 추상화 계층을 도입하는 기술로 정의된다. 이 추상화 계층은 VMM(Virtual Machine Monitor) 또는 Hypervisor라고 불리고 운영체제로부터 컴퓨팅 시스템의 물리적 자원을 감추는 역할을 한다. 하드웨어 자원이 운영체제가 아닌 VMM에 의해 제어되므로 한 하드웨어 상에서 서로 다른 여러 개의 운영체제를 병렬로 수행하는 것이 가능하다. 결과적으로 하드웨어 플랫폼은 여러 개의 논리적 단위(Unit)로 나누어질 수 있고 이 각각의 단위는 가상기계(Virtual Machine: VM)라고 불린다[16]. 이미 앞에서 살펴 본 바와 같이 가상화는 추상화를 통하여 기반의 패브릭이 자원의 풀(Pool)로 통합 되어지고 이 위에 자원 오버레이(예를 들어 데이터 저장 서비스나 웹 호스팅 환경)가 구축될 수 있도록 한다. 또 각 어플리케이션이 캡슐화될 수 있도록 함으로써 어플리케이션들이 구성, 배치, 시작, 이동, 일시 정지, 재개, 종료될 수 있도록 하며 더 나은 보안, 관리용이성, 고립성(Isolation)이 제공될 수 있도록 한다. 그러므로 가상화는 그리드와 클라우드 컴퓨팅에서 매우 중요한 개념으로 사용되고 있다. 이들 컴퓨팅 시스템들이 가상화를 채택함에는 다음과 같은 다른 유용성이 있기 때문이다.

- 다양한 어플리케이션이 같은 서버 상에서 수행될 수 있으므로 자원의 이용률이 높아진다.
- 다양한 어플리케이션들의 자원에 대한 요구가 많이 다를 수 있는데 가상화는 이러한 요구에 맞추어 가상 기계의 구성을 용이하게 할 수 있다.
- 가상 환경은 서비스 중단 없이 백업되거나 이동될 수 있으므로 가상화는 고장 발생 시 빠른 복구를 가능하게 한다.
- 자원의 제공, 모니터링과 유지보수가 자동화 될 수 있고 공통 자원들은 캐쉬되거나 재사용될 수 있다.

가상화를 위한 방법은 다음과 같이 세 가지로 분류될 수 있다.

- 완전(Full) 가상화
- 운영체제 계층 가상화
- 하드웨어 계층 가상화

완전 가상화에서 VMM은 호스트 운영체제 상에서 사용자 공간에서의 어플리케이션으로 수행된다. 그 결과 어플리케이션과 게스트 운영체제는 VMM에 의해 제공되는 가상 하드웨어 상에서 수행된다. 이 방법의 장점은 사용하기 쉽다는 것이지만 성능이 매우 나쁘다는 단점이 있다.

운영체제 계층 가상화에서는 하드웨어가 아니라 호스트 운영체제가 가상화 되므로 호스트 운영체제와 같은 운영체제의 여러 인스턴스들이 만들어진다. 다른 가상화 방법들보다 효율성이 높다는 장점이 있지만 모든 게스트 운영체제들이 호스트 운영체제와 같다는 단점이 있다.

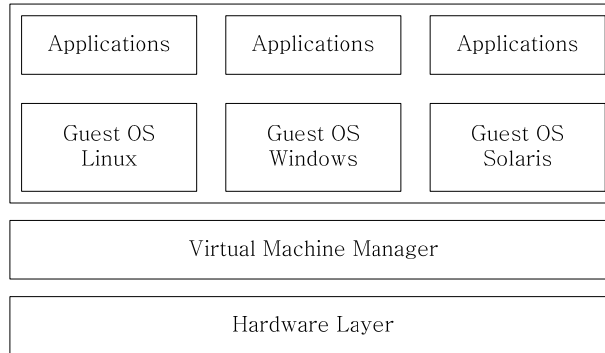


그림 7. 하드웨어 계층 가상화

하드웨어 계층 가상화는 가상 기계의 고립성(Isolation)과 성능이 높으므로 서버 마켓에서 흔히 사용되는 방법이다. 이 방법에서 VMM은 그림 7과 같이 하드웨어 상에서 직접 수행되어 게스트 운영체제들이 하드웨어 자원에 접근하는 것을 조정하고 동기화한다. 이의 예에는 VMWare ESX Server[17]와 Xen[18]이 있으며 그림 8은 Xen VMM을 수행하고 있는 기계의 구조를 보여주고 있다.

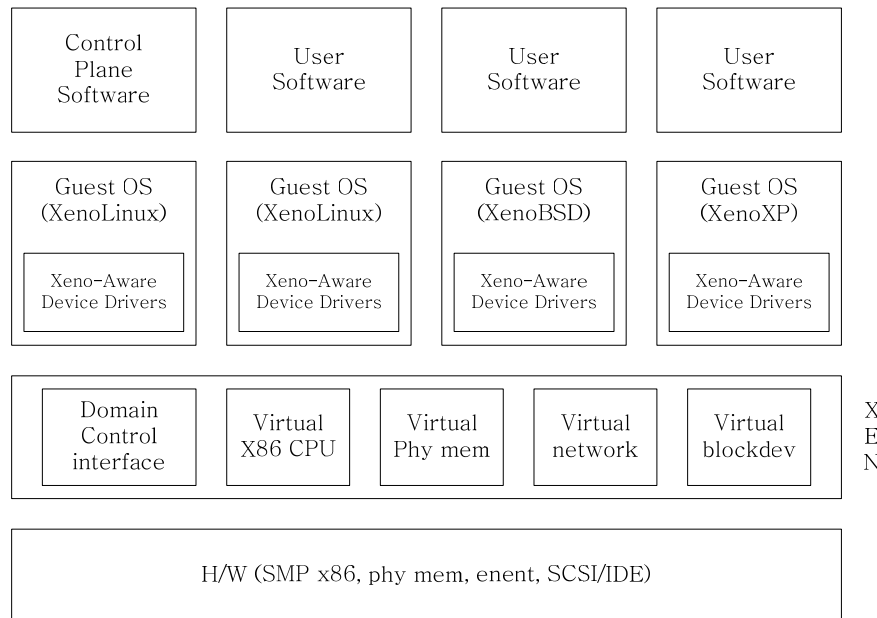


그림 8. Xen VMM을 수행하는 기계의 구조

#### 4.2. DMTF의 CIM과 가상화를 사용한 클라우드 컴퓨팅의 관리

DMTF(Distributed Management Task Force)는 IT 환경에서의 시스템 관리를 위한 표준을 개발하고 유지하는 기관이다. DMTF 표준들은 플랫폼과 기술에 독립적으로 시스템 관리 기반구조 구성요소들을 구축할 수 있도록 하며 고비용의 변환 없이도 서로 다른 IT 제품들 사이에 시스템 관리 상호 운용성을 제공한다. 이를 위하여 DMTF는 다음 분야에서의 표준을 개발한다.

- 관리 이니셔티브(Management Initiative)
- 공통 정보 모델 (Common Initiative Model: CIM)
- 웹기반 기업 관리 (Web-Based Enterprise Management: WBEM)
- 시스템 관리 BIOS (System Management BIOS: SMBIOS)
- 경보 표준 포맷 (Alert Standard Format: ASF)
- 데스크탑 관리 인터페이스 (Desktop Management Interface: DMI)

CIM 분야에서는 IT 환경에서의 관리 대상 요소(Managed Element)를 객체(Object)의 집합과 객체들 사이의 연관관계(Relationship)들로 표현할 수 있도록 하는 개념적 스키마로서 CIM(Common Information Model) 스키마를 정의하였다. CIM은 CIM 스키마를 정의하기 위하여 UML 기반의 모델을 사용하며 이 스키마는 대부분의 DMTF 표준의 기초가 된다.

이러한 DMTF 기술을 바탕으로 다음의 다섯 분야의 관리 이니셔티브들이 각각 특정 수직 응용 분야와 산업 분야에 관리 기능을 제공하고자 한다.

- 공통 진단 모델 (Common Diagnostic Model: CDM)
- 시스템 하드웨어를 위한 데스크탑과 모바일 구조 (Desktop and mobile Architecture for System Hardware: DASH)
- 서버 하드웨어를 위한 시스템 관리 구조 (Systems Management Architecture for Server Hardware: SMASH)
- 가상화 관리 (Virtualization Management: VMAN)
- 저장 관리 이니셔티브 명세 (Storage Management Initiative Specification: SMI-S)

이중 VMAN 이니셔티브에서는 상기의 목적을 달성할 수 있도록 다음과 같은 표준과 프로파일들을 개발하였거나 하고 있다.

- DSP0243: Open Virtualization Format(OVF)
- DSP1041: Resource Allocation Profile
- DSP1042: System Virtualization Profile
- DSP1043: Allocation Capabilities Profile
- DSP1044: Processor Device Resource Virtualization Profile
- DSP1045: Memory Resource Virtualization Profile
- DSP1047: Block Based Storage Resource Virtualization Profile
- DSP1048: File Based Storage Resource Virtualization Profile
- DSP1049: Storage Adaptor Resource Virtualization Profile
- DSP1050: Network Port Resource Virtualization Profile
- DSP1057: Virtual System Profile
- DSP1059: Generic Deice Resource Virtualization Profile

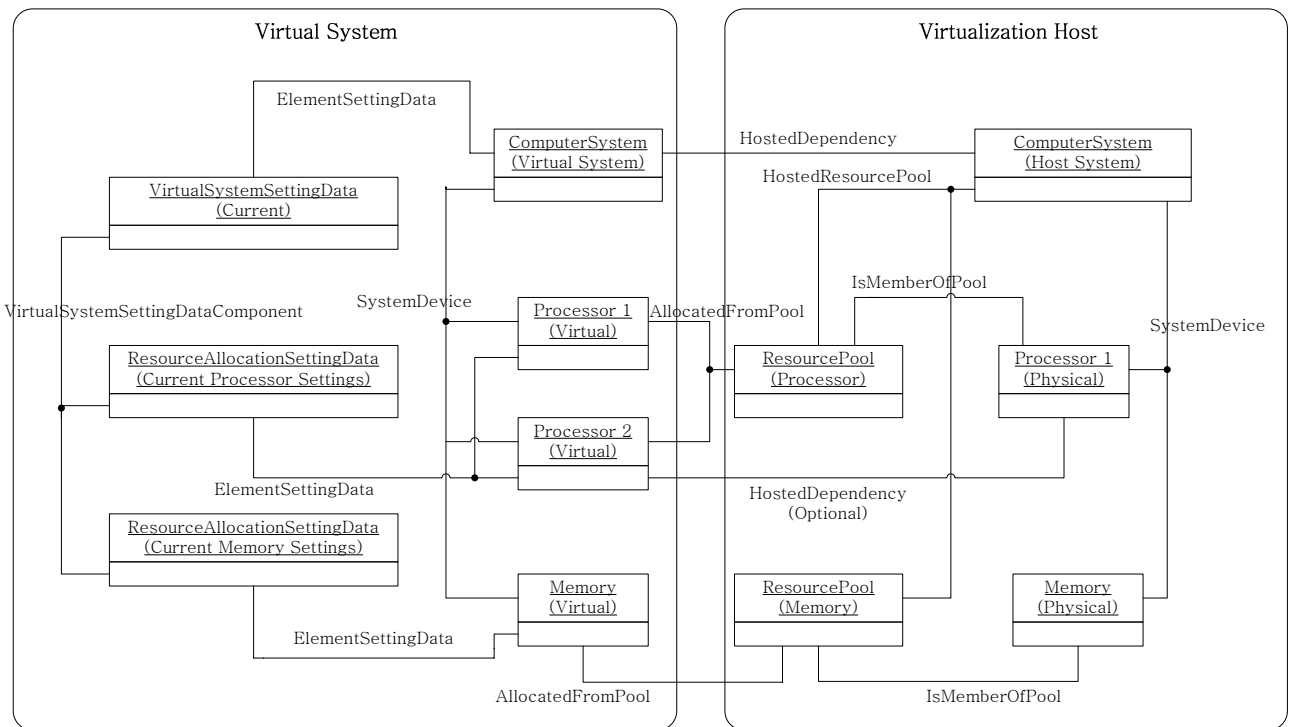


그림 9. 가상 시스템의 생성을 설명하는 객체 다이어그램

물리적 시스템으로부터 자원 풀을 생성하고 구성 데이터를 정의하여 자원 풀로부터 가상 자원을 할당 받아 가상 시스템을 구성하였을 때 CIM 객체들 사이의 관계가 그림 9에 설명되어 있다. 가상 시스템을 생성 및 제거하기 위해서는 다음의 두 관리 오퍼레이션이 정의되어 있으며

- DefineVirtualSystem(): CPU, 메모리, 디스크와 NIC 구성 매개변수를 전달하여 가상 시스템을 생성
- DestoryVirtualSystem(): 가상 시스템을 삭제

가상시스템의 라이프싸이클 관리를 위해서 RequestStateChange() 오퍼레이션이 다음과 같이 정의되어 있다.

- RequestStateChange(Enabled): 시작(Start)/재개(Resume)
- RequestStateChange(Disabled): 종료(Stop)
- RequestStateChange(Quiesce): 임시 정지(Pause)
- RequestStateChange(Enabled but Offline): 서스펜드(Suspend)

최근 Xen과 VMWare ESX Server 등의 VMM과 DMTF에서의 CIM과 VMAN 표준을 결합하는 프로젝트들이 활발히 진행되고 있다. Xen-CIM 프로젝트에서는 DMTF의 가상화와 자원 할당 모델을 Xen에 구현하고 있으며 관리 구조는 그림 10과 같다[19]. 그림에서의 각 모듈의 역할은 다음과 같다.

- CIM 클라이언트: 관리 명령을 CIM-XML 포맷으로 변환한다.
- CIMOM(CIM Object Manager): CIM 요청을 수신하여 이 요청에 해당하는 CIM 클래스를 위한 CIM 제공자에게 요청을 보내고 이 제공자로부터의 응답을 수신하여 클라이언트에 전달한다.
- CIM 제공자(Provider): 관리 대상 자원과 서비스의 오퍼레이션을 구현하며 가상 기계 별로 배치된다.
- 지역적 API, CLI와 디먼: 실제 호스트 자원을 관리한다.

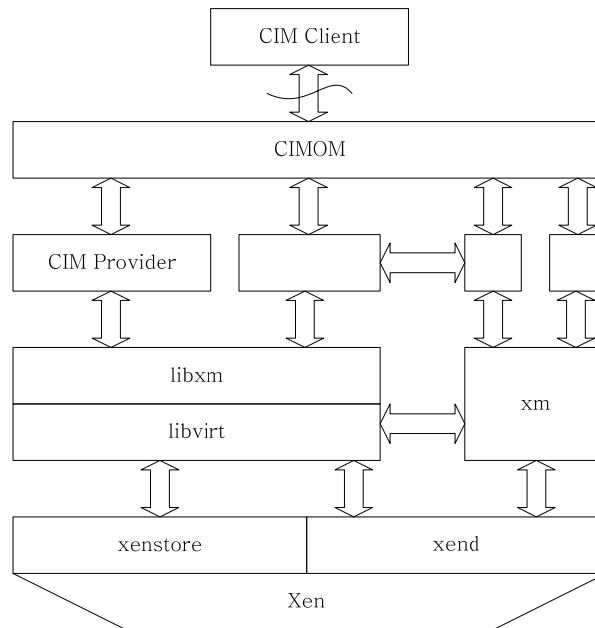


그림 10. Xen-CIM 관리 구조

VMWare는 여러 개의 관리 API를 개발하였는데 VMWare 기반구조의 관리를 위해서는 두 개의 API가 개발되었다. VMWare 기반구조 SDK는 Virtual Center에 대한 웹 서비스 인터페이스를 제삼자 도구 개발자들에게 제공한다[20]. 반면 CIM SDK는 CIM 인터페이스를 통하여 ESX 서버를 관리할 수 있도록 한다[21].

## 5. 결론

최근 클라우드 컴퓨팅에 대한 관심이 매우 높아지고 있지만 또 동시에 클라우드 컴퓨팅의 관련 기술들과의 공통점으로 인해 많은 혼란이 있는 것도 사실이다. 본 논문에서는 먼저 클라우드의 컴퓨팅의 정의,

특징, 서비스, 구조와 국내외 기술 및 표준화 동향에 대해 기술한 후 가장 중요한 관련 기술이 되는 그리드 컴퓨팅과 다양한 관점에서 비교 분석하였다.

클라우드 컴퓨팅은 인터넷 상에서의 대규모 분산시스템이므로 이의 효율적인 운용과 관리는 매우 중요한 이슈가 되며 이를 위하여 가상화 기술이 매우 중요하게 사용되고 있다. DMTF에서는 분산 시스템의 효율적인 관리를 위하여 CIM 스키마를 개발하고 이를 기반으로 하여 가상화 표준을 개발하였다. 본 논문에서는 가상화 기술에 대해 설명하고 DMTF의 CIM 스키마와 가상화 표준이 클라우드 컴퓨팅의 운용 및 관리에 어떻게 적용되는지에 대해 설명하였다.

DMTF의 CIM 스키마와 가상화 표준은 클라우드 컴퓨팅의 운용과 관리를 위한 가상 기계의 생성, 시작, 종료, 파괴 등의 오퍼레이션들과 가상 기계의 상태를 모니터링하고 변경할 수 있는 오퍼레이션 등을 제공하고 있다. 그러나 가상 기계 상에서 사용자의 어플리케이션을 사용자가 요구하는 품질 수준을 만족하며 수행하기 위하여 필요한 성능, 결합허용성, 보안 등의 측면에서 다양한 관리를 수행하여야 하는데 이를 위하여 CIM 스키마와 가상화 표준을 어떻게 활용할 것인가에 대한 구체적인 연구가 필요하다. 또 이러한 연구 과정에서 DMTF의 가상화 표준들에 변경에 필요할 수도 있는데 이에 대한 연구도 필요할 것이다.

## 6. 참고문헌

- [1] M. Armbrust et al, "Above the Clouds: A Berkeley View of Cloud Computing," University of California, Department of EECS Technical Report No. UCB/EECS-2009-28, Feb. 2009.
- [2] I. Foster, Y. Zhao, I. Raicu, & Shiyong Lu, "Cloud Computing and Grid Computing 360-Degree Compared," Grid Computing Environments Workshop, 2008.
- [3] <http://aws.amazon.com>.
- [4] IBM, "Seeding the Clouds: Key Infrastructure Elements for Cloud Computing," Feb. 2009.
- [5] <http://code.google.com/appengine>.
- [6] <http://www.microsoft.com/azure>.
- [7] <http://www.ibm.com/cloud>.
- [8] <http://www.rightscale.com>.
- [9] <http://3tera.com>.
- [10] <http://www.rackspace.com>.
- [11] <http://www.servepath.com>.
- [12] <http://www.opencloudconsortium.org>.
- [13] <http://www.occi-wg.org>.
- [14] <http://www.dmtf.org>.
- [15] I. Foster, "What is the Grid: A Three Point Checklist," July 2002.
- [16] D. Marinescu & R. Kroeger, "State of the Art in Autonomic Computing and Virtualization," Distributed Systems Lab, Wiesbaden University of Applied Sciences, 2007.
- [17] <http://www.vmware.com/products/vi/esx/>.
- [18] P. Barham et al, "Xen and the Art of Virtualization," SOSPO3, pp.164-177, Oct. 2003.
- [19] <http://wiki.xensource.com/xenwiki/XenCim>.
- [20] <http://www.vmware.com/support/developer/vc-sdk>.
- [21] <http://www.vmware.com/support/developer/cim-sdk>.



심 영 철

1979 서울대학교 전자공학과 학사

1981 한국과학기술원 전기및전자공학과 석사

1981~1984 삼성전자 컴퓨터사업부

1984~1991 University of California, Berkeley 전산학 박사

1992 University of California, Berkeley 연구원

1993~현재 홍익대학교 정보컴퓨터공학부 교수

<관심분야> 무선네트워크 프로토콜, 보안, 분산시스템관리