

클라우드 컴퓨팅을 위한 가상화 기술

- 김건우(한양대학교 컴퓨터공학과)
- 이원주(인하공업전문대학 컴퓨터정보과)
- 전창호(한양대학교 컴퓨터공학과)

I. 서론

최근 IT 기술의 집약적인 발전에 따라 대용량 컴퓨팅에 대한 관심이 증폭되고 있다. 1960년대 분산처리 시스템을 시작으로 병렬처리 시스템, 그리드 시스템, 클라우드 시스템으로의 발전이 이루어져 왔다. 그 중 클라우드 컴퓨팅은 그리드 컴퓨팅의 뒤를 이어 지금까지 연구되었던 대용량 컴퓨팅의 핵심 기술의 집약체 라고 할 수 있다. 클라우드 컴퓨팅이란 인터넷 상의 서로 다른 물리적인 위치에 존재하는 각종 컴퓨팅 자원들을 가상화 기술로 통합하여 사용자에게 언제 어디서나 필요한 양만큼 편리하고 저렴하게 사용할 수 있는 환경을 제공하는 기술이다. 즉, 개인 컴퓨터 또는 개개의 응용 서버가 컴퓨터들의 구름(Cloud of computers)으로 옮겨간 형태로 개인용 컴퓨터나 기업의 서버에 개별적으로 저장해 두었던 모든 자료와 소프트웨어를 클라우드 내의 컴퓨터에 저장해 놓고, 인터넷 접속이 가능한 컴퓨터나 모바일 기기 등을 이용해 언제 어디서나 원하는 작업을 수행할 수 있는 차세대 사용자 환경이다[1][2].

클라우드 컴퓨팅이란 용어는 Software as a Service, Utility Computing, Grid Computing, Distributed Computing, Virtualization, Service Oriented Architecture, High Performance Computing 등의 개념을 이용하여 다수의 서버를 연동하여 실현할 수 있는 다양한 기술 개념을 통칭하는 용어로 등장하였기 때문에, 지금까지 그 어떤 용어보다도 뜻하는 바가 포괄적이며, 그에 따라 기술 집단이나 사업자

별로 각자의 관심사나 이해관계에 맞추어 수많은 자의적 해석이 가능하다[1]. 이러한 포괄적인 의미에서 클라우드 환경을 완성하고, 분산된 자원을 사용자 환경의 영향 없이 통합하기 위해서는 가상화 기술이 필요 하다.

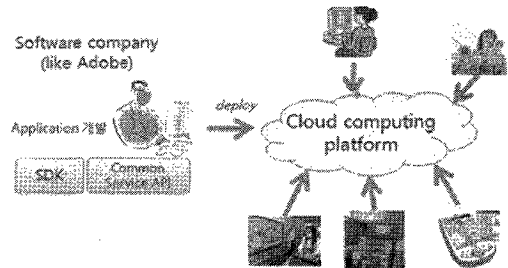


그림 1. 클라우드 컴퓨팅 구성도

본 글에서는 가상화의 개념, 발전과정, 분류에 대해 살펴보고 그리드 컴퓨팅과 클라우드 컴퓨팅의 가상화에 대해 살펴본다.

II. 가상화의 개념 및 효과

1. 개념

가상화란 물리적인 한 개의 자원을 논리적으로 분할해 효율적으로 사용하거나, 물리적으로 다른 여러 개의 자원을 논

리적으로 통합하는 기술이다. 가상화 기술은 논리적 자원들과 실제 물리적 자원들 사이에 위치하여 서로 연결을 해 줌으로써 가상화 자원을 이용하는 사용자는 더 이상 어떤 자원들이 사용되는지를 구체적으로 이해할 필요가 없어진다[3].

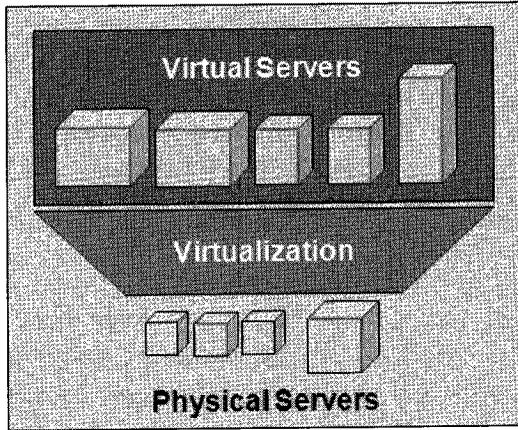


그림 2. 가상화의 구성

2. 효과

가상화 기술을 이용하게 되면 다음과 같은 이점을 얻을 수 있다[3].

• 높아진 자원 활용률

물리적 자원들과 자원 풀에 대한 동적인 공유가 가능하여 높은 자원을 활용률을 얻을 수 있다. 특히 평균 워크로드가 전체 자원의 워크로드보다 훨씬 적은 가변적인 워크로드 상황에서 더 높은 효과를 얻을 수 있다.

• 낮은 관리 비용

서비스 제공자의 인프라에 가상화 기술을 적용하게 되면 유휴 자원의 활용이 가능하게 됨에 따라 물리적 자원을 줄일 수 있고 이를 관리하는 인력을 줄일 수 있다. 또한 물리적 자원의 복잡성을 줄여줄 수 있으며, 관리 작업의 단순화 및 워크로드 관리의 자동화를 가능하게 해 준다.

• 사용의 유연성

가상화 기술이 적용된 인프라의 자원들이 자동으로 재구성되고 활용될 수 있도록 해준다.

• 향상된 보안

가상화 기술은 기존의 데이터를 공유하는 방법에서는 불가능한 분리와 격리를 가능하게 해 주어 가상화 기술이 적용된 인프라를 이용하는 사용자의 데이터와 서비스에 대해서 통제되고 안전한 접근을 가능하게 해준다.

• 높아진 확장성

가상화는 자원의 분할(Partitioning) 및 단일화(Aggregation)가 가능하여 물리적 자원 구성의 변경 없이 사용자가 요구하는 자원의 크기를 조절 할 수 있다.

3. 기능

가상화의 기술을 기능에 따라 분류해 보면 자원의 공유(Sharing), 풀링(Pooling), 치환(Emulation), 절연(Insulation)이라는 네 가지 형태로 나눌 수 있다.

• 공유(Sharing)

가상화의 가장 대표적인 기능으로 하나의 물리적 자원을 다수의 가상 자원으로 분할하는 기능이다.

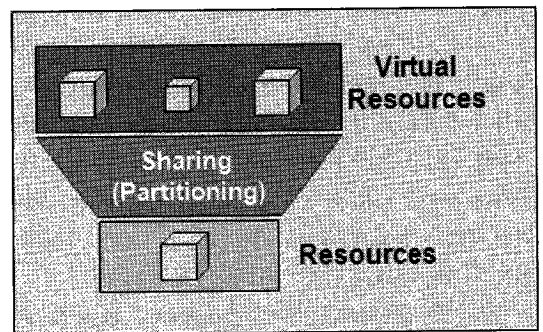


그림 3. 공유(Sharing)

- 풀링(Pooling)

공유와는 반대되는 개념으로 여러개의 물리적 자원을 하나의 가상 자원으로 통합하는 기능이다.

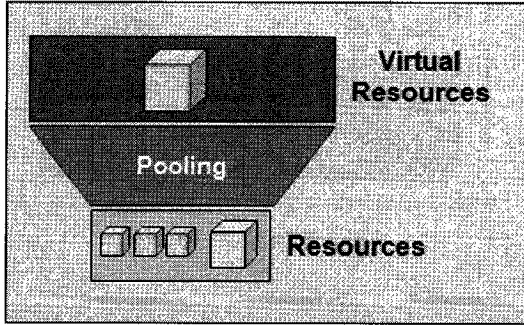


그림 4. 풀링(Pooling)

- 치환(Emulation)

물리적 자원에는 존재하지 않지만 필요에 따라 가상의 자원으로 어떤 기능이나 특성을 만드는 기능이다. Emulators, iSCSI, Virtual Tape 등이 대표적인 예이다.

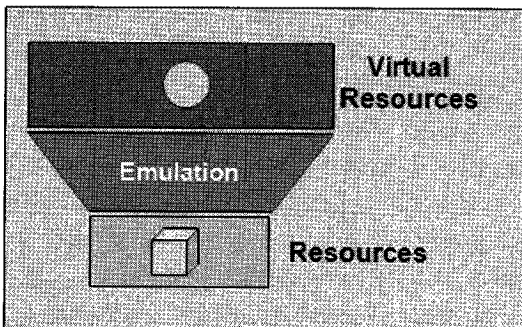


그림 5. 치환(Emulation)

- 절연(Insulation)

가상화된 자원에서 어떠한 작업이 처리되고 있을 때, 물리적 장애가 발생하였을 경우 작업에 영향을 주지 않고 물리적 자원을 교체하는 장애방지 기능이다.

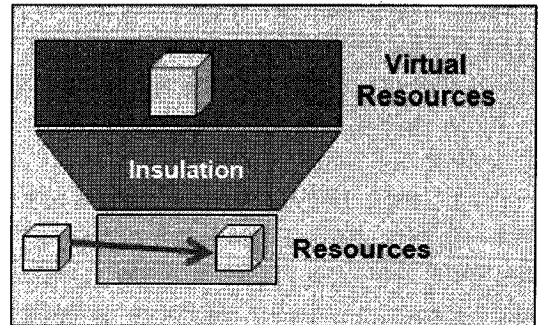


그림 6. 절연(Insulation)

III. 가상화의 발전과정

1. 등장배경

1961년부터 시작된 Time-Sharing System의 발전으로 하나의 물리적 자원을 여러 클라이언트가 공유하여 사용하는 것이 가능해 졌다. 하지만 Time-Sharing System의 자원은 사용자 입장에서 완전한 독립된 자원이 아니었기 때문에 사용자별 자원의 독립을 위해 1964년 'Virtual Machine'라는 시뮬레이트된 컴퓨팅 환경의 개념으로 IBM VM/360 시스템에 처음 등장하였다[4][5].

2. 발전과정

초기 가상화에 대한 연구는 플랫폼 가상화와 리소스 가상화 중심으로 이루어졌다.

플랫폼 가상화란 단일 하드웨어 플랫폼 위에 통합된 호스트 소프트웨어를 통해 다수의 게스트 소프트웨어를 실행시키는 방법이다. 하드웨어에 의존적인 소프트웨어를 실행시키는데 있어서 이질적인 환경에서도 가상화 레이어 상위계층에서 소프트웨어를 실행시킬 수 있는 장점이 있다. 하지만 호스트 하드웨어의 물리적 리소스의 제한을 받으며, 게스트 소프트웨어의 수 만큼 물리적 장치가 필요한 단점이 있다 [6]. 현재 JAVA Virtual Machine이 플랫폼 가상화의 대표적인 예이다.

리소스가상화란 특정한 물리적 시스템 리소스를 다수의 게스트 소프트웨어에게 제공하기 위해 논리적 리소스로 만들어주는 방법이다. 초기 가상메모리에 대한 연구를 시작으로 RAID, 논리볼륨, 저장장치가상화, VPN, NAT 등 다양한 분야에서 연구가 이루어져 왔다.

인터넷의 발달로 1990년대 중반 서버 가상화에 대한 연구가 시작되었다. 기존의 플랫폼 가상화와 리소스 가상화는 특정 물리적 자원에 대한 가상화인 반면 서버 가상화는 플랫폼 가상화와 리소스 가상화를 결합하여 다수의 물리적 자원을 완전한 논리적 가상화로 구현한 방법이다. 서버 가상화는 Virtualizing-Down Model 과 Virtualizing-Up Model 로 분류된다.

Virtualizing-Down Model은 단일의 물리적 자원을 다수의 논리적 자원으로 만드는 방법이다. 크게 서버프로세서 파티셔닝 방식과 하이퍼바이저 파티셔닝 방식으로 분류한다. 서버프로세서 파티셔닝방식은 한 개의 프로세서 상에 다수의 독립적 파티션을 구성하여 사용자가 필요로 하는 CPU 자원을 나누어 줄 수 있는 방법이다. 서버프로세서 파티셔닝 방식은 CPU의 자원을 분할하여 가상화를 구현하는 방법이기에 때문에 물리적 파티셔닝이라고도 하며, 자원을 분할하는데 프로세서의 수에 제약을 받는다.

하이퍼바이저 파티셔닝 방식은 물리적 서버 상위계층에 하이퍼바이저 계층을 두어 물리적 자원에 제약을 받지 않고 각 게스트 시스템에 자원을 할당하는 방식이다. 물리적 자원을 하이퍼바이저 계층을 통하여 완전한 논리적 자원으로 만들어지기 때문에 논리적 파티셔닝이라고도 한다.

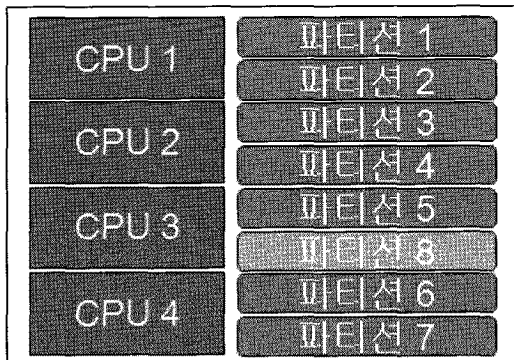


그림 7. 서버프로세서 파티셔닝

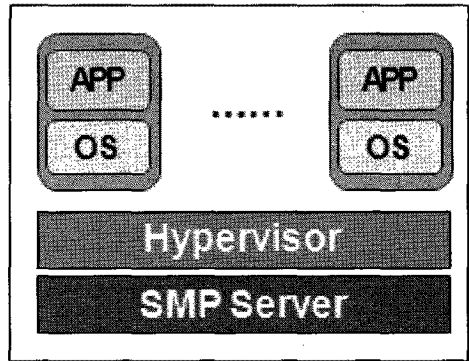


그림 8. 하이퍼바이저 파티셔닝

하이퍼바이저 파티셔닝 방식은 하이퍼바이저의 위치에 따라 베어메탈(Bare-metal) 하이퍼바이저와 호스트기반 하이퍼바이저로 분류할 수 있다. 이 중 베어메탈 하이퍼바이저 방식은 물리적 서버의 상위계층에 최소한의 O/S 기능을 가진 커널과 하이퍼바이저 계층을 둔 방식이며, 자원 관리에 따라 완전가상화와 반가상화로 분류할 수 있다. 완전가상화는 CPU, 메모리, 네트워크 장치를 비롯한 모든 자원을 하이퍼바이저에서 관리를 하기 때문에 가상화된 자원의 관리 및 운용이 쉽다는 장점이 있지만 워크로드의 부담이 발생되며 가상화된 자원을 사용하는 게스트 O/S가 자원 제어를 할 수 없다는 단점이 있다. 반 가상화는 CPU, 메모리 자원만 하이퍼바이저에서 관리를 하고 나머지 모든 자원은 리소스 가상화의 형태로 게스트 O/S에서 제어를 하는 방식이다. 따라서 게스트 O/S는 자신이 원하는 자원을 적절하게 사용할 수 있지만 하이퍼바이저에 맞게 O/S를 수정해야하는 단점이 있다. 표 1은 완전 가상화와 반 가상화의 장단점을 비교해 놓은 것이다.

표 1. 완전가상화와 반가상화의 비교

항목	장점	단점
완전가상화	게스트 O/S 설치에 제약이 없음	워크로드 부담 발생
반가상화	게스트 O/S가 자원의 직접 제어권을 가짐	게스트 O/S의 수정 필요

호스트기반 하이퍼바이저 방식은 그림 9와 같이 물리적 서버의 상위계층에 호스트 O/S를 설치하고, 호스트 O/S에 하이퍼바이저 어플리케이션을 설치하여 가상화 환경을 만드는 방법이다. 사용자가 간단하게 테스트 환경을 구현할 수 있다는 장점이 있지만 물리적 자원을 호스트 O/S가 대부분 사용하고 있기 때문에 성능과 자원관리가 베어메탈 하이퍼바이저 파티셔닝 방법에 비해 떨어지는 단점이 있다.

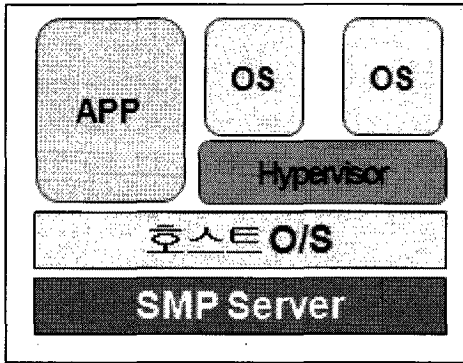


그림 9. 호스트기반 파티셔닝

표 2. 서버프로세서와 하이퍼바이저 파티셔닝의 비교

항목	장점	단점
서버프로세서	O/S사이의 격리가능	파티션수 제약, 유연한 파티션 구성의 어려움
하이퍼바이저	유연한 파티션 구성 가능, 손쉬운 자원 세분화	하이퍼바이저 종류에 따른 성능 차이

Virtualizing-Up Model은 다수의 물리적인 자원을 하나의 논리적 자원으로 통합하여 구현한 모델이다.

Virtualizing-Down Model에서 유희자원의 활용을 극대화했다면 Virtualizing-Up Model에서는 성능의 극대화를 목표로 한다. Virtualizing-Up Model 은 하이퍼바이저 방식과 모듈러 방식으로 분류한다.

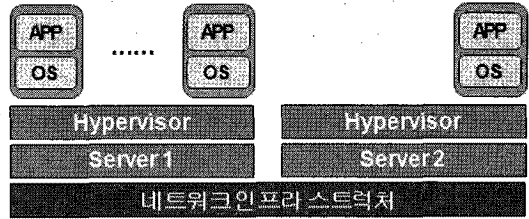


그림 10. 하이퍼바이저 Virtualizing-Up Model

Virtualizing-Up Model에서의 하이퍼바이저 방식은 각 물리적 서버에 하이퍼바이저를 탑재하여 자원을 통합하는 방식이다. 하지만 기술적 한계 때문에 모든 자원을 하나로 통합하지 못하고 게스트 O/S의 이주(Migration)정책을 통하여 작업을 처리한다. 그림 10은 Virtualizing-Up Model에서의 하이퍼바이저의 모습이다. 만약 Server 1에서 다수의 게스트 O/S가 작동을 하던 중 대용량 작업이 필요한 게스트 O/S가 본 시스템에 참여를 하게 되면 Server 1의 하이퍼바이저에서 작동 중인 게스트 O/S는 유희자원이 있는 Server 2의 하이퍼바이저로 이동하게 된다.

모듈러 방식은 하드웨어적으로 여러 대의 서버를 외부 케이블로 연결하여 하나의 큰 자원으로 통합 하는 방식으로 하이퍼바이저 방식에서의 자원 통합에 대한 한계를 극복할 수 있는 방식이다. 하지만 자원의 확장을 위해서는 전체 시스템을 정지시키고 물리적인 작업을 해야 하며, 구성이 어렵고, 비용이 비싸다는 단점이 있다.

2000년대 초반 그리드 컴퓨팅이라는 개념이 발생됨에 따라 새로운 가상화 기술에 대한 연구가 진행 되었다. 통합된 자원을 여러 사용자에게 나누어 주기 위해서는 기존의 Virtualizing-Down Model 기술과 Virtualizing-Up Model 기술의 통합이 필요했다. 두 모델을 통합하여 연구된 가상화 기술이 레이어드 가상화이다. Virtualizing-Up and Down Model이라고도 불리는 레이어드 가상화는 물리적 리소스를 하나의 논리적 리소스로 통합한 다음 자원 이용률에 따라 통합된 논리적 리소스를 Virtual Machine에 분배하는 모델이다. 그림 11은 레이어드 가상화의 모습이다. Virtualizing-Up Model에서 자원 통합의 문제를 해결하기 위해 각 물리적 서버위에 하이퍼바이저 기능을 하는 Virtual O/S를 두고, 상위 계층에서 Virtual Server을 두어 운용하는 방식이다. Virtual

O/S는 반 가상화 방식으로 자원을 운용하기 때문에 Virtual Server 운용에 맞게 물리적 자원의 영향을 받지 않고 자원 분배를 할 수 있다.

가상화의 발전 과정을 살펴보았을 때 가상화의 개념은 하

나의 자원을 내부적으로 잘게 쪼개는 파티션 기술에서 물리적으로 다수의 자원을 하나의 자원인 것처럼 보이게 하는 방향으로 변화가 일어나고 있다. 이러한 개념은 클라우드 컴퓨팅 환경에서 꼭 필요한 개념이다.

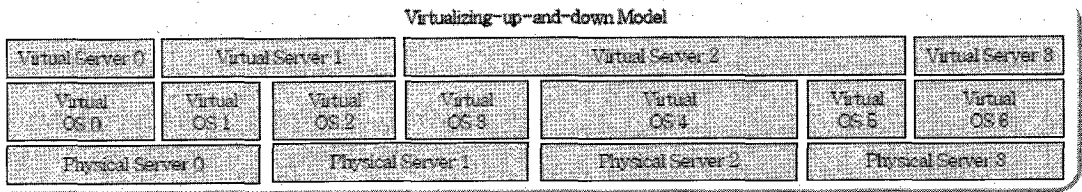


그림 11. 레이어드 가상화

IV. 가상화의 분류

현재의 가상화는 다양한 종류의 물리적 자원을 가상 자원으로 만들 수 있다. 여러 분야에서 가상화에 대한 연구가 진행됨에 따라 가상화의 종류와 개념이 많아 졌다. 다음은 일반적으로 분류되는 스토리지, 서버, 네트워크 및 서비스 가상화에 대한 설명이다[7].

1. 스토리지 가상화

스토리지 가상화는 여러대의 스토리지를 하나의 스토리지 저장 풀로 만들어서 여러 기기에 걸친 볼륨 구성, 기간 자유로운 데이터 마이그레이션 및 볼륨 복제, 데이터 공유등을 가능하게 하는 기술 이다. 스토리지 가상화의 가장 큰 이점은 복잡한 스토리지 구성을 논리적 스토리지로 단순화하여 통합 관리 통해 관리의 효율성을 높일 수 있고 스토리지의 확장 및 장애 발생 시 전체 시스템을 정지하지 않고 수정하기 때문에 서버의 가용시간을 늘릴 수 있다.

2. 서버 가상화

서버 가상화는 하나의 서버의 운영체제에 여러 개의 가상머신을 만들어 유휴 자원의 활용을 높이는 기술이다. 일

반적으로 서버 가상화는 CPU, 메모리, 입출력장치 등의 다양한 자원을 하나로 통합하여 서버의 기능 및 필요 자원에 따라 물리적 서버 위에서 작동하기 때문에 여러 대의 서버를 운용하는 것에 비해 관리가 쉽고 유지비용이 적다는 장점이 있다[8].

3. 네트워크 가상화

네트워크 가상화는 여러 사용자들을 네트워크를 통해서 하나의 공동된 가상 환경으로 묶어주는 기술이다. 네트워크에서 공유되어 사용할 수 있는 자원은 라우터, 스위치, 로드 밸런서, 방화벽 등이 있다.

4. 서비스 가상화

서비스 가상화는 스토리지 가상화, 서버 가상화, 네트워크 가상화의 기술을 통합하여 서비스 중심의 컴퓨팅 동작이 가능하게 만드는 기술이다. 그리드 컴퓨팅이나 클라우드 컴퓨팅에 적용되어야할 가상화의 일반적인 모습이다.

V. 그리드와 클라우드 컴퓨팅의 가상화

1. 그리드 컴퓨팅

그리드 컴퓨팅은 지리적으로 분산된 다양한 컴퓨팅 자원들을 네트워크로 연결하여 상호 공유할 수 있는 컴퓨팅 환경이다. 2000년 초반 그리드 컴퓨팅에서는 자원의 분산을 위해 서비스 가상화가 아닌 스토리지 가상화, 서버 가상화, 네트워크 가상화의 기술을 이용하여 상호 연결하는 형태로 시작 되었다. 하지만 여러 가지의 기술을 연결하여 서비스를 제공하는 형태는 보안문제와 자원관리의 문제점을 가지고 있었다. 스토리지 가상화 기술을 이용한 저장 공간의 공유는 개인정보 유출에 대한 보안문제가 발생하였으며, 그리드 시스템 내에서 Legacy 코드의 사용에 의해 보안 취약점이 발생하게 되면 전체 시스템에 영향을 주는 문제가 발생하였다. 또한 자원관리에 있어서 한 사용자의 자원 독점 문제가 발생 되었으며, 각 사용자들에게 QoS를 보장하지 못하는 문제점도 발생 되었다.

이러한 문제점들로 인하여 2003년 Figueiredo RJO와 Dinda PA, Fortes JAB는 그리드 컴퓨팅환경에서의 통합된 가상화 기술을 적용하는 방법을 제안했다. 제안된 방법은 그리드 자원을 사용자에게 분배할 시에 VM(Virtual Machine)을 이용하여 분배하는 방법이다[9]. VM을 이용한 자원분배 방식을 적용하게 되면 다음과 같은 이점을 얻을 수 있다.

- Isolation

그리드 자원을 사용하는 사용자는 VM 이라는 범주 안에서 자신만의 자원을 사용할 수 있기 때문에 다른 사용자로부터 자신의 데이터를 보호할 수 있다.

- On-Demand Creation

사용자가 그리드 자원을 사용하기 위해 시스템에 참여할 시에 가용 가상 자원의 확인만으로 VM을 생성할 수 있다.

- Customization

VM이 생성될 때, 사용자가 원하는 환경을 만들 수 있다. 예를 들어, 필요한 CPU 및 메모리 자원이나 사용할 어플리케이션의 종류등을 고려하여 VM을 생성할 수 있다.

- Legacy System Support

기존의 그리드는 시스템 내에서 보안에 취약한 Legacy Code가 실행되었을 시 시스템 전체에 문제를 발생시켰다. 하지만 VM을 이용함으로써 VM내에서 보안이 취약한 Legacy Code가 실행되게 되면 해당 VM의 시스템만이 문제가 발생되며 시스템 전체에는 문제를 발생시키지 않는다.

- Administration Privileges

시스템 자원을 사용하는 사용자는 VM을 이용하기 때문에 자신의 환경에 대한 관리 권한을 가질 수 있다.

2. 클라우드 컴퓨팅

그리드 컴퓨팅에 이어 사람들은 클라우드 컴퓨팅에 많은 관심을 가지고 있다. 그리드 컴퓨팅에 비해 클라우드 컴퓨팅의 가장 큰 장점은 서비스를 사용하는 사용자가 시스템에 대한 전반적인 지식 없이 시스템을 사용할 수 있다는 것이다. 또한 클라우드 컴퓨팅에서는 SLA(Service Level Agreement) Resource Allocator를 두어 서비스 제공자와 사용자 사이의 서비스 수준을 정량적으로 측정하여 특정 범위 내에서 일정 수준의 서비스 제공을 보장하고, 그에 따른 과금 시스템이 적용 된다. 그리드 컴퓨팅에서와 같이 모든 리소스들은 가상화된 자원을 제공하지만 사용한 VM 자원에 대한 요금만을 지불하는 방법은 특정 사용 기간에 대한 요금을 지불하는 방식인 그리드 컴퓨팅과는 다른 점이다.

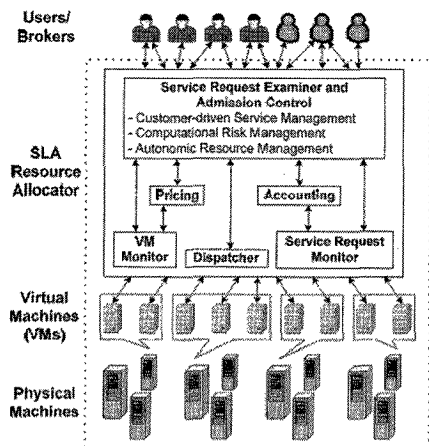


그림 12. 클라우드 컴퓨팅의 구조

그림 12는 클라우드 컴퓨팅의 구조이다[10]. 물리적 자원을 가상 자원으로 만들어 주는 Virtual Machines 라는 가상화 계층을 가지고 있으며, SLA Resource Allocator 계층에서 또한 VM의 사용량을 측정하는 VM Monitor이 존재한다. 그리드 컴퓨팅의 가상화 계층과 클라우드 컴퓨팅의 가상화 계층은 큰 차이점이 없어 보이지만 그리드 컴퓨팅에서의 가상화 기능에 과금시스템을 위한 가상화 기술이 추가 된 것을 확인할 수 있다. 또한 내부적으로 살펴 보았을 때, 모든 사용자에게 QoS를 보장해야하기 때문에 VM에 문제가 생겼거나 확장이 필요할시 VM의 데이터를 다른 VM으로 안전하고 빠르게 이전시키는 가상화 기법이 필요하다.

VI. 결론

클라우드 컴퓨팅은 지금까지 연구되었던 고성능 컴퓨팅 분야의 다양한 기술의 혼합으로 개념이 포괄적이며, 이해하는 사람의 관점에 따라 해석이 달라질 수 있다. 고성능 컴퓨팅의 다양한 기술 중 가상화 기술은 클라우드 컴퓨팅에서는 빠져서는 안 될 핵심 기술이다. 이미 그리드 컴퓨팅 분야에서 수많은 가상화 기술들이 연구되어 왔지만 그리드 컴퓨팅과 클라우드 컴퓨팅의 서비스 방법과 구조가 다르기 때문에 클라우드 컴퓨팅 환경에 맞는 가상화 기법이 필요하다.

참고문헌

- [1] 이종숙, 박형우(한국과학기술정보연구원) “국내의 클라우드 컴퓨팅 동향 및 전망”, 한국정보처리학회지, 2009년 3월호 특집.
- [2] 최성, “클라우드컴퓨팅 서비스 플랫폼 기술 동향”, 정보통신산업진흥원 주간기술동향, 제 1437호, 27-41쪽, 2010년 3월.
- [3] 한국 IBM 시스템테크놀로지그룹, “가상화 기술의 새로운 패러다임”, 한국경제신문, 2007년.
- [4] Robert. P. Goldberg, “Survey of Virtual machine research”, IEEE Computer Magazine, Vol. 7, No. 6, pp. 34-45, 1974.
- [5] R. J. Creasy, “The Origin of the VM/370 Time-Sharing System, IBM Journal of Research and Development, Vol. 25, No. 5, pp. 483-490, Sept. 1981.
- [6] R.A. MacKinnon, “The changing virtual machine environment: Interfaces to real hardware, virtual hardware, and other virtual machines”, IBM System Journal, Vol. 18, No. 1, pp.18-46, 1979.
- [7] 김진미, 배승조, 정영우, 삼규호, 고광원, 우영춘, “유틸리티 컴퓨팅 시대를 여는 가상화 기술 동향”, 주간기술동향, 통권 1208호, 2005년 8월.
- [8] “The Future of Server Virtualization”, Garther Research, July 2003.
- [9] Figueiredo RJO, Dinda PA, Fortes JAB, “A Case for Grid Computing On Virtual Machine”, Proceedings of the 23rd international conference on distributed computing systems, pp. 550-559, 2003.
- [10] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, “Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as IT Services as Computing Utilities,” in Proc. 10th IEEE Int. Conference on High Performance Computing and Communications, HPCCC 2008, Sept. 2008.

저자 소개



김 건 우

2007: 백석대학교
정보통신학부 학사.
2010: 한양대학교
컴퓨터공학과 석사.
현 재: 한양대학교
컴퓨터공학과 박사과정.
관심분야: 클라우드컴퓨팅,
가상화



이 원 주

1989: 한양대학교
전자계산학과 공학사.
1991: 한양대학교
컴퓨터공학과 공학석사.
2004: 한양대학교
컴퓨터공학과 공학석사.
현 재: 인하공업전문대학
컴퓨터정보과 부교수.
관심분야: 병렬처리시스템,
성능분석, Grid컴퓨팅,
클라우드컴퓨팅.



전 창 호

1977: 한양대학교
전자공학과 학사.
1982: Cornell University
컴퓨터공학과 석사.
1986: Cornell University
컴퓨터공학과 석사.
1977-1979: 전자통신연구소
연구원.
현 재: 한양대학교
전자컴퓨터공학부 교수.
관심분야: 병렬처리시스템,
성능분석, Grid컴퓨팅,
클라우드컴퓨팅.