KIICE 한국정보통신학회지 13권 1호
THE MAGAZINE OF KIICE
http://www.kiice.org

클라우드 컴퓨팅 기술동향 및 전망

최태영*

*금오공과대학교 컴퓨터공학과

목 차

Ⅰ. 서 론

클라우드 컴퓨팅은 최근 굉장히 활발하게 연구와 개발이 진행되고 있는 분야이다. 클라우드 컴퓨팅은 서버, 스토리지, 프로그램과 같은 IT 자원들을 구매하여 소유하지 않고 필요할 때 인터넷을 통해 서비스 받는 방식으로 정의할 수 있다 [1]. 또는 어디서나 이용가능하고 (ubiquitous), 편리하고, 공유되는 컴퓨터 자원들 (네트워크, 서버, 저장장치, 응용프로그램, 그리고 서비스)을 주문에 따라 구성할 수 있게 해주는 모델이며 특히 빠른 시간 내에 사용할 수 있고, 유지비용이나 수고가 매우 적을 수 있는 조건을 가진 모델이라고 정의하기도 한다 [2]. 이와 같이 클라우드 컴퓨팅은 기존의 기술들 (유비쿼터스, 공유 자원, 온-디멘드 구성 등)을 통합하면서 빠른 시간 내 사용과 유지비용 및 인력이 최소화되는 높은 수준의 조건이 부가된 개념이다.

클라우드 컴퓨팅을 성공적으로 적용한 사례로 자주 언급되는 것이 뉴욕타임즈의 신문기사 디지털화 작업이다. 미국 뉴욕타임즈는 1,100만 건에 해당하는 신문기사를 아마존의 클라우드 컴퓨팅을 이용하여 수일 만에 수백 달러의 비용으로 전환하는데 성공하였다. 이러한 클라우드 컴퓨팅의 서비스는 어떤 서비스를 제공하는가에 따라 Infrastructure as a Service (IaaS), Platform as a Service (PaaS), 그리고 Software as a Service (SaaS)와 같이 분류된다. IaaS는 저장공간이나 데이터베이스와 같이 IT 자원을 제공하는 것으로 국내의 다음 클라우드, 네이버 N드라이버, 유클라우드 등과 해외의 아마존 Elastic Computer Cloud (EC2)가 이에 해당한다. PaaS는 사용자가 서비스를 사용할 수 있는 플랫폼 또는 환경을 제공하는 것으로 구글 앱 엔진 (Google App Engine), 마이크로소프트 윈도우 애저 (Microsoft Window Azure), 그리고 아마존 웹 서비스 (Amazon Web Service, AWS)가 이에 해당한다. SaaS는 인터넷을 통해 응용프로그램을 수행할 수 있도록 지원하는 서비스로 구글의 지메일 (Gmail), 구글 독스 (Google Docs), 그리고 세일즈포스 (Salesforce)의 CRM 등이 대표적이다.

클라우드 컴퓨팅의 상용화 과정은 그 서비스가 제공되는 대상에 따라 크게 두 가지 방향으로 진행되고 있는데, 하나의 조직에게만 자동화된 컴퓨팅 자원을 제공하는 것을 목표로 하는 Private cloud와 다수의 개인을 대상으로 클라우드 서비스를 제공하는 Public cloud로 나눌 수 있다. 물론 단일 대상이 아닌 특정 그룹들에게만 제공하는 Community cloud나 위의 모델들이 혼합된 Hybrid cloud도 상용화 모델에 포함될 수 있다.

Public cloud 모델들은 구글 앱, 아마존 웹 서비스, 그리고 세일즈포스의 CRM 등이 대표적이며 클라우드 컴퓨팅 모델을 가장 잘 반영하고 있다. 사용자는 필요에 따라 컴퓨팅 파워나 저장 공간의 용량을 변경할 수 있으며, 사용량이 가변적일 때 매우 유용하다. 반면 서비스 업체가 제공하는 인터페이스만 사용할 수 있으

며, 사용자가 자신의 데이터에 대해서 완벽하게 제어할 수 없고 보호와 보안이 유지되는 지를 확신할 수 없다는 단점들이 있다.

Private cloud를 제공하는 곳은 VMware, IBM, 그리고 마이크로소프트 등이 있다. Private cloud는 내부망이라는 물리적으로 격리될 수 있는 범위 내에서 구축되는 클라우드 모델로서 물리적인 보안의 지원을 받는다는 장점이 있다. 반면 private cloud의 단점은 IT 부서가 시스템을 유지하기 위해서 지속적으로 하드웨어나 소프트웨어를 구입하고, 구축하고, 관리해야 한다는 점이다. 또한 초기 설치비용이 많이 들기 때문에 cloud 시스템 정의의 일부인 필요할 때 빨리 이용할 수 있어야 한다거나 유지비용이 적어야 한다는 점이 보장되지 않는다는 것이다 [3].

Hybrid cloud는 private cloud나 public cloud가 두 개 이상 모여 만들어진 클라우드 구조이지만 내부의 각 cloud가 별도로 동작하는 것이 아니라 표준과 특성 기술이 같이 운용되는 시스템이다. 예를 들어 방대한 자료의 저장을 위해서 public cloud인 Amazon의 Simple Storage Service (A3)을 이용하지만 고객 관리 자료는 회사 내 저장 시스템을 이용하는 경우이다.

Ⅱ. 하드웨어와 운영체제

클라우드 컴퓨팅은 어디서나 쉽게 접근할 수 있는 기능을 포함해야 하므로 기존 PC나 터미널뿐만 아니라 하드웨어적 파워가 상대적으로 약한 임베디드 시스템에서도 접근할 수 있어야 한다. 그러한 요구조건들 중 하나는 조작성과 접근성이며 큰 화면, 키보드, 그리고 마우스 대신 작은 화면, 몇 개의 버튼, 그리고 터치스크린으로 이루어진 터미널에서도 접근할 수 있도록 인터페이스와 성능이 맞추어져야 한다. 하지만 이러한 접근성의 다양함은 클라우드 컴퓨팅을 사용하는 모든 하드웨어와 운영체제가 이런 디바이스에 대해 준비가 되어 있어야 한다는 것을 뜻하는 것이 아니라 클라우드 컴퓨팅 제공자는 다양한 디바이스를 통해 접근하는데 장애가 되지 않도록 보편적인 인터페이스를 제공하면 되고, 터미널 입장에서는 특정한 환경에 수행될 수 있도록 다른 기능을 배제시켜도 됨을 뜻한다. 이러한 개념위에 등장한 개념이 JeOS (Just Enough Operating System)이며 Google Chrome OS와 Ubuntu JeOS가 이에 해당한다. JeOS는 특정 응용프로그램에 적합하도록 특화된 운영체제를 의미한다. JeOS에서 수행되는 응용프로그램은 어플라이언스 (appliance)라고 불리며 모든 상황을 고려할 필요 없이 특정된 하드웨어와 JeOS 상에서만 수행되면 충분하므로 매우 적은 크기를 가지도록 만들어질 수 있다.

Joli OS는 리눅스 운영체제의 일종으로 넷북과 같이 제한된 자원을 가진 컴퓨터를 위해 설계되었다. 사용자 인터페이스는 HTML5로 작성되었으며 크로미엄 웹 브라우저를 웹 접근을 위한 미들웨어로 사용한다. 크로미엄 웹 브라우저는 소스로 제공되는 구글 크롬 웹 브라우저 버전이다.

Glide OS는 웹 OS의 일종이며 윈도우즈와 같은 데스크탑 운영체제나 구글 안드로이드와 같은 모바일 운영체제에서 모두 수행될 수 있는 특징을 가지고 있으며 HTML5로 구현되어 있다. 또한 마이크로소프트 인터넷 익스플로러나 구글 크롬과 같은 주요 웹 브라우저와 호환되는 특징을 가진 범용 클라우드 터미널 기능을 제공한다.

Ⅲ. 파일시스템

클라우드 컴퓨팅 환경 내에 포함된 각 컴퓨터들은 규격화되어 있지 않기 때문에 사용 가능한 저장장치의 종류나 용량이 모두 다르고, 파일 시스템도 제각각으로 구성되어 있다. 특히 클라우드 컴퓨팅 환경에 포함되어 있는 컴퓨터의 개수는 수백 개에서 심지어 수만 개가 넘을 수 있으므로 항상 고장 난 컴퓨터가 존재하게 된다. 따라서 데이터는 여러 개로 복제되어 저장 공간이 있는 여러 컴퓨터에 배치되어야 하고 이를 관리하기 위한 메타 데이터 또한 중복 및 분산되어 저장되어야 한다. 하둡 (Hadoop)에서 파일 저장을 위해 개발된 HDFS (Hadoop Distributed File System)는 기본적으로 매우 큰 크기의 파일들을 저장하기 위해 만들어져 있으며, 성능을 중시해서 POSIX 표준을 따르지 않고 있다. HDFS는 성능 유지를 위해 구성 컴퓨터들의 위치를 모두 파악하고 있으며 노드들을 랙 (rack, 네트워크 스위치를 공유하는 노드들의 집합)의 단위로 그룹을 만들어 관리한다. 랙은 하나의 로컬 네트워크와

같은 역할을 수행한다. 한 파일은 여러 조각으로 나누어지고 각 조각은 여러 노드들에 저장되는데 기본적으로 3개의 복사본이 존재한다. 2개의 복사본들은 하나의 랙에 존재하고, 나머지 하나는 다른 랙에 저장함으로써 노드나 네트워크 스위치가 고장 나더라도 여전히 파일에 접근할 수 있도록 한다.

구글 (Google)은 자신의 데이터 관리를 위해서 구글 파일 시스템 (Google File System, GFS)을 개발했다. GFS가 다루는 파일은 기본적으로 대용량의 파일이며, 각 파일은 64메가바이트 크기의 청크 (chunk)라고 불리는 블록들로 구성된다. 이와 같이 과도하게 큰 크기의 블록들로 파일들을 구성하는 이유는 GFS를 사용하는 응용프로그램들은 파일의 중간 부분을 갱신하지 않고 끝에 추가하는 작업이 대부분이므로 한꺼번에 큰 크기를 읽더라도 부담이 되지 않으며 큰 크기의 읽기 작업을 한번 수행하는 것은 작은 크기의 읽기 작업을 여러 번 수행하는 것보다 처리율을 높일 수 있기 때문이다. 청크들은 chunkserver라 불리는 노드 (node)에 저장되며 각각의 청크를 구분하기 위해 64-bit 크기의 라벨을 붙인다. 한 청크는 원본을 포함해 최소한 3개 이상의 복사본이 다른 노드들에 존재해서 한 개나 두 개의 노드가 고장이 난다 하더라도 접근할 수 있는 청크가 존재하도록 구성된다. GFS에서 chunkserver외에 마스터서버가 존재하며 이 서버는 각 파일이 어떤 청크들로 이루어져 있고, 각 청크의 라벨이 무엇이며 그 복제본들이 어느 노드들에 존재하고 있는지에 대한 메타데이터를 담고 있다. 또한 어떤 프로세스가 어떤 청크에 접근하고 있는지에 대한 정보도 포함하며 이들 메타데이터는 chunkserver로부터의 heart-beat 메시지를 사용하여 주기적으로 갱신된다 [4]. GFS의 특징 중 하나는 기존의 커널에서 지원되는 파일시스템들과는 달리 라이브러리의 형태로 지원되며 리눅스 파일 시스템 위에서 수행된다는 점이다.

Ⅳ. 프로그래밍 환경 및 스케줄링

클라우드 컴퓨팅에서 큰 크기의 작업을 분할하여 각 작업 노드에 할당하고 수행된 결과를 모으는 것은 매우 복잡한 과정을 거쳐야 한다. 예를 들어, 어떤 작업 노드는 작동이 되지 않는 상태일 수도 있고, 매우 느리게 동작하는 상황일 수도 있다. 이러한 상황을 모두 고려한 프로그래밍은 많은 시간과 비용을 필요로 하기 때문에 이를 단순화하기 위한 프로그래밍 기술이 필요하다. 구글의 MapReduce는 개별 작업 노드에서 공통적으로 수행하는 map과 이들의 결과가 통합되는 과정인 reduce를 제공하여 노드가 매우 많은 환경에서도 프로그래머가 쉽게 작업을 분할하고 그 결과를 모을 수 있도록 한다 [5]. 이 기능을 사용하기 위해서 사용자는 우선 마스터 노드에서 수행할 코드와 작업 노드에서 수행할 코드를 각 노드에 올린 뒤 마스터 노드에서 map 과정을 수행하도록 요청한다. Map 과정에서는 마스터 노드가 입력 데이터를 받아 작은 크기의 문제들로 분할한 뒤 작업 노드들에게 할당한다. 각 작업 노드는 자신이 가진 파일로부터 마스터 노드가 요청한 작업을 수행하고, 그 결과를 마스터 노드에게 보내기 위해 reduce 과정을 수행한다. Reduce 과정에서는 작업 노드들이 구한 작은 크기 문제에 대한 해답들이 결과파일에 저장된다. MapReduce는 수행 중간에 작업 노드들 간에 상호 통신이 전혀 필요 없는 문제에 대해서만 적용할 수 있다는 단점이 있지만 구글 검색에서와 같이 MapReduce의 형태로 분할 수행 가능한 작업들에 대해서는 병렬화 작업이 매우 쉬운 프로그래밍 코드로 이루어지며 한 두 노드나 파일의 고장이 있는 경우라도 다시 스케줄을 하거나 복제된 청크를 이용함으로써 결함 포용에 강하다는 장점이 있다.

스케줄링, 특히 작업 할당 (Job Scheduling)은 클라우드 컴퓨팅 미들웨어의 성능을 결정하는 중요한 요소들 중 하나이다. 그리드 시스템과 같은 이전 시스템에서의 스케줄링은 주로 사용자의 QoS (Quality of Service)를 어떻게 만족시킬 것인지에 중점을 두었다고 한다면, 클라우드 컴퓨팅에서의 스케줄링은 자원 제공자의 이익도 고려해야 한다. 사용자의 QoS와 자원 제공자의 이익은 서로 상충되는 요소들이다. 따라서 스케줄러는 두 요소를 모두 고려하여 QoS를 만족하면서도 클라우드 컴퓨터의 자원은 최소한으로 사용하도록 작업을 할당해야 한다. 제한 조건 속에 자원을 최소한으로 사용하는 것은 Linear programming으로 풀 수 있지만 많은 시간이 소모되므로 Luqun Li는 각 작업의 우선순위를 조절하고 QoS를 만족하면서 자원 사용을 더 줄일 수 있는 지를 반복적으로 검사하는 알고리즘을 제안하였다 [6].

각 작업 노드에는 Map 함수 등에 의해 여러 개의 작업들이 배당되는데, 클라우드 컴퓨팅은 응답시간 보다는 처리율을 중시하므로 각 노드의 스케줄러는 FIFO 스케줄링을 선호한다. 따라서 작업 노드에서의 스케줄링 알고리즘은 대체로 간단하며, 하둡의 경우 FIFO 스케줄링에 5단계의 비선점 우선순위를 사용한다.

하둡 (Hadoop)은 구글의 MapReduce와 구글 파일 시스템 (GFS)로부터 파생된 소프트웨어 프레임워크로서 다량을 데이터 처리를 위한 프로그램을 지원하기 위해 오픈 라이선스 소프트웨어로 개발되었다. 하둡은 리눅스 운영체제 위에서 라이브러리와 응용프로그램으로 사용자에게 하둡 파일 시스템 (Hadoop Distributed File System, HDFS)과 MapReduce 엔진을 제공한다. 하둡을 사용하는 프로그램으로는 Yahoo! Search Webmap이 있으며 10,000개 이상의 리눅스 클러스터에서 수행되고 있다. Facebook은 30 페타바이트 크기의 용량을 가진 하둡 클러스터를 운영하고 있다.

Ⅴ. 보안관련이슈

클라우드 컴퓨팅의 보안 이슈는 지속적으로 언급되고 있으며, 특히 시스템이 일반에게 노출되어 있는 public cloud의 경우 사용자들이 특히 우려하고 있는 주제이다. Private cloud는 물리적으로 고립될 수 있으므로 외부로부터의 공격에는 비교적 안전하다고 볼 수 있지만 부분적으로 공유된 데이터에 허락받지 않은 내부 사용자가 접근을 시도하게 되면 이를 얼마나 잘 막아낼까 하는 의문에 대해서는 public cloud의 경우와 그다지 다르지 않다.

앞에서도 언급했지만 클라우드 컴퓨팅이 가용성을 어떻게 제공할 것인가는 여전히 중요한 이슈이다. 클라우드 컴퓨팅 환경은 수백 또는 수만 대의 다양한 컴퓨터들로 구성되어 있으며, 그 개수에 비례하여 고장과 같은 오류가 발생한다. 구글과 하둡에서 파일을 복제함으로써 가용성 문제를 해결하려는 시도를 보이고 있다.

클라우드 컴퓨팅 시스템에서 부분적으로 공유되는 자원들을 어떻게 유지하는가 하는 문제는 인가에 관련된 문제이며 특히 사용자의 수가 수만 명이 넘는 경우에 이를 효과적으로 관리하기 위한 방법이 필요하다 [7].

프라이버시 문제는 많은 사용자들이 존재하는 컴퓨팅 환경과 같이 빠른 시간 내에 민감한 내용이 급속도로 개인의 프라이버시는 무시되는 경향이 강하므로 이를 제어할 방법이 필요하다는 것이다 [8]. 이를 위해 사용자 개인 정보를 잘 알아볼 수 없도록 하는 (obfuscation) 서비스가 제안되었다.

이와 유사한 문제로 개인의 비밀 정보가 클라우드 상에 노출될 수 있다는 두려움 때문에 많은 사람들이 클라우드 컴퓨팅을 이용하는 것을 주저하는 점이 있다. 클라우드 컴퓨팅에 저장된 자료들은 그 자료에 대한 소유자가 완전히 제어할 수 있지 않다. 예를 들어 한 파일이 클라우드 컴퓨팅 파일 시스템에 의해 여러 노드에 분산되어 저장되는데, 사용자는 그 노드의 위치를 알 수 없고, 저장 위치가 시스템의 상태에 따라 변경되기도 한다. 이에 대해 제시된 방법들 중의 하나는 클라우드 컴퓨팅 시스템의 제공자가 사용자에게 보호에 대한 다계층 정책을 제시하고 그 중 하나를 선택하도록 하는 것이다. 또한 그 제공자는 다계층 정책을 지킨다는 보장을 제공하도록 한다 [9].

클라우드 컴퓨팅 환경에서 많은 서비스 제공자들이 다양한 형태의 서비스를 제공할 때, 그 서비스에 대해 청구하는 사용료가 제공된 서비스와 일치하는 지를 확인하는 방법도 고려해야 할 필요가 있다. 이는 사용자에 대한 정보가 잘못 입력되었을 가능성뿐만 아니라 정보가 저장되어 있는 클라우드내의 자료가 공격당했을 가능성도 염두에 두어야 하기 때문이다.

클라우드 컴퓨팅 시스템상의 컴퓨터들 사이에서는 XML 형태의 메시지를 이용한 통신 방법이 확대되고 있다. 그런데 이 XML 형태의 메시지에 대해 인증을 위한 서명을 무력화하는 wrapper attack이 가능하며 이를 막기 위한 방법이 필요하다. XML 메시지는 태그를 사용하여 본문과 서명을 구분하기 때문에, 그 메시지를 강탈한 공격자는 서명의 위치를 쉽게 파악할 수 있다. Wrapper attack은 전송되는 XML 문서를 가로채어 문서의 서명 주위에 수신 컴퓨터가 불필요한 작업을 하도록 유인하는 코드를 집어넣는 공격방식이다 [10]. Wrapper attack을 방지하는 방법은 공격 방식에 따라 다르지만 기본적인 형태의 공격에 대해서는 메시지를 받는 쪽과 보내는 쪽의 보안 정책을 메시지 내에 명시하는 것이다.

크롬OS는 웹 브라우저의 입장에서 클라우드 컴퓨터

내의 웹 서버로부터 서비스를 받을 때, 가짜 웹 서버의 피싱 (phishing)이나 악의적인 공격에 노출될 가능성이 많다. 웹 서버에 대한 쿼리나 메시지에 XML 암호화와 서명을 적용하여 이 문제를 해결할 수 있지만 XML을 지원하는 웹 브라우저나 서버는 일반적이지 않은 편이다. 대신 보편적으로 사용되고 있는 TLS/SSL (Transport Layer Security, Secure Socket Layer)를 이용할 수 있는데 모든 서버와 브라우저가 검증된 인증서를 준비해 두지는 않기 때문에 보편화하기에 어려움이 많다 [11].

클라우드 컴퓨팅 시스템이 DoS (Denial of Service) 공격에 상대적으로 취약한 것은 아니지만 일단 공격에 성공하여 클라우드 컴퓨팅 시스템 사용자의 계정을 획득할 수 있다. 공격자는 그 계정을 통해 대규모의 컴퓨팅 파워를 요청하여 다른 사이트에 DoS 공격을 가할 수 있다. 수 만개의 사이트로부터 DoS 공격을 당한 서버는 부하를 감당하기 매우 어려운 상황에 놓이게 되고, 계정을 빼앗긴 사용자에게는 막대한 사용료가 부가된다 [12].

클라우드 컴퓨팅과 같이 한 하드웨어를 여러 사람들이 공유하는 경우에는 개인의 평판이 이 하드웨어에 의해 관계를 가진 사람들 모두에게 영향을 끼치게 된다. 예를 들어, 앞에서 언급한 바와 같이 클라우드 컴퓨팅을 이용한 DoS 공격이 발생하면 그 공격을 당한 단체나 인터넷 관리자들은 그 클라우드 컴퓨팅 시스템을 블랙리스트에 올려 그 시스템으로부터의 패킷을 차단할 가능성이 늘어난다. 더욱이 이러한 사이버 범죄를 조사하는 경찰은 그 클라우드 컴퓨팅 시스템을 이용하는 사용자들을 대상으로 소환 조사를 할 수 있으며 사용자들은 이로 인해 많은 시간과 경제적 손해를 입을 수 있다.

여러 개의 가상 머신이 수행되는 클라우드 컴퓨터 상에 이들 가상 머신이 공유하는 자원이 있고 이를 통해 가상 머신들 간에 정보 교환이 이루어지는 환경이라면 공격자는 이 자원을 이용하여 정보 교환을 가로막거나 변조할 수 있다.

클라우드 컴퓨팅 시스템은 네트워크에 매우 의존적이기 때문에 네트워크가 테러나 재난에 의해 중단되는 경우 이와 관련된 많은 사용자들이 서비스를 받을 수 없게 된다. 이러한 문제를 해결하기 위해 지리적으로 분리되어 있는 곳에 보조 클라우드 시스템을 두거나 둘 이상의 클라우드 컴퓨팅 업체가 협약을 맺어 서로 간의 지원을 할 수도 있다.

클라우드 컴퓨터에 포함되어 있는 자원들에 대해 보안 수준을 정할 때, 전체 자원들에 대해 동일한 보안 수준을 적용하는 정책을 택할 수도 있고, 자원의 소유자가 보안 수준을 변경하도록 할 수도 있다. 하지만 사용자가 보안 수준을 변경하는 경우 보안 수준이 높은 자원에 대해서만 보안 기능을 할당하면 되는 효율성이 있는 반면, 공격자가 중요해 보이는 자원이 무엇인지를 눈치 채게 하는 문제점도 가지고 있다.

Ⅵ. 결함포용

많은 클라우드 컴퓨팅 시스템들이 결함 포용을 위해서 다수의 복제본을 관리하고 있지만 반대로 복제본의 존재는 일관성 문제를 일으키는 요인이 된다 [13]. 클라우드 컴퓨팅 시스템은 많은 수의 컴퓨터들에 의해 구성되어 있으므로 고장은 일상적으로 발생하지만 이와 마찬가지로 복제본이 서로 다른 경우가 존재할 수 있다. 사용자가 한 파일의 변경을 요청하면 그 요청을 받은 원본은 요청에 따라 파일을 변경하고 자신의 복제본들이 있는 노드들에게 변경 요청을 보낸다. 일시적인 네트워크의 오류 또는 외부의 공격으로 인해 그 요청을 담은 패킷이 중간에 소실되거나 변경되는 경우 원본과 복제본은 다른 내용을 가지게 되는 문제가 발생한다. 또는 원본에서 복제본으로 보내지는 내용 변경 패킷이 전달되는 시간이 너무 오래 걸려 일시적으로 일관성이 깨지게 되고 시스템은 일관성에 문제가 있다고 판단하여 잘못된 동작을 취할 수도 있다. 이러한 문제에 대해 LLFT 미들웨어는 요청 메시지나 변경 메시지들의 순서를 결정하여 이를 통해 일관성이 유지되지 않는 이유를 판단하여 적합한 수정 작업을 수행한다 [14].

Ⅵ. 전망

IBM을 포함한 많은 IT 업체들의 적극적인 보급으로 클라우드 컴퓨팅의 보급은 계속 가속화되어 왔다. 클라우드 컴퓨팅의 초창기에는 기업체들이 보안문제를

두려워하여 private cloud를 선호했지만 hybrid cloud에 대한 관심이 증가되고 있다. 또한 뉴욕 타임즈의 클라우드 컴퓨팅 적용 사례들이 보급되면서 클라우드 컴퓨팅을 도입하려는 기업체와 단체들이 증가하고 있다.

반면 클라우드 컴퓨팅은 지금까지 언급한 바와 같이 여전히 해결해야 하는 문제들과 개선될 여지를 안고 있는 열려있는 분야이다.

참고문헌

[1] 성병용, "국내 기업의 클라우드 컴퓨팅 동향 및 전략", *SW Insight,* July 2009.

[2] Peter Mell and Timothy Grance, *The NIST Definition of Cloud Computing,* NIST, 2011.

[3] John Foley, "Private Clouds Take Shape," *InformationWeek,* August 09, 2008.

[4] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google File System," *SOSP'03*, Bolton Landing, New York, USA, October 19~22, 2003.

[5] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Communications of the ACM, Vol. 51, No. 1, pp.107-113, January 2008.

[6] Luqun Li, "An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers," *Third International Conference on Multimedia and Ubiquitous Engineering, 2009.* (*MUE '09*). pp.295 – 299, 2009.

[7] Wenchao Zhou, Micah Sherr, William R. Marczak, Zhuoyao Zhang, Tao Tao, Boon Thau Loo, and Insup Lee, "Towards a Data-centric View of Cloud Security," *CloudDB2010*. 2010.

[8] Siani Pearson, Yun Shen, and Miranda Mowbray, "A Privacy Manager for Cloud Computing," *COMSWARE'09*, Chicago USA. 2009.

[9] Dan Lin and Anna Squicciarini, "Data Protection Models for Service Provisioning in the Cloud," SACMAT'10, Pittsburgh Pennsylvania USA. 2010.

[10] Michael McIntosh and Paula Austel, "XML Signature Element Wrapping Attacks and Countermeasures," *Technical report*, IBM Research, Hawthorne, New Yerk, 10532, 2005.

[11] John C. Roberts II and Wasim Al-Hamdani, "Who Can You Trust in the Cloud? A Review of Security Issues within Cloud Computing," *Information Security Curriculum Development Conference 2011,* Kennesaw GA USA. October 7-9, 2011.

[12] Huan Liu, "A new form of DOS attack in a cloud and its avoidance mechanism," in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop* (*CCSW '10*), pp.65-76, 2010.

[13] Flavio Lombardi and Roberto Di Pietro, "Transparent Security for Cloud," *SAC'10*, Sierre Switzerland. March 22-26, 2010.

[14] Wenbing Zhao, P.M. Melliar-Smith, and L.E. Moser, "Fault Tolerance Middleware for Cloud Computing," *IEEE 3rd International Conference on Cloud Computing (CLOUD)*, 2010.

저자소개

최태영(Tae-Young Choe)

포항공과대학교 컴퓨터공학과 공학박사

포항공과대학교 컴퓨터공학과 석사

고려대학교 수학교육과 학사

※관심분야 : 병렬/분산 처리, 대용량 저장장치, 컴퓨터 시스템보안