KNOM Review, Vol. 12, No. 2, Dec. 2009, pp. 40-48

가상화와 데이터 관리 기술들에 대한 확장성 및 신축성 평가

(Evaluation of Scalability and Elasticity on Technology of Virtualization and Data Management)

최영락*, 리건**, 홍원기*
* 포항공과대학교 정보전자융합공학부
** 포항공과대학교 컴퓨터공학과
{dkby, gunine, jwkhong}@postech.ac.kr

요 약

클라우드 컴퓨팅은 컴퓨팅에 사용되는 자원들을 확장성 및 신축성있게 관리하여 서비스를 제공하는 기업 및 사용자 모두에게 자원의 활용도를 높이는 이점을 제공한다. 본 논문의 목표는 클라우드 컴퓨팅에 필수적으로 적용되는 가상화와 데이터 관리 기술에 따라 확장성 및 신축성이 어떻게 달라지는지를 살펴보는 것에 있다. 이를 설명하기 위해 본 논문에서는 우선 클라우드 컴퓨팅의 정의와 함께 확장성 및 신축성이 클라우드 컴퓨팅에 어떻게 반영되는지 명확히 알아보고, 가상화 및 데이터 관리 기술이 클라우드 컴퓨팅의 확장성 및 신축성에 미치는 영향을 자세히 기술하고자 한다.

Keywords: Cloud Computing, Scalability, Elasticity, Virtualization, Data Management

1. 서론

컴퓨팅 속도 및 성능이 점차 향상되면서 컴퓨팅 환경 및 컴퓨팅에 사용되는 기술들도 점차 복잡해지고 있다. 과거에는 데이터를 가져오고 통신하는 데 있어 중앙 집중형 컴퓨팅(Centralized Computing)이 주요하게 이용되었다. 이후 컴퓨터 네트워크가 보급되고 확산되면서, 분산형 컴퓨팅(Distributed Computing)이 등장하여 보다 향상된 가용성(Availability)과 장애 허용성(Fault-tolerance), 그리고 확장성(Scalability)을 지원할 수 있게 되었다[1]. 이러한 분산형 컴퓨팅 방식은 1970년대에 그리드 컴퓨팅 기술의 출현에 영향을 주었고, 그리드 컴퓨팅 기술은 오늘날 유틸리티 컴퓨팅 및 클라우드 컴퓨팅의 탄생에도 영향을 주었다.

현재 수많은 기업들은 클라우드 컴퓨팅을 이미 도입하고 상품화하여 서비스를 제공하고 있다. 아마존은 Amazon EC2 서비스를 시작하였고, Google은 Google App. Engine을 홍보하고 있다. 또한, Microsoft는 자사가 개발한 서버 및 어플리케이션 기능들을 클라우드 컴퓨팅 환경을 통해 제공하는 Azure 서비스를 제공한다[2]. 현재 이러한 기업들은 클라우드 컴퓨팅의 주체인 사용자들을 관리하는 고민과 함께 사용자들이 점유하는 클라우드 컴퓨팅 내 자원들을 어떻게 관리할 것인지에 대한 고민을 갖고 있다. 특히 클라우드 컴퓨팅을 통해 사용자들에게 제공되는 자원의 양은 언제든지 변할 수 있는 특성을 지니고 있어, 기업들은 사용자들이 요구하는 자원의 양을 정확하게 제공하는 동시에, 사용자들의 자원에 대한 요구량이 급격하게 변하여도 이에 대응하여 처리할 수 있어야 한다. 이는 기업이 사용자들에게 제공하는 자원의 양에 대해 필요에 따라 늘이거나 줄일 수 있는 확장성 및 신축성이 고려되어야만 가능하다. 이러한 확장성 및 신축성은 특히 클라우드 컴퓨팅을 구성하는 데 있어 필수적인 기술 요소인 가상화와 데이터 관리에서 보다 명확히 드러난다.

본 논문의 구성은 다음과 같다. 2장에서는 클라우드 컴퓨팅과 확장성 및 신축성의 명확한 정의, 그리고

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업(NIPA-2010-C1090-0902-0045)과 한국과학재단을 통해 교육과학기술부의 WCU 사업(WCU, World Class University)으로부터 지원받아 수행되었습니다(과제번호: R31-2008-000-10100-0).

확장성과 신축성이 클라우드 컴퓨팅에서 중요하게 고려되는 이유에 대해 알아본다. 3장에서는 가상화를 이루는 주요 기술들이 클라우드 컴퓨팅의 확장성 및 신축성에 어떤 영향을 미치는지 살펴보고, 4장에서는 마찬가지로 데이터 관리에 관련된 구체적인 기술들이 미치는 영향을 살펴보고자 한다. 마지막으로 5장에서 결론을 맺는다.

2. 배경

이 장에서는 클라우드 컴퓨팅과 확장성 및 신축성에 대한 정의를 먼저 살펴본다. 이후, 확장성 및 신축성이 클라우드 컴퓨팅에서 중요하게 고려되는 이유를 Real-time Provisioning 사례를 통해 알아보고자 한다. 마지막으로, 클라우드 컴퓨팅의 핵심을 이루는 기술들인 가상화와 데이터 관리 기술에 대해 살펴본다.

2.1 클라우드 컴퓨팅의 정의

클라우드 컴퓨팅은 용어 자체에서 느낄 수 있는 것처럼 '구름을 이용하는, 아니면 구름 위에서 이루어지는 컴퓨팅인가?'와 같이 의미를 파악하기 쉽지 않다. 클라우드 컴퓨팅은 인터넷을 기반으로 하는 새로운 컴퓨팅 기술로, 마치 하늘에 있는 구름 속을 파악하기 어렵듯이 사용자 입장에서 클라우드 컴퓨팅 내부를 파악하기 어려운 것이 사실이다[3]. 미국 국립표준기술연구소(NIST: National Institute of Standards and Technology)에서는 클라우드 컴퓨팅을 '최소한의 관리 노력 또는 서비스 제공 기업의 상호작용을 통해 재빠르게 Provisioning되거나 제거할 수 있도록 구성 가능한 네트워크, 서버, 저장장치, 어플리케이션 및 서비스들과 같은 컴퓨팅 자원들의 공용 공간에 대해 실시간으로 네트워크 접근이 가능하도록 편리하게 구성한 모델'로 정의하고 있다[4].

2.2 확장성과 신축성

일반적으로, 확장성은 비즈니스 상에서 동작하는 어플리케이션의 특징을 나타내는 폭넓은 의미를 가진다. 이러한 확장성은 단순히 규모를 늘리는 것에 초점을 맞추는 단순한 확장성과 확장 및 축소가 가능한 신축성의 2가지로 나눌 수 있는데, 본 논문에서는 단순한 확장성을 확장성이라 규정하고, 이 단순한 확장성을 신축성과 구분 지어 설명하고자 한다.

2.2.1 확장성

보다 많은 어플리케이션들을 성능 저하 없이 동시에 실행하도록 지원하기 위해서는 하드웨어를 추가해야만 하는데, 어플리케이션의 인프라 구조 및 플랫폼에 따라 하드웨어를 추가한 이후 어플리케이션의 실행 환경이 달라지거나 재실행해야 하는 경우가 발생한다. (단순한) 확장성은 이와 같이 늘어나는 규모를 만족시키기 위한 능력에 초점을 맞추어, 어플리케이션을 재설계나 재배포하지 않고도 하나의 어플리케이션을 동시에 실행시킬 수 있는 능력과 밀접하게 관련되며 하드웨어를 통해 어플리케이션의 확장성을 지원하는 전략을 지원하도록 플랫폼이 설계되어야 한다. 만약 어플리케이션 및 인프라 구조가 해당 전략을 지원할 수 없도록 설계되면 실제 어플리케이션의 효율 및 처리량은 증가하지 않을 것이기 때문이다. 그리고 실시간 확장성을 지원하기 위해서는 어플리케이션 및 인프라 구조뿐만 아니라 하드웨어적인 기술들을 사용하여 플랫폼 및 어플리케이션에 대한 중단 과정 없이 어플리케이션이 계속 실행될 수 있도록 해야 한다.

2.2.2 신축성

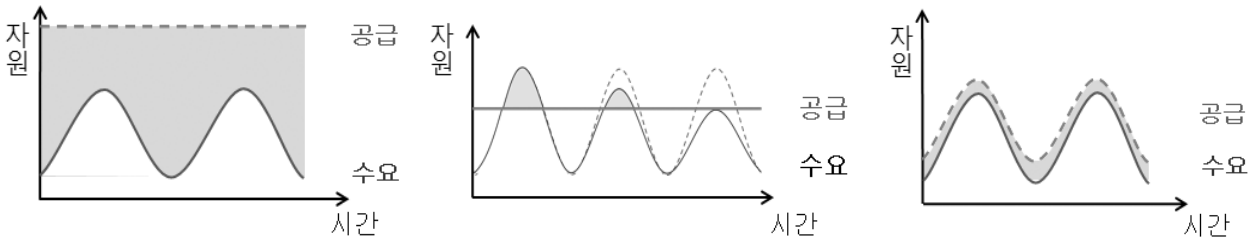
컴퓨팅 환경에서 신축성은 지속적으로 변하는 고객의 수요를 충족시키기 위해 컴퓨팅 능력을 증가하거나 축소시킬 수 있을 뿐만 아니라, 한 사용자의 수요가 감소했을 때 해제된 자원을 재사용 가능하게 하고, 이 때 실제 사용된 자원의 양만큼 과금 처리를 수행하는 기능을 의미한다[5]. 자동화된 신축성은 새롭게 바뀐 사용자의 자원 요구량이 시스템에 의해 자동적으로 감지되고 이에 따라 추가적인 컴퓨팅 자원이 해당 어플리케이션에 할당되거나, 컴퓨팅 자원이 더 이상 필요하지 않은 경우에는 해제되는 기능을 내포한다. 또한, 신축성은 이 때 제거되는 컴퓨팅 자원이 다른 어플리케이션에서 필요로 하는 경우 재사용이 가능하도록 조절하는 능력까지 포함한다. 이러한 신축성은 다수의 사용자가 한정된 양의 자원을 사용하는 경우 다수의 사용자들에게 모두 이익을 가져다 주는 이점을 제공한다. 단일 사용자가 자원을 이용하는 경우에는 사용하는 자원의 양을 증가시키거나 감소시켜도 증가 감소분에 대한 자원을 사용하는 사용자가 존재하지 않기에 자원 활용에 대한 이점을 제공하지 못하기 때문이다. 클라우드 컴퓨팅은 이러한 신축성을 지원하여 컴퓨팅 자원들을 경제적으로 사용할 수 있다는 이점을 제공한다. 또한 기업이 자동화된 신축성을 통해 사용된 사용자가 필요한 만큼의 자원의 양에 대해서만 과금을 수행하는 Pay-as-you-go 형태를 지원한다.

2.3 클라우드 컴퓨팅에서 확장성 및 신축성의 중요성

확장성 및 신축성은 클라우드 컴퓨팅에서 Provisioning을 보다 효율적으로 지원하는 데 영향을 미치는 요소이다. 클라우드 컴퓨팅 환경에서는 기존 컴퓨터들뿐만 아니라 모바일 장치, 노트북, PDA와 같은 보다 다양한 장치들이 네트워크를 통하여 컴퓨팅 자원에 접근할 수 있어야 한다. 다양한 장치들이 필요로 하는 서비스를 충족시키기 위해서는 향상된 확장성 및 신축성을 기반으로 Provisioning이 지원되어 요구하는 만큼의 자원을 제공하고, 더 이상 자원을 필요로 하지 않는 경우에는 해당 자원을 필요로 하는 다른 사용자에게 할당하여 클라우드 컴퓨팅 내 자원을 효율적으로 관리해야 한다. 클라우드 컴퓨팅은 이러한 확장성 및 신축성을 기반으로 컴퓨팅 자원들을 경제적으로 사용할 수 있다는 이점을 제공하는 동시에, 자동화된 신축성을 통해 기업들은 사용자가 필요한 만큼의 자원의 양에 대해서만 과금을 수행하는 Pay-as-you-go 형태를 지원한다.

이러한 Provisioning 특징은 클라우드 컴퓨팅의 Real-time Provisioning 사례를 통해 확장성과 신축성이 명확히 드러난다. Real-time Provisioning이란 사용자가 요구하는 자원의 양만큼을 기업이 제공하면서, 사용자가 요구하는 자원의 양이 변하여도 기업에서는 변화량에 맞게 동적으로 자원의 할당량을 실시간으로 변화시켜 사용자에게 해당 양만큼의 자원을 제공하는 방식을 의미한다. 이러한 Provisioning 방식은 기존의 Over-provisioning 및 Under-provisioning 방식과는 차별화된 특성을 지닌다.

그림 1(a)는 Over-provisioning 을, 그림1(b)는 Under-provisioning 전략을 사용할 때, 사용자 수요에 따른 자원과의 관계를 나타낸다[6]. Over-provisioning이나 Under-provisioning 전략을 사용할 때에는 사용자가 원하는 자원의 수요가 변하여도 기업 입장에서는 자원을 과도하게 또는 수요량보다 적게 공급하는 수밖에 없었다. 그 결과 자원이 더 많이 공급되는 경우 기업에서는 자원의 낭비에 따른 손실을 충당해야 하는 반면 적게 공급하는 경우 사용자의 초과되는 수요에 대해 추가적인 자원 Provisioning과 같은 별도의 대응책을 마련해야 하는 단점이 있었다. 또한 사용자 입장에서는 불필요한 자원이 과도하게 공급되는 경우 해당 양에 대한 과다 비용을 청구 받는다고 인식할 수 있으며, 자원이 적게 공급되는 경우에는 그 이상의 컴퓨팅 자원 필요로 하는 경우를 접할 수 있다는 단점을 가지고 있다. 반면 그림1(c)과 같이, Real-time Provisioning 전략은 기업과 사용자 모두에게 단점을 가져다 주었던 기존의 상황을 극복한다[6]. 사용자가 필요로 하는 자원의 양을 기업에게 알려주기만 하면, 기업은 사용자가 필요로 하는 자원의 양만큼만 준비하여 Provisioning을 수행하는 방식을 취한다. 따라서 Real-time Provisioning 전략을 사용하여 기업은 사용자가 필요로 하는 양 만큼만 자원을 할당하고, 사용자는 필요로 하는 만큼만 자원을 사용하는 이점을 가져다 줄 수 있다. 이와 같이 클라우드 컴퓨팅에서 확장성 및 신축성은 Real-time Provisioning을 통해 기업 및 사용자 모두에게 이점을 제공하는 특성이라 할 수 있다.



(a) (b) (c)

그림 1. Over-provisioning, Under-provisioning & Real-time Provisioning

2.4 가상화 및 데이터 관리 기술

가상화(Virtualization) 기술은 컴퓨팅에 사용되는 물리적인, 또는 논리적인 자원을 추상화시켜 하나의 자원으로 관리될 수 있도록 하는 기술을 의미한다. 가상화가 가능한 자원으로는 CPU, 메모리, 저장 장치와 같은 물리적인 자원뿐만 아니라 운영체제와 같은 논리적인 자원까지도 포함한다. 이 중 컴퓨팅에 대한 가상화 기술은 클라우드 컴퓨팅을 이루는 핵심 기술로, 컴퓨팅 가상화를 통해 무한한 확장성 및 신축성을 보다 적은 노력으로 제공 가능하도록 한다. 컴퓨팅 가상화 기술을 이용한다면, 기존에 제어하던 물리적인 자원들을 그대로 둔 상태에서 컴퓨팅 가상 자원을 복제하여 늘이거나 할당된 자원을 해제하는 것만으로 컴퓨팅 가상 자원의 양을 줄이는 것이 가능하다. 또한 컴퓨팅 가상화를 관리하는 하이퍼바이저 기술을 통해 구성 변경에 대한 관리 노력을 감소시킬 수 있다. 이러한 가상화 기술의 장점은 클라우드 컴퓨팅의 Real-time Provisioning 전략과 가장 잘 부합한다고 할 수 있다.

클라우드 컴퓨팅 환경을 구성하기 위해서는 가상화 환경을 갖추는 데 필요로 하는 수많은 서버들을

물리적으로 구성하는 과정이 먼저 필요하다. 클라우드 컴퓨팅에서 사용되는 데이터 관리 기술은 데이터 센터(Data Center)에 존재하는 여러 대의 서버들이 분산된 환경에서 데이터들을 분산시켜 관리할 수 있는 기술을 의미한다. 이러한 기반 기술을 통해 클라우드 컴퓨팅에서 사용되는 데이터들은 물리적인 서버의 개수와 상관없이 데이터가 잘 분산되어 분산형 환경에서의 속도를 보장하며, 실시간으로 데이터 센터의 서버 대수를 늘리거나 줄이는 것을 가능하게 한다. 따라서 데이터 관리 기술 또한 클라우드 컴퓨팅의 확장성 및 신축성과 직결되는 중요한 기술이라 할 수 있다.

3장과 4장에서는 이러한 가상화와 데이터 관리에 대한 구체적인 기술이 클라우드 컴퓨팅의 확장성 및 신축성에 미치는 영향에 대해 자세히 살펴보고자 한다.

3. 가상화 기술 및 사례

클라우드 컴퓨팅은 서비스되는 모델에 따라 IaaS(Infractructure as a Service), PaaS(Platform as a Service), 그리고 SaaS(Software as a Service)로 구분된다[7]. 클라우드 컴퓨팅의 확장성 및 신축성은 위 3가지 배포 모델 모두와 관련되어 있지만, PaaS와 SaaS는 IaaS 모델에 의존하고 있을 뿐만 아니라 가상화된 컴퓨팅 자원들에 대한 Provisioning은 IaaS와 밀접하게 관련되어 있다. 따라서 이 장에서는 가상화 기술들 중 클라우드 컴퓨팅에 직접적인 영향을 미치는 컴퓨팅 가상화 기술들을 IaaS를 중심으로 클라우드 컴퓨팅의 Provisioning과의 관계 및 확장성 및 신축성에 미치는 영향을 살펴보고자 한다. 컴퓨팅 가상화 기술들을 가상 머신이 작동할 때 메모리 점유율, 성능 격리 및 가상 머신들 사이의 통신 등의 비교 기준에 따라 비교해 본다[8]. 또한, 가상 머신들을 관리하는 하이퍼바이저(Hypervisor)에 따른 확장성 및 신축성에 대한 영향을 알아보고자 한다.

3.1 하이퍼바이저에 따른 컴퓨팅 가상화 기술의 구분

컴퓨터 가상화는 가상 머신(Virtual Machine)과 함께 이를 관리하는 하이퍼바이저를 통해 구현된다. 이 때, 가상 머신 기술은 분산되어 있는 이 기종간의 물리적인 자원들을 가상 머신이라는 독립적인 동시에 개인화된 환경에서 동작할 수 있도록 하고, 커스터마이징한 가상 이미지를 사용할 수 있는 기술까지 포함한다. 가상 머신은 클러스터화하여 구성 가능하며, 이 때 각 가상 머신들은 그리드 컴퓨팅 및 고성능 컴퓨팅(High Performance Computing)에서 이루어졌던 방식과 마찬가지로, 각각의 노드가 멀리 떨어져 있어도 분산형 컴퓨팅을 수행할 수 있도록 동적인 가상 클러스터 생성을 가능하게 한다[9, 10]. 하이퍼바이저는 가상 머신 컨테이너(Virtual Machine Container) 들을 관리하고 감시하는 도구들을 의미한다.

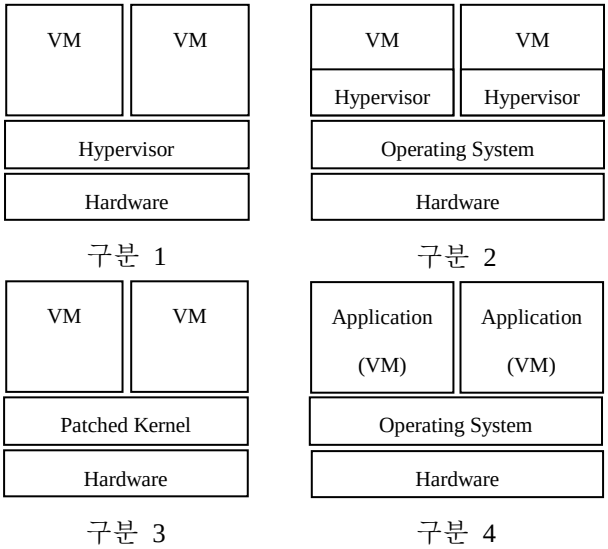


그림 2. 하이퍼바이저 종류에 따른 구분

그림 2는 하이퍼바이저 종류에 따른 컴퓨팅 가상화 기술의 4가지 분류를 나타내고 있다. 첫 번째 구분(그림 2-1)에서는 하이퍼바이저들이 직접적으로 컴퓨팅 하드웨어들과 인터페이스를 구성한다. 해당 구분에서 모든 운영체제들은 가상 머신 안에서 동작하며, 다른 가상머신들을 관리할 수 있는 가상 머신에 대한 허가 권한을 부가적으로 갖는다. 일반적으로, 구분1의 하이퍼바이저는 Para-virtualization 기술을 활용한다. 두 번째 구분(그림 2-2)에서 하이퍼바이저는 대개 Full-virtualization 기술을 활용하여 운영체제 내에서 하나의 프로그램처럼 실행된다. 이 때 실행되는 운영체제는 호스트로 동작하며, 각 사용자 운영체제들은 호스트 운영제제 내에서 프로세스로써 실행된다. 세 번째 구분(그림 2-3)에서는 가상화

기술이 임의의 커널 수정을 통해 원래의 커널과 분리되어 동작하여 독립된 가상 개인 서버 (Virtualized Personal Server)를 구성하는 형태이다. 해당 구분에서 가상 개인 서버는 물리적인 머신 상에서 직접적으로 동작하며 하드웨어 자원들을 공유하여 사용한다. 마지막 구분(그림 2-4)에서는 커널 복제를 통해 가상화 기술을 구현한 방식을 의미한다. 각 가상 운영체제는 이미 운영체제가 동작 중인 호스트 머신의 어플리케이션으로 실행되어, 각 게스트 운영체제들이 호스트 머신 운영체제 내에서 사용자 공간에서 동작하는 프로세스로써 실행된다.

3.2 가상화에서 확장성 및 신축성에 대한 고려 사항

컴퓨팅 가상화 기술은 크게 Start-up 소요 시간, 메모리 점유율과 가상 머신간의 통신 방식에 따라 클라우드 컴퓨팅의 확장성과 신축성에 직접적인 영향을 미친다고 할 수 있다.

Start-up 소요 시간: 클라우드 컴퓨팅의 Provisioning에서 각 사용자들은 커스터마이징된 가상 이미지를 가상 머신에 설치하거나 제거할 수 있어야 한다. 가상 머신이 부팅된 이후 초기화까지 걸리는 총 소요 시간은 Provisioning에서는 재설정이 빈번하여 총 소요 시간의 증가로 인해 시스템 전체 성능을 저하시킨다. 해당 시간을 줄이기 위해서는 가상 머신이 세션상에서 이미 동작하는 경우에만 세션상의 가상 머신을 활용하는 기술을 사용할 수 있는데, 이 때 크게 모든 자원들을 초기화하고 검사하는 과정을 거치는 Cold Start-up 전략과 반드시 초기화해야 하는 자원들만 초기화하고 검사를 생략하는 Hot Start-up 전략을 사용할 수 있다.

메모리 점유율: 가상 머신의 메모리 점유율이 적을수록 보다 많은 가상 머신들이 동시에 실행될 수 있기 때문에 메모리 점유율을 엄밀하게 측정하여 가상 머신의 확장성을 살펴볼 수 있다. 이 때, 비효율적인 메모리 사용은 시스템 확장성에 악영향을 미치는 만큼 메모리 점유율을 지속적으로 모니터링하여 효율적으로 메모리를 사용하는지 확인해야 한다.

가상 머신간의 통신: 가상 머신들끼리 메시지 교환이 빈번하게 일어나는 경우, 메시지를 주고 받는 통신 성능은 가상화가 가상 머신들 사이에서 내부 통신을 어떻게 구현했는지에 따라 달라진다. 통신 성능이 우수할 수록, 같은 서버 내에 동작할 수 있는 가상 머신의 개수를 보다 많이 지원함에 따라 클라우드 컴퓨팅의 확장성 및 신축성에 영향을 미친다.

해당 하이파바이저에 대한 여러 고려 요소들을 측정하기 위해서 사용되는 벤치마크 프로그램들에는 benchvm [14], Phoronix Test Suite [20], vConsolidate [15, 16], VMmark [17], 그리고 Virtbench [18] 등이 있다. 해당 벤치마크 프로그램들을 통해 각 요소들을 모니터링하여 가상화 기술들이 확장성 및 신축성을 잘 반영하는지를 확인할 수 있다.

3.3 가상화에서 확장성 및 신축성에 대한 고려 사항

현재 기업들이 사용하고 있는 주요 하이퍼바이저들로는 Xen, Vserver, VMWare, UML이 있다. 본 장에서는 각각의 하이퍼바이저에 적용된 상세한 구조 및 기술들을 살펴보고, 확장성 및 신축성 면에서 비교하고자 한다.

Xen[11]은 x86 아키텍처 상에서 동작하는 가상 머신 모니터로, 동작하는 운영체제들 사이에 자원의 격리 및 실행을 가능하게 할 뿐만 아니라, 여러 운영체제들이 동시에 실행되도록 지원한다. Xen은 Para-virtualization형태의하이퍼바이저로, 가상 CPU, 가상 메모리, 가상 네트워크 등과 같은 추상화된 집합을 제공하며 서로 다른 종류의 운영체제들을 동시에 지원한다.

Vserver[12]는 '보안 컨텍스트' 로 분리된 사용 가능한 자원들을 기반으로 커널 패치를 수행하는 방식을 취한다. 이 방식은 독립적인 가상 개인 서버들을 생성하여 하나의 물리적인 머신에서 동시에 실행될 뿐만 아니라 하드웨어 자원들을 공유 가능하도록 한다.

VMWare[13]는 가상화된 x86 호환 CPU 프로세서를 기반으로 x86 아키텍처에 대한 가상화 환경을 제공하는 하이퍼바이저이다. 현재 VMWare는 가상화 기술을 이용하는 범주에 따라 크게 데이터 센터 제품과 데스크톱 제품으로 분리한다.

UML (User Mode Linux)[19]는 리눅스 운영체제가 실행 되고 있는 호스트 머신에서 리눅스 OS가 어플리케이션마냥 실행되도록 설계 되었다. 이러한 특성은 머신 가상화가 호스트 머신의 특성들에 구애받지 않게 다양한 방식으로 설정되거나 활용 가능하게 끔 지원 한다.

표1은 위 4가지 하이퍼바이저에 대해 확장성과 신축성 면에 있어서 비교한 결과를 나타낸다[8]. 우선 가상 머신들 사이의 통신 방식에서는 vServer가 가장 나은 성능을 보여주는데 그 원인은 프로세스 및 프로세스 사이의 통신이 루프백 장치를 통해 이루어져 있기 때문으로 판단된다. 반면 UML은 커널 통신을 위한 계층을 유저모드로 추가하여 과부하가 발생하므로 낮은 성능을 나타낸다. Start-up 소요시간 면에서 vServer가 Hot Start-up 시간을 감소시켜 가장 적은 시간을 쓰고 있다. 메모리 점유율은 vServer을 제외하고는 하이퍼바이저들에 의하여 관리되는 가상 머신들이 예약된 메모리를 사용하여 가상 머신의 개수에 따라 메모리 점유율이 달라지며, vServer에서는 가상 머신들이 단일 커널 인스턴스에만 의존하여 메모리 제약에 상관없이 가상 머신들이 동작하는 특징을 지닌다.

표 1. 주요 하이퍼바이저들에 대한 확장성 및 신축성 비교

구분	비교 기준	Xen	VMware	vServer	UML
성능	가상 머신들 사이의 통신 방식	UML보다는 나은 성능을 보인다.	Xen보다 조금 나은 성능을 발휘한다.	가장 우수한 성능을 발휘한다.	가장 낮은 성능을 보인다.
사용자에게 직접적인 확장성 & 신축성에 영향을 미치는 요소	Start-up 소요 시간	vServer보다 느리며 Hot Start-up과 Cold-Start-up 소요 시간이 거의 비슷하다.	4가지 하이퍼바이저 중에서 가장 느리다.	가장 빠르며, 특히 Hot Start-up 소요 시간이 매우 빠른 이점을 나타낸다.	vServer보다는 느리지만 Xen보다는 더 짧은 Start-up시간을 보여준다.
	메모리 점유율	가상 머신의 개수에 따라 달라진다. 가상 머신들은 예약된 메모리를 사용한다.	동작하는 가상 머신의 개수에 따라 달라진다.	메모리 한계에 상관없이 가상 머신이 동작한다.	가상 머신의 개수에 따라 달라진다. 가상 머신들은 예약된 메모리를 사용한다.

4. 데이터 관리 기술 및 사례

이 장에서는 데이터 관리 기술에 따른 클라우드 컴퓨팅의 확장성 및 신축성을 설명한다. 4.1장에서는 현존하는 데이터베이스 기술들이 확장성 및 신축성에 미치는 영향을 설명하고, 4.2장에서는 데이터베이스의 기본 4속성은 ACID에 따라 확장성 및 신축성을 비교한다. 4.3장에서는 클라우드 컴퓨팅에서 사용되는 Shared-nothing 데이터베이스를 소개하고, 4.4장에서는 클라우드 컴퓨팅에서 사용되는 데이터베이스들을 확장성 및 신축성 기준에 따라 비교한다.

4.1 데이터 관리 어플리케이션들과 확장성 및 신축성

현존하는 데이터 관리 어플리케이션들은 사용 목적에 따라 크게 트랜잭션형 데이터베이스와 분석용 데이터베이스로 구분한다. 트랜잭션형 데이터베이스는 인터넷 뱅킹에서부터 항공권 예매 등 다양한 활용 범위를 지원한다. 이 때 동작하는 어플리케이션들은 데이터베이스 트랜잭션의 4가지 기본 요소인 ACID(원자성, 일관성, 독립성, 지속성)를 완벽히 보장하여 사용자 입력을 처리하는 데이터베이스에 많이 활용되고 있다. 그러나 분산 환경에서는 ACID를 지원하기 위해 각 데이터베이스 시스템들끼리 통신하는데 오버헤드가 발생하여 성능 저하를 감수해야 하는 문제점을 안고 있다. 이에 대해서는 4.2장에서 자세히 기술한다. 반면, 분석용 데이터베이스는 비즈니스 계획, 문제 해결 및 의사 결정 지원 등과 같이 대량의 데이터에 대한 분석을 통하여 결과를 산출하는 상황에 많이 사용된다. 분석 작업에 특화되어 설계되었기 때문에 트랜잭션형 데이터베이스보다 더 큰 데이터 크기 및 시스템 규모를 지원해야 하며, 또한 해당 어플리케이션들은 많은 양의 쓰기 작업을 필요로 하기 때문에 같은 작업을 처리하는 데 있어 트랜잭션형 데이터베이스보다 높은 Throughput이 요구된다.

4.2 ACID와 확장성, 신축성

클라우드 컴퓨팅이 이루어지는 분산형 환경에서는 데이터베이스 역시 확장성 및 신축성을 지원해야 하는데, 데이터를 관리하는 데이터베이스의 종류에 따라 ACID를 모두 만족시키기 위해 여러 연구들이 진행되고 있다. 1980년대 데이터베이스 연구원들 및 설계자들은 분산형 환경에서 트랜잭션을 지원하는 동시에 완벽한 확장성을 지원하는 데이터베이스를 설계하는 것이 불가능하다고 판단하였다[26, 27]. 해당 연구 결과를 기반으로 트랜잭션형 데이터베이스를 통해 확장성을 지원하는 대신 ACID의 4가지 특징 중 하나 이상을 지원하지 않는 방식을 선택하기도 하였으며[28], key-value 구조집합을 기반으로 하여 보다 단순한 데이터 구조를 만들어 현존하는 관계형 데이터베이스에 존재하는 테이블들을 대응시키기도

하였다[29]. 최근에는 트랜잭션 기능을 데이터 처리 기능에서 분리하는 방법으로 트랜잭션을 지원하는 분산형 데이터베이스에 적용하는 연구 역시 진행되고 있다[30][31][21]. 반면, 분석용 데이터베이스는 ACID를 모두 지원하지 않아도 문제없이 어플리케이션이 실행되기에 확장성 및 신축성을 보장하기에 보다 용이하다. 분석용 데이터베이스는 입력 위주의 데이터베이스가 아니기 때문에 트랜잭션을 지원해야 하는 데이터베이스처럼 분산된 데이터 복사본을 만드는 방식으로 확장성을 보장하는 방식을 적용할 필요가 없으므로 확장성 및 신축성을 구현하기가 보다 편리하다[32].

4.3 Shared-nothing 데이터베이스

Shared-nothing (SN) 데이터베이스는 분산된 데이터를 공유하여 처리하는 기존 방식과는 구조적으로 다르게 설계하여, 데이터를 공유하지 않도록 구성한 아키텍처 상에서 실행되도록 설계한 데이터베이스이다. SN은 각 머신들 사이에서 어떠한 자원도 서로 공유하지 않는 방식을 취하여, 각 가상 머신들이 독립적으로 동작하고, 데이터 의존성을 제거하도록 한다. 이는 대량의 데이터들이 중앙에 먼저 저장된 후 각 머신들이 공유하는 시스템과는 대조를 이룬다[25]. SN 데이터베이스에서 각 머신들은 데이터 자원에 대한 중복 접근 없이 독립적으로 컴퓨팅을 수행한다. Google에서 사용되는 순수 SN 기반 시스템은 저가의 컴퓨터들로 이루어진 노드들을 시스템에 추가 하는 방식으로 그 규모를 무제한으로 확장하기에 용이한 구조를 갖는다.

SN 아키텍처를 기반으로 확장성을 제공할 때에는 트랜잭션형 데이터베이스에 적용하는 것이 분석용 데이터베이스에 적용하는 것보다 쉽지 않은데, 데이터들이 여러 사이트들에 분산되어 있고 이런 분산된 데이터들을 접근 & 조작하는 트랜잭션들로 하여금 단일 사이트로부터 데이터에 대한 접근 제한을 풀어주는 것이 상당히 어려운 작업이기 때문이다. 분석용 데이터베이스에서 사용되는 데이터는 상대적으로 큰 용량을 처리하며, 시간이 흐를수록 저장된 용량이 점점 늘어나는데 이는 바로 SN 아키텍처가 등장한 배경과 부합한다.

4.4 데이터 관리 시스템들의 비교

표 2. 확장성 및 신축성에 따른 각 데이터 관리 시스템 비교

DBMS	어플리케이션 목적에 따른 분류	ACID 보장 여부	Shared-nothing 지원 여부
BigTable	대부분 분석적 데이터 관리 목적으로 사용이 된다. Bigtable은 일관성을 잘 지원하지 못하여 트랜잭션 목적의 어플리케이션에 사용하기에는 적합하지 않다.	관계형 API를 완벽히 지원하지 않고, 특히 A(원자성)에 대해 잘 지원하지 못한다. 즉, 데이터를 변경한 후 실제 데이터베이스에 적용되기 까지 걸리는 시간을 예측하기 어렵다.	Shared-nothing 아키텍처를 기반으로 구현되어 SN에서 제공해주고 있는 로드밸런싱 기술을 잘 활용 할 수 있다.
SimpleDB	분석적 데이터 관리 목적으로 많이 사용 된다.	제한된 레벨의 ACID를 보장한다.	Bigtable과 유사한 아키텍처를 기반으로 구현되어 로드밸런싱 기술을 잘 활용 할 수 있다.
HBase	ACID를 완벽히 지원하지 않기 때문에 트랜잭션형 보다는 분석적 데이터 관리 목적으로 많이 사용 된다.	완벽한 ACID를 지원하지는 못한다.	Bigtable과 흡사한 아키텍처를 기반으로 구현이 되어졌기 때문에 Shared-nothing 아키텍처를 따르고 있다.

현재 기업들이 사용하고 있는 주요 데이터 관리 시스템들로는 Google Bigtable, Amazon SimpleDB, Hbase등 이 있다. 표2는 위 4가지 데이터 관리 시스템에 대해 적용된 구체적인 기술에 따라 확장성 및 신축성을 비교한 결과를 나타낸다. Google에서 개발한 Bigtable[22]은 여러 대의 서버에 분산 저장된 페타바이트(Petabytes) 이상의 데이터를 효율적으로 저장하기 위하여 설계된 분산 저장장치 시스템으로, 트랜잭션형 목적용에만 해당하는 어플리케이션을 사용하기에는 적합하지 않다. 반면, Amazon에서 개발한 SimpleDB[23]는 Amazon Web Services 기반 애플리케이션의 구조화된 데이터를 쉽게 저장하고 검색할 수 있는 빠르고 확장 가능한 실시간 데이터세트 인덱싱 및 쿼리 프레임워크를 지원하여 제한된 레벨의 ACID를 보장하는 가운데 확장성 및 신축성을 지원한다. 그리고 Hadoop Project에 속하는 Hbase[24]는 Bigtable을 기반으로 오픈소스화 되었으나 Bigtable보다 낮은 Concurrency Control 및 보다 적은 수준의 ACID를 보장하는 가운데 확장성 및 신축성을 지원하도록 개발되었다.

5. 결론

클라우드 컴퓨팅을 이루는 핵심 기술들인 가상화 및 데이터 관리 기술들은 보다 확장성과 신축성을 지원하도록 하는데 주요한 역할을 제공한다. 본 논문에서는 위 두 가지 기술이 클라우드 컴퓨팅의 확장성과 신축성에 미치는 영향들에 대해 기술적인 관점 및 적용 사례들을 위주로 살펴보았다. 가상화 기술은 클라우드 컴퓨팅에 사용되는 자원들을 가상적으로 관리하여 물리적인 자원들을 필요한 때에 자원의 양을 보다 쉽게 늘리거나 줄일 수 있도록 지원한다. 가상화의 종류 및 가상화에서 사용되는 하이퍼바이저 기술의 차이에 따라 클라우드 컴퓨팅에 미치는 확장성과 신축성에 차이가 있다. 그리고 여러 명의 사용자들이 수많은 양의 자원을 동시에 늘이거나 줄일 수 있어야 하기에, 클라우드 컴퓨팅에 사용되는 데이터 관리 기술 또한 확장성 및 신축성에 영향을 미치는 또 하나의 중요한 기술이다. 데이터 관리 어플리케이션들에 따라 확장성 및 신축성이 달라지며, 클라우드 컴퓨팅의 확장성 및 신축성을 보다 향상시키기 위해 ACID의 일부를 포기하거나, Shared-nothing 데이터베이스 기술을 이용하기도 하며, 이러한 데이터들을 관리하기 위한 기술을 필요로 한다.

확장성과 신축성의 관점에서 클라우드 컴퓨팅을 살펴보았을 때, 클라우드 컴퓨팅은 기존 컴퓨팅 방식에 비해 기업 및 사용자 모두가 필요한 자원의 양만큼을 활용할 수 있도록 하는 이점을 제공한다. 이러한 이점이 보다 잘 반영될 수 있도록 본 논문에서 언급한 기술들뿐만 아니라 다른 클라우드 컴퓨팅 기반 기술들이 클라우드 컴퓨팅의 확장성과 신축성에 어떤 영향을 미치는지에 대한 연구가 필요하다. 그리고 이러한 과정에서 해당 기술들이 보다 확장성과 신축성을 잘 지원할 수 있도록 하는 방법에 대한 연구 역시 필요할 것이다.

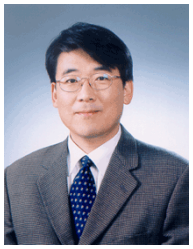
참고 문헌

[1] Anu A. Gokhale,"Introduction to Telecommunications", 2E p22~23, 2004.
[2] Brett D. McLaughlin, Sr., "Navigate the cloud computing labyrinth: Make an educated decision about the best cloud computing platform for your application", IBM, 2009.
[3] 류희수 "새로운 세상을 여는 클라우드 컴퓨팅(Cloud Computing)", TTA: IT Standard Weekly 2009-41호, 2009
[4] Peter Mell and Tim Grance, "The NIST Definition of Cloud Computing", NIST, 2009.
[5] Yefim V. Natis at el. “Scalability, Elasticity, and Multitenancy on the Road to Cloud Services ", Gartner, Nov. 2009.
[6] 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.
[7] “Cloud Computing Use Case White Paper Version 2.0”, Cloud Computing Use Case Discussion Group, Oct 2009.
[8] Benjamin Quétier, Vincent Neri, Franck Cappello. "Scalability Comparison of Four Host Virtualization Tools", J Grid Computing 83–98, 2007.
[9] F. Berman, G. Fox, and T. Hey, "Grid Computing: Making the Global Infrastructure a Reality", Wiley and Sons, 2003.
[10] I. Foster and C. Kesselman, "The Grid – Blueprint for a New Computing Infrastructure", Morgan Kaufmann, 1998.
[11] “What is Xen?”, http://www.xen.org/.
[12] “Welcome to Linux-VServer.org”, http://linux-vserver.org/Welcome_to_Linux-VServer.org, Sep 2009.
[13] “VMWare”, http://www.vmware.com/.
[14] “Benchvm - A virtulization benchmarking tool suite”, http://code.google.com/p/benchvm/, Sep 2008.
[15] P. Apparao et al, "Characterization & Analysis of a Server Consolidation Benchmark", ACM/USENIX International Conference on Virtual Execution Environments (VEE), 2008.
[16] J.P. Casazza, M. Greenfield, and K. Shi, "Redefining Server Performance Characterization for Virtualization Benchmarking", Intel Technology Journal, 2006.
[17] V. Makhija et al, "VMmark: A Scalable Benchmark for Virtualized Systems", Technical Report, VMware, 2006.
[18] “Virtbench”, http://freshmeat.net/projects/virtbench/, Apr 2007.
[19] “The User-mode Linux Kernel Home Page”, http://user-mode-linux.sourceforge.net/.
[20] “Phoronix Test Suite – Linux Testing & Benchmarking Platform”, http://www.phoronix-test-suite.com/.

[21] Zhou Wei, Guillaume Pierre, and Chi-Hung Chi, "Scalable Transactions for Web Applications in the Cloud", Springer-Verlag Berlin Heidelberg, 2009.
[22] Fay Chang et al, "Bigtable: A Distributed Storage System for Structured Data". OSDI. 2006.
[23] “Amazon SimpleDB™ (beta)”, http://aws.amazon.com/simpledb/ .
[24] “Welcome to HBase!”, http://hadoop.apache.org/hbase/, Sep 2009.
[25] Michael Stonebraker, "The Case for Shared Nothing Architecture", Database Engineering, Volume 9, Number 1, 1986.
[26] B. G. Lindsay, L.M. Haas, C.Mohan, P. F.Wilms, and R. A. Yost, "Computation and communication in R*: a distributed database manager", ACM Trans. Comput. Syst., 2(1):24–38, 1984.
[27] J. B. Rothnie Jr et al, "Introduction to a System for Distributed Databases (SDD-1)", ACM Trans. Database Syst., 5(1):1–17, 1980.
[28] P. Helland, "Life beyond distributed transactions: an apostate’s opinion", In CIDR, pages 132–141, 2007.
[29] S. Das, S. Antony, D. Agrawal, and A. El Abbadi, "Clouded Data: Comprehending Scalable Data Management Systems", Technical Report 2008-18, UCSB, 2008.
[30] M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska, "Building a database on S3", In SIGMOD, pages 251–264, 2008.
[31] D. B. Lomet, A. Fekete, G. Weikum, and M. J. Zwilling, "Unbundling transaction services in the cloud", In CIDR Perspectives, 2009.
[32] Daniel J. Abadi, "Data Management in the Cloud Limitations and Opportunities", IEEE Computer Society Technical Committee on Data Engineering, 2009.

최 영 락
2009 포항공과대학교, 컴퓨터공학과 학사
2009 ~ 현재 포항공과대학교, 정보전자융합공학부 석사 과정
<관심분야> Autonomic 네트워크 관리, 클라우드 컴퓨팅

리 건
2007 연변과학기술대학교, 전자통신공학과 학사
2009 ~ 현재 포항공과대학교, 컴퓨터공학과 통합 과정
<관심분야> 클라우드 컴퓨팅, 스마트 그리드 네트워크 관리



홍 원 기
1983 Univ. of Western Ontario, BSc in Computer Science
1985 Univ. of Western Ontario, MS in Computer Science
1985 ~ 1986 Univ. of Western Ontario, Lecturer
1986 ~ 1991 Univ. of Waterloo, PhD in Computer Science
1991 ~ 1992 Univ. of Waterloo, Post-Doc Fellow
1992 ~ 1995 Univ. of Western Ontario, 연구교수
1995 ~ 현재 포항공과대학교 컴퓨터공학과 교수
2007 ~ 현재 포항공과대학교 정보통신대학원장
2007 ~ 현재 포항공과대학교 정보통신연구소장
2008 ~ 현재 포항공과대학교 컴퓨터공학과 주임교수
2008 ~ 현재 포항공과대학교 정보전자융합공학부장
<관심분야> 네트워크 트래픽 모니터링, 네트워크 및 시스템 관리.