

특집  
05 |

## 컴퓨팅 클라우드 자원 제어 기술

### 목 차

1. 서 론
2. 클라우드 자원 관리의 중요성
3. Open Cloud Platform 구축 시 고려되는 자원관리 기법
4. 결 론

송혜원 · 김희영 · 윤찬현  
(KAIST)

### 1. 서 론

2009년의 IT산업의 주요 화두는 클라우드와 가상화 기술로 요약될 수 있다. 소프트웨어 전문가들은 수백만의 소비자들이 개인 컴퓨터에서 실행하기보다 서비스로서 사용하는 소프트웨어를 만들어내기 위해 수많은 도전들에 직면해 있다. 멀티코어 프로세서나 네트워크 컴퓨팅 환경처럼 급격하게 진보하는 기술들을 이용한 새로운 컴퓨팅 패러다임들이 제안되고 도입되고 있다. 이러한 새로운 컴퓨팅 패러다임에는 클러스터 컴퓨팅, 그리드 컴퓨팅, P2P 컴퓨팅, 서비스 컴퓨팅, 시장 지향 컴퓨팅 등이 포함되며, 최근에는 클라우드 컴퓨팅까지 새롭게 등장하고 있다[1~5]. 컴퓨팅 서비스들은 높은 신뢰성과 확장성, 그리고 유비쿼터스 액세스나 동적 발견과 구성을 지원하는 자율성을 필요로 한다. 특히 소비자들이 서비스 품질(Quality of Service: QoS) 파라미터와 서비스 요구 수준 협약 (Service Level Agreement: SLA)를 통해 요구된 서비스 수준을 결정할 수 있어야 한다.

주요 IT기업들의 관심이 집중되고 있는 클라

우드 컴퓨팅은 프로그램이나 문서를 인터넷으로 접속할 수 있는 대형 컴퓨터에 저장하고 PC/휴대폰과 같은 각종 단말기로 원격에서 원하는 작업을 수행할 수 있는 사용 환경을 말한다. 프로그램이나 자료를 사용자의 PC가 아닌 인터넷으로 접속가능한 대형 컴퓨터에 저장하고, PC는 물론 휴대전화 같은 다양한 기기를 이용해 원하는 작업을 수행하는 점에서 기존의 썬 클라이언트(Thin Client) 혹은 메인프레임 환경과 유사한 모습을 띠고 있다. 이론적으로 '클라우드 컴퓨팅'이 확실히 자리를 잡으면 이용자는 인터넷 접속과 연산 기능만 갖춘 '단말기'만 있으면 장소를 불문하고 원하는 전산 작업을 할 수 있다. 하지만 수많은 개인들의 정보 관리 및 가공을 위한 SW는 누군가의 관리를 받는 컴퓨터들에 저장돼야 하기 때문에 PC혁명이 시작된 지 20년 만에 다시 중앙 집중식 '메인 프레임'컴퓨터를 이용하는 시대가 올 지도 모른다는 전망이 제기되고 있다[3].

클라우드는 가상화된 컴퓨터 자원들의 풀(pool)로 정의될 수 있으며, 서로 다른 작업량(workload)의 다양성을 수용하고 클라우드 안에

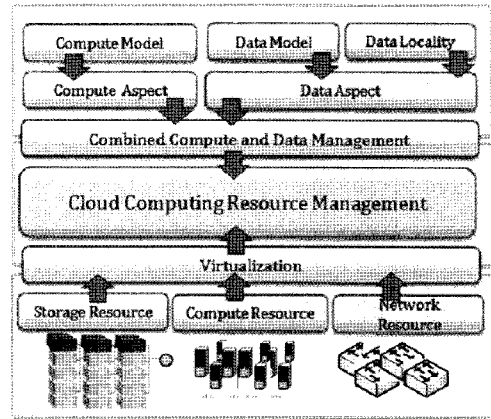
위치한 가상 머신(Virtual Machine: VM)이나 물리적 머신에 대한 신속한 프로비저닝을 제공하며, 또한 자율화된 자원 및 작업 관리를 위한 확장된 자원 관리 모델 등을 지원하여 원할 때마다 자원 할당의 밸런싱을 맞추거나 효율적인 자원 할당 및 스케줄링을 위해 실시간으로 자원 사용에 대한 모니터링 기능을 제공한다[5]. 특히 클라우드 컴퓨팅 인프라는 엔터프라이즈들이 훨씬 효율적으로 그들의 IT 하드웨어와 소프트웨어 장치를 이용하도록 하여, 고립된 시스템들간의 물리적인 벽을 허물고 마치 하나의 존재로서 전체 시스템 그룹의 관리를 자동화를 지원한다. 즉, 클라우드 컴퓨팅은 궁극적인 가상화 시스템이며 자율화 시스템 관리와 작업량 밸런싱 기법, 그리고 가상화 기술이 적용된 가상 데이터 센터 혹은 가상 컴퓨팅 센터가 된다. 따라서, 이런 클라우드 환경에서 사용자들이 요청한 작업들에 대해 그들이 명시한 QoS 및 SLA를 준수하여 자원을 제공하기 위하여 보다 클라우드 환경에 적합한 자원 관리 및 제어 기술이 요구된다.

본문에서는 클라우드 환경에서 자원 관리의 주요 이슈 및 그 중요성에 대해 기술하고, 가상화 기반의 클라우드 플랫폼인 사이언스 클라우드의 구축 예시와 함께 다양한 가상화 자원 관리 - 특히 스케줄링 - 기법에 대하여 기술한다.

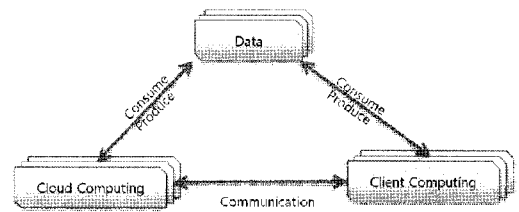
## 2. 클라우드 자원 관리의 중요성

클라우드는 가상화 기술이 적용된 가상 데이터 센터 혹은 가상 컴퓨팅 센터라고 할 수 있다. 이러한 클라우드 환경에서 대규모의 가상화 서버를 운영 및 관리 할 때, 작업량의 밸런싱 등과 같은 정교한 스케줄링 및 할당 기술이 적용된 자율 관리(Self-managing) 시스템 기반의 자원 관리 시스템이 요구된다. 특히, 클라우드 환경에서 가상 클러스터(Virtual Cluster) 내 자원의 사용을 정교하게 제어하여 데이터 센터 내의 모든 가상 클러스터들이 양호한 수행을 유지하도록

록 지원하는 최적화 자원 할당 및 스케줄링 기법이 필요하다. (그림 1)은 이러한 클라우드 자원 관리 기술에 대해 컴퓨팅 및 데이터 모델 측면에서, 클라우드 핵심 기술인 가상화 측면에서, 그리고 자원 모니터링 측면에서 나타내고 있다[6].

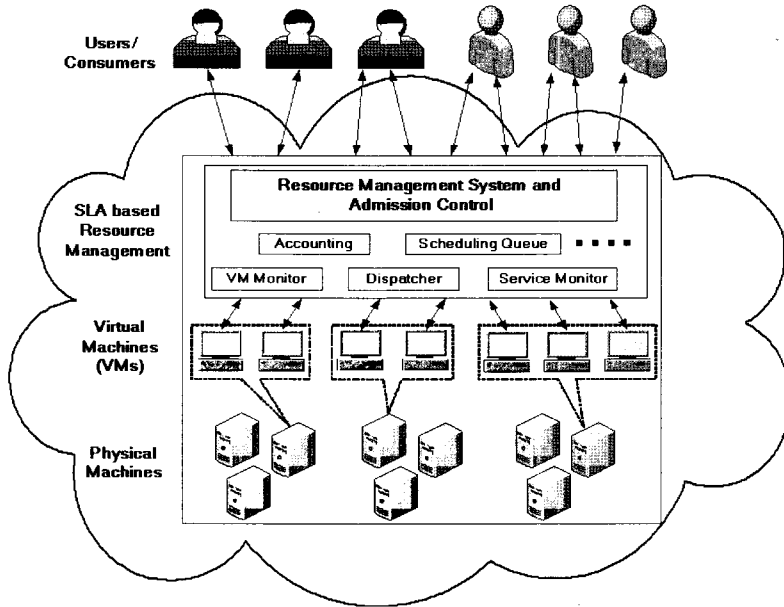


(그림 1) 클라우드 컴퓨팅 자원관리 이슈



(그림 2) 클라우드 컴퓨팅, 클라이언트 컴퓨팅 및 데이터의 연관 모형[6]

클라우드 컴퓨팅은 모든 사용자들이 자원을 동시에 공유하고 있으며, 스토리지와 컴퓨팅, 그리고 모든 종류의 자원들이 제공되는 환경이다. 이러한 환경 제공을 위해 데이터와 클라우드 컴퓨팅, 그리고 클라이언트 컴퓨팅으로 연계되는 모델 (그림 2)이 사용되고, 이러한 관계에서는 데이터 관리 (매핑, 파티셔닝, 쿼링, 이동, 캐싱, 복제 등)가 필요한 데이터 집중형 응용의 증가할수록 그 연계성이 커지게 된다. 또한 컴퓨팅 응용을 처리할 때, 네트워크 상에서 확장성을 얻



(그림 3) 계층별클라우드구조[7]

기 위하여, 통신 비용을 최소화 하는 최적의 장소에서 데이터가 처리될 수 있도록 조정해야만 하는데, 이 과정에서 계산과 데이터 자원 관리의 결합은 데이터 이동을 최소화하고 중단 응용의 수행능력과 확장성을 개선하기 위한 중요한 해결 방안이 될 수 있다.

가상화는 기본 실제 자원들(컴퓨터, 스토리지, 네트워크 자원)이 풀(pool)로써 통합되고, 그 위에 오버레이 자원 환경이 구축되어 필수적인 추상화를 제공한다. 클라우드 환경에서는 여러 개의 사용자 응용들이 수행될 필요가 있고, 모든 응용들은 동시에 수행되고, 클라우드에 있는 모든 자원을 사용할 수 있는 것처럼 사용자들에게 보여야 한다. 따라서, 클라우드는 가상화를 지원함으로써, 다양한 응용들이 같은 서버에서 수행될 때, 자원을 좀 더 효율적으로 사용할 수 있는 서버와 응용 통합을 지원하고, 다양한 응용을 위한 자원 요구사항이 확연히 다를 때, 다양한 필요에 따라 동적으로 자원을 구성하고 묶어 지원할 수 있는 구성력을 제공할 수 있다. 또한 가상

환경에서 데이터가 백업되고 서비스 중지 없이 서비스를 이전할 때, 계획되지 않은 오작동이나 작업 실패로부터의 빠른 복구 지원을 통해 응용의 유효성을 증가시키고, 자원 프로비저닝과 모니터링, 유지의 자율화가 지원될 수 있다. 그러나, 클라우드 환경에서 가상화의 어려움은 실시간으로 자원 모니터링하고, 자율 유지(Self-Maintained) 및 자율 치료(Self-Healing)가 제공되는 자율 관리에 의한 정교한 제어를 하는데 있다. 특히, 자원 모니터링을 위해, 클라우드는 비즈니스 응용 모니터링과 엔터프라이즈 서버 관리, 가상 머신 모니터링, 하드웨어 유지 간의 밸런스를 요구하고 있다.

이러한 자원 관리 상의 이슈들을 고려할 때, 클라우드 환경에서의 효율적인 자원 관리를 위한 체계화된 구조가 필수적이다. (그림 3)은 데이터 센터와 클라우드에서 자원 할당을 지원하기 위한 기본적인 계층구조를 나타낸 것이다. (그림 3)에서 볼 수 있듯이, 클라우드 자원 관리 구조는 사용자/브로커 (User/Broker), SLA 자

원 할당자(SLA Resource Allocator), 가상 머신(Virtual Machines) 및 물리적 머신(Physical Machines)의 4계층으로 구성된다. 사용자나 브로커는 지리적인 위치에 관계없이 데이터 센터와 클라우드에 작업 처리를 위해 서비스 요청을 제출할 수 있으며, SLA 자원 할당자는 데이터 센터나 클라우드를 소유한 서비스 제공자와 사용자/브로커 사이에 인터페이스와 같은 역할을 한다. SLA 자원 할당자 계층에는 요청된 서비스에 대한 수용 여부를 결정하고, 수용이 결정된 서비스를 수행하기 위해 적절한 자원을 선정/할당/스케줄링하는 서비스 요청 수용자 및 작업 수용 관리자와 서비스 요청에 대한 과금이나 그 사용비용을 계산하거나 부과하기 위한 계정, 가상 머신과 자원의 가용성을 관리하는 가상 머신 모니터, 그리고 수용된 작업이 할당된 가상 머신을 실행시키는 디스패처(Dispatcher)로 구성된다. 가상 머신은 서비스 요청의 세부적인 요구사항을 기반으로 동일한 실제 머신 내에서 자원을 분할하여 사용하는 형태로 제공되며, 하나의 동일한 실제 자원 내에서 독립적이고 분별된 형태로 제공될 수 있어 자원 사용의 증대를 가져온다. 물리적 머신들은 사용자의 요구사항을 만족시키기 위한 실제 자원인 다양한 서버와 네트워크로 구성된다.

이러한 클라우드 자원 및 작업 관리 기술은 클라우드 컴퓨팅이 좀 더 광범위하게 적용되고 배치될 때 더 중요한 요소가 될 것이며, 클라우드 컴퓨팅의 기술 성숙을 위해 필수적인 요소라고 할 수 있다. 따라서 다음 장에서는 현재 연구 진행 중인 클라우드 플랫폼 기술 중 사이언스 클라우드 플랫폼의 사례를 통해서 현재까지 연구되어 온 다양한 클라우드 자원 관리 기술에 대해 기술한다.

### 3. Open Cloud Platform 구축 시 고려되는 자원관리 기법

클라우드 컴퓨팅 구현 스텝은 1)클라우드 구축, 2)클라우드 관리, 3)클라우드 실험 세가지 스텝으로 이루어져 있다. 클라우드 구축은 컴퓨팅 자원과 네트워크 자원을 가상화 시켜 클라우드에서 관리 가능한 시스템을 구축하는 작업을 말하며 클라우드 관리는 가상화된 자원들을 어떻게 효과적으로 관리할 것인가를 나타낸다. 또 마지막으로 클라우드 실험은 최종적으로 클라우드를 운영하는 과정에서 발생하는 문제점을 확인하거나 최종 서비스를 위한 부가 기능을 다루는 것을 목표로 한다.

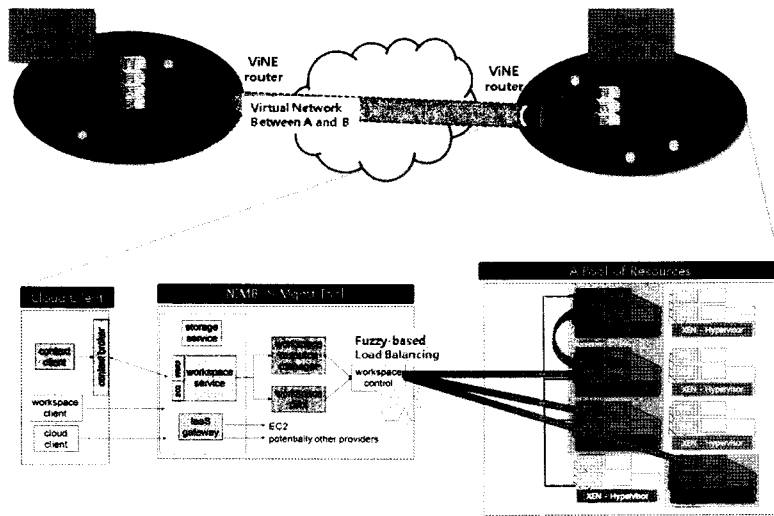
#### 3.1 클라우드 구축

현재 다양한 클라우드 자원들이 다양한 관리 자들에 의해 구축되어 운영 중이다. 미국의 플로리다 대학 연구그룹에서는 이와 관련하여 네트워크 연결성을 지원하기 위한 ViNe[8] 미들웨어와 클라우드 컴퓨팅 센터 Open 플랫폼인 Nimbus[9]를 이용하여 (그림 4)과 같은 클라우드 컴퓨팅 플랫폼을 구축하여 시험 운영 중이다.

ViNe은 지역적으로 분리된 네트워크 도메인 간에 협업을 할 때, 네트워크의 대칭성을 제공하여 서로 다른 도메인 간의 협업이 보다 쉽게 제공될 수 있는 가상 네트워크 환경을 제공하는 기술이며, Nimbus는 클라우드 컴퓨팅을 실현하기 위해 가상 머신의 실행환경을 동적으로 설정할 때, 특히 소프트웨어의 동적 설정이 가능한 기술이다. 따라서, 본 장에서는 클라우드 구축을 위한 기본 인프라가 되는 ViNe과 Nimbus 기술을 기술한다.

##### 3.1.1 Virtual Network (ViNe)[8]

그리드 컴퓨팅 환경에서와 같이 분산된 자원들간에 협업을 하기 위해서는 대칭형 네트워크 연결이 필수적이지만, 현재 인터넷은 비대칭형 환경이다. 이러한 네트워크 비대칭은 주로 사설망(Private Network)의 네트워크 주소 변환



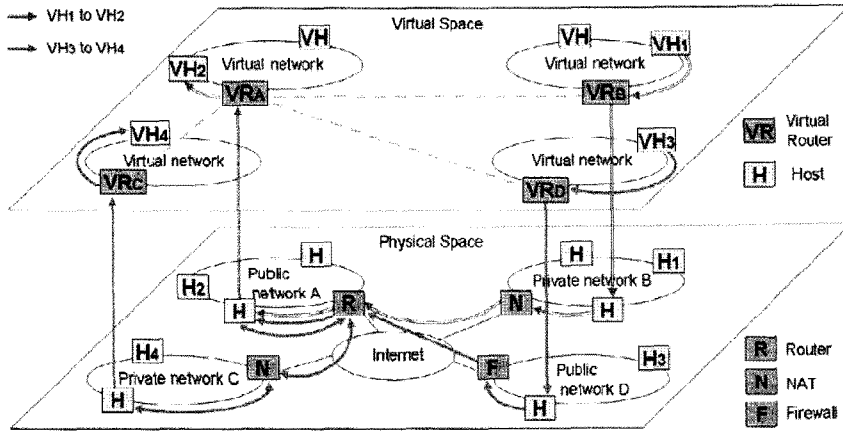
(그림 4) Nimbus 기반의 Science Cloud 플랫폼

(Network Address Translation: NAT) 게이트웨이 기술과 방화벽(Firewalls) 때문에 발생할 수 있다. 따라서 서로 다른 도메인에 속한 네트워크 간 자원들을 사용하여 작업을 수행하는 그리드 컴퓨팅과 같은 기술을 지원하기 위해서, 사설망이나 방화벽에 관련된 복잡한 문제를 먼저 해결해야만 한다. [8] ViNe는 이러한 사설망이나 방화벽에 상관없이 그리드 컴퓨팅 환경과 같이 서로 분산된 자원들 간에 협업을 지원하기 위해서, 1) 존재 요소 (Entity) 간의 대칭형 통신과 방해 요소에 의해 중단되지 않는 네트워크 환경, 2) 용이한 네트워크 구성, 각각의 자원 제공자에 의해 제시된 보안 정책에 대한 독립성, 3) 네트워크 정의와 배치에 대한 자동화를 지원하는 메커니즘의 이용성, 4) 수정 없이 응용을 수행하는 능력, 5) 변화 없는 운영체제(Operating System: OS) 네트워크 스택의 사용, 6) 중단 없는 인터넷 인프라, 7) 플랫폼 독립성과 확장성 등을 만족시키는 가상 네트워크 구조를 제공한다.

ViNe의 1차적인 목표는 호스트들 간에 양방향 통신이 가능한 가상 네트워크 환경의 생성이다. 변경 없는 응용을 지원하기 위해, ViNe 구조는 IP-overlay 구조이며, 네트워크 인터페이스,

라우터, 링크들과 같은 구성요소들은 가상화가 된다. ViNe에 참여한 호스트는 원래 할당된 IP 주소 외에 하나 이상의 IP 주소를 추가적으로 할당 받으며, 이렇게 추가 할당된 IP 주소는 물리적 네트워크와 충돌 없는 가상 주소가 된다. 일단 인터넷망은 ViNe 트래픽을 다루지 않기 때문에, ViNe 라우팅 소프트웨어를 실행시키는 물리적 네트워크에 연결된 호스트로서 가상 라우터 (Virtual Router: VR) 가 필요하다. (그림 5) 이러한 ViNe의 전체 구조를 나타내며, 이러한 ViNe 구조를 기반으로 가상 네트워크의 생성과 삭제는 관리적인 간섭 없이 ViNe에 의해 동적으로 제어되며, VR은 물리적 공간에서 터널을 통해 서로 연결되어 있다.

우선 ViNe는 방화벽이나 NAT 게이트웨이에 의한 연결 제약이 있는 호스트가 대칭형 통신이 가능하도록 하기 위해서 IP aliasing을 사용하여 여러 개의 IP 주소를 가질 수 있도록 한다. 또한, 가상 주소 스페이스를 위해 사설 주소 (Private Addressing) 스페이스를 예약하여 이미 사설망 내에 위치해서 사설 주소를 사용하고 있을 경우를 위해서 20-bit 스페이스를 사용한다. 게다가, 방화벽이나 NAT 게이트웨이에 의해 제한된



(그림 5) ViNe의 전체 구조[9]

limited-VR과의 통신의 경우, VR을 대기 행렬 서버(Queue Server)인 queue-VR로서 할당하고, limited-VR과의 사이에 TCP 채널을 통해 패킷을 전송함으로써 통신을 할 수 있다.

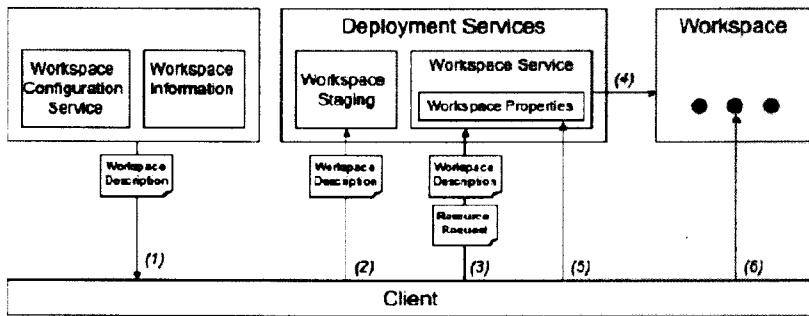
앞서 말한 바와 같이, VR은 ViNe 패킷을 전송하는 매개물로서, IP 전송 구조를 가지고 있다. VR은 VR간의 통신을 위해 로컬 네트워크 기술 테이블(Local Network Description Table: LNDT)와 글로벌 네트워크 기술 테이블(Global Network Description Table: GNDT)를 관리한다. LNDT는 같은 물리적 네트워크에 연결된 호스트들에 대한 정보를, GNDT는 패킷이 전송되어야 할 목적지 VR의 물리적 IP 주소를 나타내는 가상 네트워크 구조에 관련된 정보를 저장한다. 일단 호스트가 특정 네트워크로 패킷을 전송할 때, 우선 가상 네트워크 ID는 LNDT 내에서 검증되고 난 후, 상응하는 GNDT를 이용하여 패킷을 목적 VR까지 터널링한다. VR은 가상 네트워크 공간에서 라우터인 동시에 방화벽으로 동작하므로, 라우팅 테이블에 의해 정의된 룰(Rule)에 의해 패킷을 포워딩(Forwarding)하거나 누락(Drop) 시킴으로써 가상 네트워크의 고립성(Isolation)을 보장한다.

마지막으로, ViNe는 VR 간의 통신을 위해 사

용되는 패킷은 공용 키 인프라(Public Key Infrastructure: PKI) 모델을 사용해 보호되며, VR은 공용 키(Public Key)에 의해서 확인되고, 신용 있는 증명 허가(Certificate Authority: CA)에 의해 유효한 보증 과정을 거쳐야 ViNe에 가입할 수 있다.

### 3.1.2 Nimbus[9]

가상화를 기반으로 하는 클라우드 컴퓨팅을 실현하려는 경우 가상 머신의 실행환경을 동적으로 실행환경을 설정하는 것이 중요하다. 여기서 가상 머신의 실행 환경이란 실제 물리적 컴퓨터로부터 할당 받은 자원의 량(CPU, Memory 할당량)과 소프트웨어 설정 상태를 말한다. 일반적인 가상화 기반 관리 툴이 가상 머신이 할당 받을 자원량을 동적으로 설정하는 데에는 연구가 많이 진행되었지만 가상 머신에 동적으로 소프트웨어의 배치하고 운영하는 것은 활발히 연구되어지지 않았던 부분이다. NIMBUS는 이러한 동적 소프트웨어 설정에 관심을 두고 연구를 진행하고 프로젝트 및 시스템 이름이다. NIMBUS는 동적 소프트웨어 설정을 위한 구성 요소를 첫 번째 가상 머신 이미지 실행 요구사항 기술 단계, 두 번째 동적으로 설치되어야 할 소



(그림 6) NIMBUS 가상 머신 배치 순서도[10]

소프트웨어 기술 단계 그리고 세 번째 그렇게 설치된 소프트웨어들의 동적인 환경 설정 단계로 나누었다. 이러한 기술내용들은 가상 머신의 메타데이터로 작동하며 전체 이미지가 아닌 핵심 정보만을 따로 추상화하여 가지고 있기 때문에 관리자 입장에서 기존의 시스템과 달리 막대한 양의 가상화 이미지를 유지할 필요가 없다.

NIMBUS는 이러한 가상화기반의 클라우드 컴퓨팅을 위해 클러스터를 가상 이미지와 관련된 메타 데이터를 유지하는 시스템 그리고 관련된 실제 커널(Kernel)과 소프트웨어를 위한 저장소, 가상 머신을 실제 배치될 물리적 자원들, 그리고 앞의 3가지 자원을 통해 실제 가상 머신을 배치하는 관리자로 구분하여 운영하게 된다.

그러면 NIMBUS에서 세 종류의 정보로 구성된 메타 데이터를 통해 가상 머신이 운영되는 단계를 살펴보자. 우선 가상 머신과 관련된 메타데이터는 클러스터 내 일정한 저장장소에 저장되게 된다. 만약 외부로부터 서비스 요청의 량이 현재 시스템이 처리하는 량보다 커지는 경우 이를 시스템이 감지하고 이에 대응하기 위한 추가적인 가상 머신의 운영을 명령하게 된다. 이때 NIMBUS는 이러한 서비스에 대응하기 위해 필요한 가상 머신에 대해 기술한 메타 데이터 정보를 확인하고 세가지 단계에 맞추어 서비스 실행을 위해 단계를 수행하게 된다. 우선 첫 번째 요소인 가상 머신 이미지 실행 요구사항을 파악하

는 단계이다. 가상 머신은 어떠한 하이퍼바이저(Hypervisor)를 사용하느냐 그리고 어떠한 구조를 사용하느냐에 따라 종속적이기 때문에 이를 파악하는 단계가 선행되어야 한다. 그리고 메타데이터 기술된 요구조건을 만족 시 이에 따라 중앙 저장소로부터 우선 가상 머신에 커널을 가져와서 실행을 시키게 된다. 그러나 현재의 가상 머신 위의 커널의 소프트웨어 설정이 되어 있지 않기 때문에 현재의 외부 서비스 요청을 받아들일 준비가 되어 있지 않은 상황이다. 두 번째는 동적으로 설치되어야 할 소프트웨어의 기술 정보를 보고 관련된 소프트웨어를 중앙 저장소로부터 가져와 커널이 운영되고 있는 가상 머신에 설치하는 단계이다. 그리고 마지막을 이렇게 설치된 소프트웨어의 운영을 위해 세 번째 메타데이터인 환경 설정 기술서를 보고 환경 설정을 마무리하고 시스템은 최종적으로 외부 요청에 서비스를 처리할 수 있게 된다. NIMBUS는 현재 이러한 단계를 수행하는 4가지 시스템 (메타데이터 저장소, 커널과 소프트웨어를 위한 저장소, 가상 머신을 위한 자원들, 관리자들) 간의 프로토콜을 위해 Web Service Resource Framework (WSRF)를 이용하고 있다.

NIMBUS가 제안하는 이러한 가상 머신 관리 기술은 현재 오픈 프로젝트로써 누구나 이를 테스트해 보고 관련 방법을 수정해 볼 수 있다. 최근 들어 NIMBUS의 연구는 다양한 수의 소

소프트웨어 환경을 대상으로 동적인 설치 및 환경 설정 방법을 연구 중에 있으며 더불어 아마존 EC2 서비스와 같은 기존의 가상화 기반 클라우드 컴퓨팅 서비스와의 호환성을 연구하고 있다[9][11].

## 3.2 클라우드 관리

가상화 기술을 이용하여 클라우드의 사용량을 사용자의 요구사항에 따라 동적으로 제공하는 기술은 관련 연구 그룹들이 가장 주목하고 있는 주제이다. 특히, 클라우드 자원을 관리 유지하는 입장에서 클라우드 운영자의 경제적 이익과 밀접한 관련을 갖는 자원 관리 기술은 최소한의 비용으로 클라우드 사용자의 요구사항에 따라 자원을 효과적으로 제공할 수 있어야 한다. 현재 가상화 된 자원 상태에 대한 예측을 기반으로 동적인 자원 관리를 제공하는 기술들이 다양하게 연구되고 있는데, 본 장에서는 이러한 자원 관리 기술 중 가상화된 자원을 효율적으로 지원할 수 있는 자원 스케줄링 기법을 기술한다.

### 3.2.1 가상화 기반 자원 스케줄링 기법

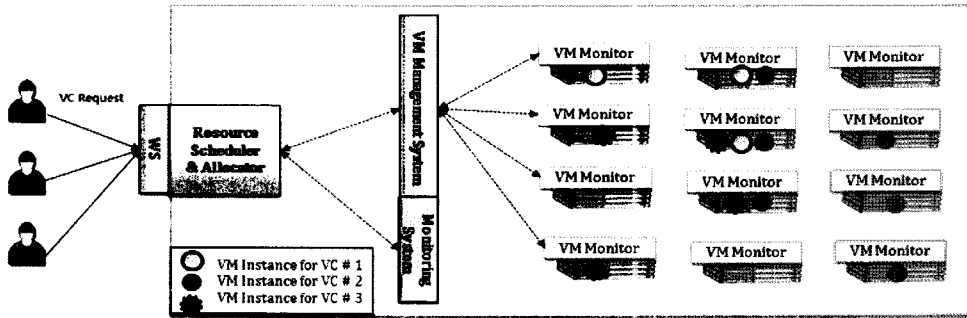
가상 머신은 실제 컴퓨터와 같이 프로그램들을 수행할 수 있는 소프트웨어 기반 머신의 구현을 의미한다. 즉, 시스템 가상 머신은 운영체제(OS)의 완전한 실행을 지원할 수 있는 시스템 플랫폼을 의미한다. 또한, 가상 클러스터(Virtual Cluster: VC)는 여러 가상 머신들이 연결되어 있는 그룹을 의미하며, 이를 통해 마치 하나의 컴퓨터처럼 구성할 수 있는 것을 의미한다. 일반적으로 가상 머신 사이에는 LAN, VLAN, 또는 VPN을 이용하여 고성능 네트워크를 구성하여 사용한다.

이러한 가상 머신의 사용은 하나의 컴퓨팅 노드 안에 독립적이고 구별된 여러 개의 운영체제 환경을 구현할 수 있다는 이점이 있다. 또한, 동일한 컴퓨팅 노드에서 가상 머신 환경을 설정함

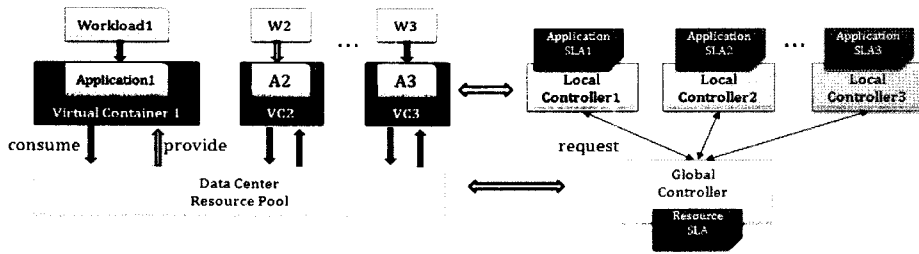
으로써 컴퓨팅 노드 내에 있는 자원을 분할하여 할 당하고 이용할 수 있는 이점도 제공한다. 따라서, 클라우드 환경에서 가상화 기반의 자원 스케줄링을 위해서는 제한된 컴퓨팅 자원에서 다양한 사용자들이 요청한 자원에 대한 요구사항들을 만족시키기 위해 동적으로 가상 머신을 생성하여 실행되고 정지될 수 있는 가상 머신 기술을 사용할 필요가 있다. 특히, 데이터 센터나 클라우드 환경에서 SLA 기반의 자원 할당을 더욱 효과적으로 지원하기 위해서 여러 사용자들의 필요나 요구사항에 맞게 서비스를 제공하기 위해서 다양한 자원 관리 정책을 가상 머신에 할당할 수 있는 방법이 연구되어야 한다.

(그림 7)은 클라우드 컴퓨팅 환경에서 동일한 컴퓨팅 노드로 구성된 데이터 센터 내 가상클러스터를 실행하는 시스템의 구조를 나타낸다. 데이터 센터내의 각 컴퓨팅 자원은 가상 머신 모니터(VM Monitor, Hypervisor)가 설치되어 있고 LAN으로 연결되어 있다. 작업의 실행은 사용자의 작업 요청으로부터 시작된다. 사용자가 자신의 작업을 실행하기 위해서 작업을 요청하면 시스템은 사용자가 요청한 작업을 실행하기 위해서 가상 머신의 집합이나 가상클러스터를 할당한다. 이때 자원 스케줄링이나 작업 스케줄링이 수행되어 최적의 자원 선정이나 작업 선정이 이루어진다. 각 실제 컴퓨팅 노드에 할당된 가상 머신들은 각 컴퓨팅 노드에 있는 가상 머신 모니터에 의해 조절된다. 가상 머신 모니터는 각 호스트에서 수행되는 가상 머신들의 자원 사용률 등을 설정하고 조절할 수 있다. 또한, 모든 가상 머신 모니터는 최종적으로 가상 머신 관리 시스템(VM Management System)의 컨트롤에 의해 조정된다. 가상 머신 관리 시스템이 제공하는 중요한 기능 중에 하나는 실제 컴퓨팅 자원 사이에서 가상 머신 이동(VM Migration)을 가능하게 하는 것이다. 마지막으로, 자원 스케줄러 및 할당자는 사용자로부터 작업 요청이 들어왔을 때,





(그림 7) 클라우드 컴퓨팅 환경에서 가상화 기반 자원 관리 시스템의 구조[12]



(그림 8) 퍼지 모델링을 이용한 가상 데이터 센터 관리 시스템 구조 [13]

가상 머신의 자원 사용률과 가능한 가상 머신의 이동을 고려하여 작업의 수용과 거절을 결정한다.

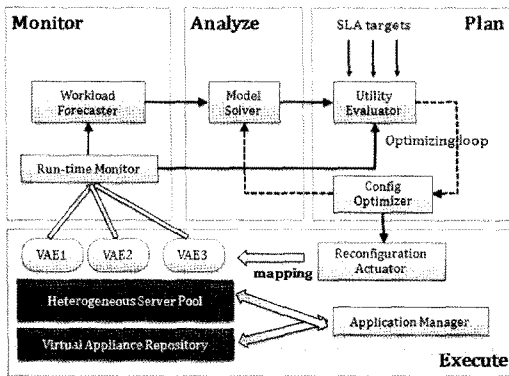
### 3.2.2 예측 기반 가상화된 데이터 센터 관리 기법

가상화된 컴퓨팅 환경은 사용자가 하나의 물리적 컴퓨터에 복수개의 가상의 컴퓨터를 운영할 수 있도록 해준다. 이러한 운영은 하나의 물리적 서버를 좀 더 효과적으로 활용하기 위해 사용된다. 특히나 이러한 가상화 환경은 각각의 가상화 컴퓨팅에 대한 정밀한 리소스 할당을 할 수 있기 때문에 이를 고려한 전체적인 가상 컴퓨터의 수와 운영 방법은 차세대 데이터 센터를 위한 핵심 기술로 떠오르고 있다. 앞서 언급한 가상화된 컴퓨터 환경에서 특별히 가상 컴퓨터 상의 응용은 다양할 수 있으므로 응용의 리소스 요구량에 맞추어 정확한 양의 리소스를 할당하는 것은 사용자의 QoS에 대한 요구 수준을 맞추고, 전체 데이터 센터의 운영 비용을 낮추기 위해 중요하다. 따라서 현재 모니터링 된 가상 자원의 상태

및 필요한 자원 할당량 예측을 통해 적절하게 자원을 할당하는 예측 기반의 자원 할당 기술들이 연구되고 있다[13-16].

(그림 8)은 퍼지(Fuzzy) 모델링을 이용한 가상 데이터 센터 관리 시스템 구조를 나타내고 있다. 이는 퍼지 이론을 기반으로 응용을 모델링하고 이에 따라 필요한 서버 자원 양을 예측하여 관리하는 시스템으로, 2 계층의 자원 관리 시스템을 이용함으로써 각각 응용의 관리 품질의 독립성을 보장하고 서버 운영 비용을 최소화하기 위한 자원 할당 시스템 구조를 가진다[13]. (그림 8)에서 볼 수 있듯이, 시스템은 크게 로컬 제어기(Local Controller)를 통해 관리되는 가상 컨테이너(Virtual Container)와 글로벌 제어기(Global Controller)를 통해 관리되는 데이터 센터 자원 풀(Resource Pool)로 나눌 수 있다. 로컬 제어기는 가상 컨테이너가 제공하는 응용의 작업량과 사용될 자원의 양을 예측하기 위해 응용의 성능을 만족하기 위한 작업량과 사용되는 자

원량 사이의 관계를 퍼지 모델을 이용하여 도출해 낸다. 도출된 정보와 함께 로컬 제어기는 글로벌 제어기에게 적절한 자원에 대한 요청을 하게 되고, 글로벌 제어기는 그 요청에 따라 데이터 센터가 보유하고 있는 자원들 중에서 가장 적은 비용의 것을 로컬 제어기에게 제공하는 역할을 한다. 이때 로컬 제어기는 각각의 응용을 위한 SLA를 최대한 만족시켜야 하며, 글로벌 제어기는 전체 시스템 운영자의 서비스 수준 합의를 최대한 만족시켜야 한다.

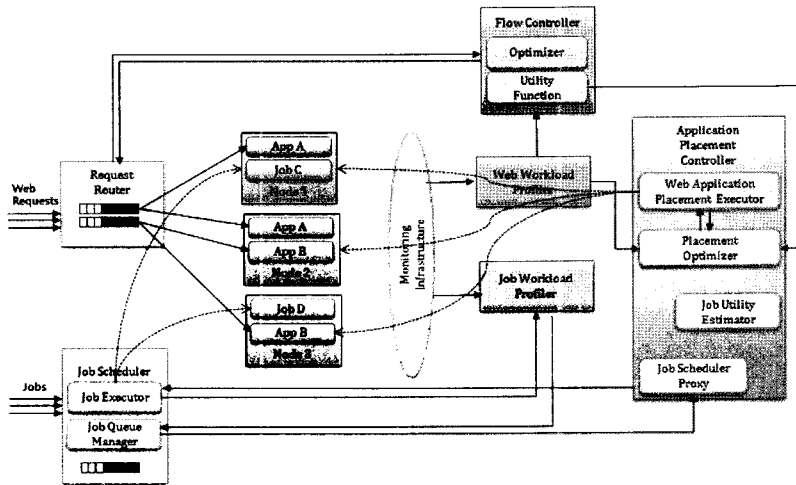


(그림 9) 비선형 기법 기반 관리 제어 구조 [14]

(그림 9)는 가상 데이터 센터 자원을 할당하는 시스템에서 비선형 예측 기법을 이용하여 자동적으로 현재의 응용 작업량을 분석하고, 이를 가상 서버들이 가진 자원들에 할당하기 위한 제어 구조를 나타내고 있다[14]. 가상 서버는 물리적 서버 위에서 운영되며, 각각이 한 가지 목적의 응용을 지원하기 위해 환경이 갖추어져 있다. 이러한 가상 서버는 같은 역할을 하는 다른 물리적 머신(Physical Machine)의 가상 서버와 그룹을 이루어 가상 응용 환경을 이루게 된다. 가상 응용 환경은 외부로부터의 응용 작업 요구를 처리하기 위한 가상 서버 그룹이며, 주문식 라우터(On-demand Router)를 통해 가상 응용 환경 내부에 존재하는 가상 서버의 수와 자원 할당량을 제어한다. (그림 9)에서 보듯이, 제어 시스템은

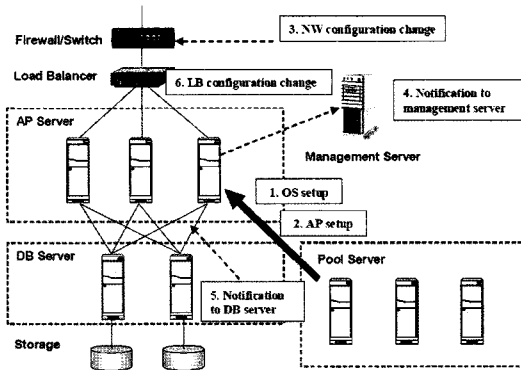
실제 서버 할당을 담당하는 실행(Execute) 파트, 할당된 서버의 현재 상태를 관찰하는 모니터(Monitor) 파트, 관찰된 현재 상태를 분석하는 분석(Analyze) 파트, 그리고 그러한 분석결과를 토대로 다음 번 자원 할당의 양을 정하는 계획(Plan) 파트로 나누어진다. 실행 파트의 이형 서버 풀(Heterogeneous Server Pool) 위에 가상 머신이 운영되고, 이들이 모여서 가상 응용 환경(Virtual Appliance Environment: VAE)를 이루게 된다. 이렇게 운영되는 VAE는 모니터 파트의 실시간 모니터를 통해 현재의 응용의 요청량과 작업 사용량을 분석하게 된다. 작업량 예측자(Workload Forecaster)가 이를 이용하여 미래 작업량의 변화를 예측하게 되고, 분석 파트와 계획 파트는 이러한 예측과 SLA를 기반으로 한 스케줄링 작업을 통해, 현재 VAE 내에 운영되고 있는 가상 머신의 양에 대한 재구성 요청을 실행 파트에 하게 된다. 이러한 제어 과정을 통해, 데이터 센터는 자원을 현 상태에 맞추어 최적으로 할 수 있다.

또 다른 데이터 센터 관리 기법으로서, (그림 10)은 특별히 다양한 응용의 작업 요구사항을 효과적으로 만족시키는 데이터 센터 관리 시스템을 제시하고 있다[15]. (그림 10)에서 보듯이, 우선 데이터 센터는 물리적 서버의 집합으로 이루어진 컴퓨팅 노드이다. 이러한 노드들 위에서 웹 서버와 같은 응용이 복수 개 위치하게 되며 이들이 응용 클러스터를 이루으로써 서비스가 수행되게 된다. 이 시스템은 작업의 종류를 짧고 상호작용적인 작업인 웹 요청 작업과 길고 상호작용하지 않는 작업으로 나누고 있으며, 웹 요청 작업은 플로우 제어 기술을 이용하여 대상을 응용, 서비스 등급에 따라 분리하여 가중 공정(Weighted-Fair) 스케줄링 알고리즘을 통해 최종 자원을 할당하게 되고, 길고 상호작용하지 않는 작업의 경우는 앞선 경우와 달리 작업 스케줄러에 입력으로 들어가 응용 배치 제어기



(그림 10) 응용 요구사항에 대한 관리를 위한 가상화 서버 관리 시스템 구조 [15]

(Application Placement Controller)를 이용하여 작업 목적에 맞는 스케줄링이 이루어지게 된다.



(그림 11) 가상화 서버 관리 시스템에서 서버 프로비저닝을 위한 웹서버의 구조 [16]

마지막으로, (그림 11)은 가상화된 서버에 부담을 주지 않으면서 정확하게 응용의 사용량을 예측할 수 있는 관리 시스템을 나타내고 있다 [16]. 전통적인 웹 서버 시스템이 고정된 수의 응용 서버를 유지한다고 가정할 때, 이 시스템은 응용 서버가 작업량이 늘어날 때 이를 보완하기 위해서 가상의 풀 서버를 두고 있다. 이러한 풀 서버를 이용한 프로비저닝 과정은 1) 운영체제 설치, 2) 응용 설치, 3) 네트워크 설정 변경 4) 관

리 서버에 알림 5) DB 서버에 알림, 그리고 최종적으로 6) 스위치에 변경사항을 알림으로써 이루어지게 되고, 이 과정에 걸리는 시간이 얼마나 단축될 수 있는가가 주요 문제라고 할 수 있다. 따라서 이 시스템은 미리 관련 환경이 설정되어 있는 대기 가상 머신을 도입하고 동적으로 현재의 시스템 성능을 모니터링 하는 동시에, 다음의 응용 작업량 예측을 위해 논리적 회귀 (Logistic Regression) 모델 기반의 응용의 요구량에 대한 예측기법을 사용함으로써 보다 짧은 프로비저닝 시간을 제공할 수 있다.

#### 4. 결론

본문에서는 차세대 컴퓨팅 패러다임으로써 시스템 및 애플리케이션의 추상화와 편리한 사용자 접근성, 유틸리티 컴퓨팅 (Utility, On-Demand Computing) 실현 등이 가능한 클라우드 컴퓨팅에 대해 설명하였다. 이러한 클라우드 환경은 사용자/소비자와 브로커/자원관리자, 자원제공자로 그 역할이 나누어지는 구조를 가지고 있으며, 사용자는 웹 인터페이스를 통해서 클라우드에 접근할 수 있고, 자원 제공자에 의해서 가상화된 자원들이 제공되어 그 가상화된 자원

들을 효율적으로 프로비저닝하고, 자원 관리 시스템에 의해서 보다 효과적으로 관리 (실시간 모니터링을 통한 자원 회복 기능 등) 하는 형태의 프레임워크를 제공한다. 또한 이러한 프레임워크를 실현하기 위해 분산 데이터 저장 기술과 분산 컴퓨팅 기술, 클러스터 관리 기술 등의 기술이 필요로 하며, 특히 자원들에 대한 프로비저닝과 자원 관리에의 필요성이 큰 클라우드 환경을 위한 다양한 자원 관리 기법들에 대한 연구가 필요하다. 따라서 본문에서는 다양한 연구가 진행 중인 클라우드 자원 관리 기술에서의 다양한 이슈들과 클라우드 자원 관리 구조에 대해 설명하였다. 제한된 컴퓨팅 자원에서 사용자들이 요청한 자원에 대한 다양한 요구사항들을 만족시키기 위해 동적인 가상 머신 기술을 사용할 때, 데이터 센터의 운영비용을 낮추고 사용자에게 필요한 만큼의 자원을 제공할 수 있는 서버 가상화 기술을 적용하게 되고, 이러한 상황에서 현재 모니터링 된 가상 자원의 상태 및 필요한 자원 할당량을 정확하게 예측하여 자원을 적절히 할당하는 다양한 예측 기반의 자원 할당 기술들을 설명하였다.

**참고문헌**

[1] D. A. Bader and R. Pennington, "Cluster Computing: Applications", The International Journal of High Performance Computing, 15(2):181-185, May 2001.

[2] I. Foster and C. kesselman, "The Grid: Blueprint for a Future Computing Infrastructure", Morgan Kaufmann, San Francisco, USA, 1999.

[3] D. S. Milojicic, et al. "Peer-to-Peer Computing", Tech Report: HPS-2002-57, available on line at: <http://www.hpl.hp.com/techreports/2002/HPL-2002-57.pdf>.

[4] M. P. Papazoglou, et. al, "Service-Oriented Computing Research Roadmap", European Union Information Society Technologies (IST), Directorate D - Software Technologies (ST) 2006.

[5] Greg Boss, et al, "Cloud Computing". IBM documents, <http://www.ibm.com>

[6] Ian Foster, et al. "Cloud Computing and Grid Computing 360-Degree Compared", Grid Computing Environments Workshop, 2008 (GCE '08)

[7] Rajkumar Buyya, et al. "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities", Proc. Of the 10<sup>th</sup> IEEE International Conference on High Performance Computing and Communications (HPCC08), Sept. 25-27, 2008.

[8] M Tsugawa, J. A. B. Fortes, "A Virtual Network (ViNe) Architecture for Grid Computing", In Proc. of 20<sup>th</sup> International Parallel and Distributed Processing Symposium (IPDPS2006), Rhodes Island, Greece

[9] Science Cloud, <http://workspace.globus.org/clouds>

[10] K. Keahey, et al, "Virtual Workspaces: Achieving Quality of Service and Quality of Life in the Grid", Scientific Programming Journal 2006

[11] Amazon Elastic Compute Cloud, <http://aws.amazon.com/ec2>

[12] Sven Granpner, Jim Pruyne, Sharad singhal, "Making the Utility Data Center A

Power Station for the Enterprise Grid”, HP Labs, 2003

- [13] Jing Xu, et al. “On the Use of Fuzzy Modeling in Virtualized Data Center Management”, Fourth International Conference on Autonomic Computing (ICAC’ 07)
- [14] Xiao Ying WANG, et al. “Appliance-based Autonomic Provisioning Framework for Virtualized Outsourcing Data Center”, Fourth International Conference on Autonomic Computing (ICAC’07)
- [15] Malgorzata Steinder, “Server virtualization in Autonomous Management of Heterogeneous workloads”, ACM SIGOPS Operation System Review, Vol42, Issue 1, Jan 2008
- [16] Fumio Machida, et al. “Just-in-Time Provisioning using Virtual Machine Standby and Request Prediction”, International Conference on Autonomic Computing.

**저자약력**



**송 예 원**

2002년 경북대학교 전자공학과 졸업  
 2005년 한국정보통신대학교 공학석사 취득  
 2009년~현재 KAIST 정보통신공학과 박사과정 재학 중  
 관심분야 : 그리드 컴퓨팅에서 자원 관리 이슈와 유틸리티 컴퓨팅, 클라우드 컴퓨팅 등



**김 희 영**

2007년 중앙대학교 컴퓨터공학과 졸업  
 2009년~현재 KAIST 정보통신공학과 석사과정 재학 중  
 관심분야 : 그리드 컴퓨팅, e-Health 서비스 시스템 등



**윤 잔 연**

1981년 경북대학교 전자공학과 졸업, 육군통신장교로 근무  
 1994년 일본 동북대학(Tohoku Univ)에서 공학박사를 취득  
 1986년 KT 연구소에서 CSMS, MOVE 등 전자교환기 관리 시스템 개발 담당  
 1995년 KT 통신망연구소 연구팀장으로, 초고속통신망 구축, KT-KDD ATM 상용망구축, APII-Testbed구축, 초고속 국가망구축 등 고성능 네트워크 구축 연구 개발 및 상용기술 개발에 참여  
 1997년 12월~2009년 2월 ICU 공학부 교수로 재직, 기획 처장(2006년), 산학협력단장(2006년) 역임  
 2003년~2004년 MIT 방문 교수 및 MIT-HST객원 연구원  
 2004년~현재 Grid Forum Korea 부의장  
 2009년 3월~KAIST 교수, 그리드 미들웨어 연구센터장  
 2009년~현재 정보처리학회 논문편집 총괄위원장  
 관심분야 : 고성능 네트워킹 서비스 구조 및 관리, 통신 미들웨어, 그리드 컴퓨팅, 클라우드 컴퓨팅, e-Health 서비스 시스템 등