모바일 클라우드 컴퓨팅 소프트웨어 기술

이성원 | 김현준
경희대학교

요 약

본 고에서는 최근 IT 분야의 주요 이슈로 부각되고 있는 클라우드 컴퓨팅 환경과 관련한 소프트웨어 기술에 대해서 알아본다. 특별히 IMT-Advanced를 통한 4세대 이동통신의 상용화가 몇 년 앞으로 다가와 있으며, Apple iPhone/iPad의 확산에 따른 모바일 기기와 서비스에 대한 급격한 관심과 사용의 증가를 고려하여, 모바일 클라우드 컴퓨팅 환경을 고려한 기술에 대하여 알아본다.

I. 서 론

클라우드 컴퓨팅이라는 단어는 그 동안 그리드(Grid) 컴퓨팅, 유틸리티(Utility) 컴퓨팅, thin-client 기반의 컴퓨팅 등의 다양한 모습으로 발전해 왔으며, 2008년 ~ 2010년간 이를 활용한 상용 서비스의 개시와 실질적인 성능과 장점 등의 입증이 가시화됨에 따라서 많은 관심을 받고 있다. 특히 일반 기업체에서 보안적인 측면을 고려하여, 모든 자료가 서버에 저장되고 개인이 유출하기 어려운 장점으로 인하여 긍정적으로 평가되고 있는 실정이다. 아울러 Green IT에 대한 관심 증가와 맞물려서 기업의 서버 등 인프라 비용 감소와 함께 전력 사용량 감소라는 부수적인 효과까지 고려되어 활발하게 도입을 고려하고 있다.

특히 클라우드 컴퓨팅은 client 기기에 대한 하드웨어 및 소프트웨어 측면의 요구사항이 대부분 웹 브라우저의 구동만 이루어지면 되는 수준으로 낮아 졌다는 점, 그리고 최근 들어 노트북과 Apple iPhone/iPad 류의 모바일 휴대기기에서의 인터넷 사용이 급격하게 확산되는 점과 맞물려 모바일 비즈니스 및 콘텐트 시장에서 큰 관심을 받고 있다. 이러한 추세는 현재 일반화된 3세대 이동통신을 넘어서, 표준이 거의 완료된 3GPP LTE(Long-Term-Evolution)/LTE-Advanced와 Mobile-WiMAX 16m의 상용화가 본격화 되면, 현재보다 더 많은 부문으로의 확산이 이루어 질 것으로 보인다. Juniper Research에 의하면 cloud 기반의 모바일 어플리케이션은 2009년 $400 million 이었으며, 2014년에서는 $9.5 billion으로 증가할 것으로 예상하고 있다.

현재 클라우드 컴퓨팅에 대한 정의 자체도 모호하여, 솔루션을 제공하는 업체나, 서비스를 제공하는 업체 별로도 다른 정의를 내리고 있는 상황이기에 모바일 클라우드에 대한 정의를 내리기에는 아직 이른 시점이지만, 현재 논의되고 있는 모바일 클라우드 환경은 크게 2가지 부문으로 고려되고 있다.

첫 번째 정의는 현재의 클라우드 모델과 유사한 형상으로서, 클라우드 서버에 접속하는 client 기기가 모바일 환경을 사용하기에, 현재의 유선 기반 기기보다 제한된 성능과 용량을 가지면서, 유선 대비 동적이고 불안정한 무선 통신을 이용하고, 휴대폰과 같이 자유롭게 이동까지 가능한 환경을 의미한다. 이로 인하여 기존의 유선 통신 중심의 클라우드 컴퓨팅 대비 네트워크에서 클라우드 서비스의 효과적인 지원을 위하여 제공해야 하는 추가 기능이 필수적으로 요구되며, 단말과 클라우드 서버간에도 무선 통신 환경의 특성을 고려하여 최적화된 프로토콜이 필요하게 된다.

두 번째 정의는 클라우드 서버들간의 통신이 모바일 통신을 활용하는 경우이다. 특정 지역에 무작위적으로 혹은 계획적으로 배포된 대규모의 센서 기기들이 각자의 프로세싱 파워와 저장공간을 클라우드 형태로 활용하고, 이들간의 통신이 모바일 통신으로 이루어 지는 경우가 고려할 만한 사례로 볼 수 있다[1].

본 고에서는 LTE 기반 이동통신이 사용화되고 있는 현시점에서 첫 번째 정의에 부합하는 환경에서의 모바일 클라우드 소프트웨어 기술에 대하여 다루고자 한다. 이후 클라우드 컴퓨팅의 환경이 보다 확산되고 사용처가 많아지면 두 번째 정의에 대한 연구도 보다 확산되리라 본다.

본 서론에 이어서, 2장에서는 모바일 클라우드 컴퓨팅을 위한 단말 소프트웨어 기술을 살펴보며, 3장에서 서버 측의 소프트웨어 기술을 모바일 상황에서 발생하는 네트워크 이슈와 이에 대한 해결 방안이 4장에서 논의되고, 5장에서 결론을 내린다.

II. 모바일 클라우드 단말 SW 기술

현재 사용되고 있는 클라우드 컴퓨팅의 단말은 thin-client 형상으로 개발이 되고 있다. 대부분 Microsoft Windows, Linux, Unix 등 범용 운영체제를 사용하는 컴퓨터에서 별도의 실행 프로그램으로 개발이 되거나, 웹 브라우저 상에서 동작하는 thin-client 소프트웨어가 서버의 기능을 client protocol을 사용하여 작업의 실행을 요청하고, 서버로부터의 응답을 받아 처리하는 방식이다.

표준화된 단말-서버간 통신 규격이 없는 현재 시점에서 thin-client와 서버간에 사용되는 client protocol은 다양한 방식들이 사용되고 있으며, 대표적인 비 표준 규격들은 Simple Object Access Protocol (SOAP), Representational State Transfer (REST), Python Application Programming Interface (API), Hyper Text Transfer Protocol (HTTP), C++ API 등이다.

본 장에서는 단순하게 웹 브라우저 상에서 동작하는 프로그램 형태의 단말을 벗어나서 클라우드 환경에 최적화된 구조로 진화하고 있는 기술 중에서 Google의 Chromium 운영체제에 대하여 알아본다. 특히, Chromium은 무선랜 및 셀룰라 이동통신에 대한 고려를 처음부터 적용하여 개발하고 있다. 이와 함께, Cloud Computing 혹은 thin-client를 표방하고 있지는 않지만, 이미 Mobile-Me 등의 클라우드 서비스를 유선과 모바일 환경에서 지원하고 있으며, iPhone과 iPad를 통하여 모바일 트래픽의 큰 규모를 장악하고 있는 Apple의 iPhone Operating System (OS)에 대하여 알아본다. Apple iPhone OS의 경우도 최근의 업데이트를 통하여 무선통신 환경에 대한 지원을 핵심 기능으로 향상하였다.

2.1 Google Chromium OS

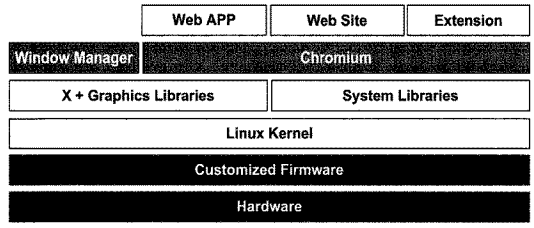
2009년 11월 Google은 open-source 형태로 Chromium OS를 발표하였다. Google은 Chromium의 개발을 위하여 GNU, the Linux Kernel, Moblin, Ubuntu, WebKit 등을 활용한 것으로 발표하였다. 개방된 Chromium OS 코드는 무료이고, 누구든지 활용하거나 개량할 수 있도록 하였다. Chromium OS는 대부분의 컴퓨터활용을 웹 브라우저를 통하여 수행하는 사람들을 목표로 하고 있다. 현재는 개발자들을 대상으로 하여 가장 기본적인 기능을 지원하도록 개발되어 있으며, 1년여의 시간을 통하여 추가적인 기능을 개발하여, 2010년 말에 일반 사용자가 활용할 수 있는 수준으로 발표하는 것을 계획하고 있다[2].

2.1.1 Layered 구조와 주요 기능

Chromium OS는 (그림 1)과 같은 계층적인 구조를 가지고 있다. 하부 계층부터 설명을 하며, (Customized) Firmware는 하드웨어를 제어하면서 부팅을 담당한다. Kernel과 System Library는 Linux를 기반으로 하여 만들어 졌으며, 범용 운영체제의 Kernel 대비 경량화된 작업을 한다. Chromium은 경량화를 표방하지만, X + Graphic Library를 기본적으로 제공함으로써 멀티미디어 작업에서도 문제가 없도록 설계되었다. Windows Manager와 Chromium 계층은 사용자 인터페이스와 윈도우 화면 처리 기능을 수행한다.

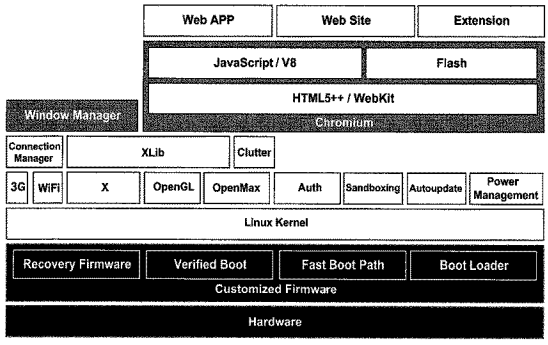
Chromium OS 각 Layer에서 수행하는 세부 기능을 보다 자세하게 나타낸 것이 (그림 2)이다.

Firmware는 운영체제의 부팅을 빠르고 안전하게 하도록 한다. Firmware는 Boot Loader와 Fast Boot Path를 통하여 빠르게 부팅되도록 되어 있으며, 부팅의 전 과정은 Verified



(그림 1) Chromium OS 내부 계층 구조도

Boot에 의한 검증을 통하여 바이러스 등으로부터의 오염여부를 확인한다. Recovery Firmware는 시스템의 오류 발생시 복구를 지원한다.



(그림 2) Chromium OS 세부 기능 구조도

Kernel은 최소화된 Patch이외에는 변하지 않는 Linux Kernel, Driver Software 및 Daemon들로 구성된다. System 및 Graphic Library는 Open Graphic Library (OpenGL)과 OpenGL Embedded System (ES)를 사용하는 응용 프로그램을 지원하는 기능, Network-Time-Protocol(NTP)/syslog/cron 등의 표준 Linux 서비스, 스크린 세이버 (xsreensaver) 기능, 저전력화를 위한 Power Management 기능, Library의 업데이트를 자동으로 하는 Autoupdate 기능, 무선 네트워크로의 연결을 위한 WPA 기능, 네트워크 (특히 3G, Wireless 등) 연결 설정과 관리를 위한 Connection Manager 기능 및 운영체제 주요 기능간 정보 교환을 위한 D-Bus 등의 기능으로 이루어져 있다.

Window Manager는 X Window Manager처럼 복수의 창에 대한 지원을 한다. Chromium 계층은 HTML5++, WebKit, Flash, Javascript 등의 기능을 지원한다.

2.1.2 주요 특징

궁극적으로 모바일 기기에서 클라우드 컴퓨팅 서버가 제공하는 작업을 주로 사용하는 기기에 최적화된 Google Chromium OS는 다음과 같은 설계철학을 가지고 있다.

첫 번째로 모든 응용 프로그램은 웹 기반의 응용 프로그램이다. 사용자는 오로지 웹 브라우저만을 사용하여 작업을 하게 되며, 데스크톱 응용 프로그램은 지원하지 않는다. 따라서 사용자는 자신의 컴퓨터에 응용 프로그램을 설치/관리하고, 업데이트 하는 작업을 하지 않는다.

두 번째로 모든 응용 프로그램들이 웹 브라우저 안에서만 구동하게 설계되어 강화된 security를 제공한다. Chrome OS는 기본적으로 OS 자신을 제외한 모든 응용 프로그램을 신뢰하지 않는 구조로 만들어져 있다. 따라서 보안이 강화된 security sandbox안에서만 각 응용 프로그램이 동작하는 구조를 통하여 컴퓨터에 대한 바이러스 혹은 malware의 공격을 효과적으로 차단하도록 설계되어 있다. Chrome OS 자체에 대해서도 부팅 시에 OS의 integrity를 매번 검사하도록 하고, 바이러스 등의 오염이 의심되면 이를 수정하고 재 부팅할 수 있도록 설계되어 있다.

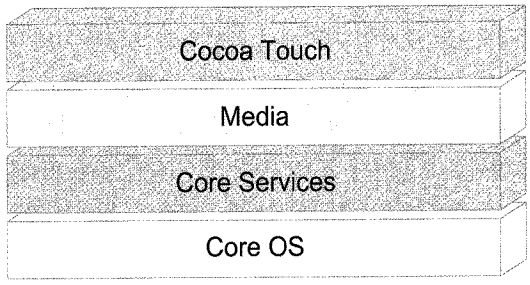
마지막으로 Chromium OS를 Google이 만든 가장 큰 이유로 삼는 것은 기존의 범용 운영체제가 너무 많은 기능을 제공하기에 크기면 에서도 커졌으며, 부팅 시간도 길어 졌다는 점이다. 따라서 웹을 통한 작업을 주로 하는 경우에 있어서 불필요한 기존 범용 운영체제의 기능을 대폭 제거/축소하였으며, 최대한 운영체제의 기능을 병렬적으로 동작하게 하여 부팅 및 운영체제 동작 시간을 줄일 수 있도록 하였다.

2.2 Apple iPhone OS

Apple iPhone OS는 휴대폰인 iPhone, MP3 플레이어인 iPod Touch, 2010년 올해 발표한 테블릿 컴퓨터인 iPad에 적용된 운영체제로서 기존 Apple의 UNIX 기반 데스크톱 컴퓨터용 운영체제인 OSX를 경량화 한 것으로 2007년 발표되었다.

2.2.1 Layered 구조와 주요 기능

iPhone OS의 내부 계층 구조도가 (그림 3)에 나타나있다 [3].



(그림 3) iPhone OS 내부 계층 구조도

최 상단의 Cocoa Touch 계층은 응용 프로그램의 구동에 필요한 infrastructure 기능을 제공하는데, 이들은 Apple Push Notification Service, Address Book UI Framework, Email 등을 위한 Message UI Framework, Map Kit Framework, Peer-to-Peer Game Kit Framework, UIKit Framework 등이다.

Media 계층은 미디어 처리를 위한 기능을 제공하는데, 이들은 Core Graphics Framework, Quartz Core Framework, Open GL ES Framework, AV Foundation Framework Foundation, Core Audio Foundation, Open Audio Library, Media Player Framework 등이다.

Core Service 계층은 가장 기본적인 시스템 서비스를 제공하며, 이들은 Address Book Foundation, Core Data Framework, Core Foundation Framework, Core Location Framework, Foundation Framework, 콘텐트와 서비스의 구매를 위한 Store Kit Framework, SQL Light Library, XML Parser 등이다.

Core OS는 운영체제의 핵심이 되는 System, Security, Network, Power Management, File System 등의 기능을 지원한다.

2.2.2 주요 특징

개인 및 가정용 서비스/콘텐트 부문에 있어서, 이미 Apple은 Mobile Me와 같은 클라우드 컴퓨팅 기반의 서비스를 유료로 제공하고 있다. Mobile Me는 이메일/주소록/일정 등의 개인 정보관리 서비스를 Apple의 클라우드 컴퓨팅 서비스에 가입하여 실행함으로써, 언제 어디서든 어떤 모바일 휴대 기기 및 유선 기기를 통해서도 접근할 수 있는 서비스다. iDisk는 웹 하드디스크 서비스로서 Mobile Me에 포함되어 제공된다. 최근 멀티미디어 콘텐트 스트리밍 업체를 인수하여 사업 분야를 확장하고 있다.

Apple이 '10년 4월 iPhone OS 4.0의 발표와 함께 영역을 더욱 확보하고자 하는 분야는 모바일을 활용하면서 iPhone/iPad 등의 thin-client를 활용하는 비즈니스 환경이다. 최근 모바일 기기와 클라우드 컴퓨팅을 결합한 모바일 클라우드 업무 환경 분야에 대한 확대를 위하여 Apple은 iPhone OS 4.0에 다음과 같은 기술을 새롭게 적용하였다 [4][5][6].

Wireless Application Distribution : 기업에서 자체 개발한 업무용 응용 프로그램을, 해당 기기를 사용하는 사용자가 인식하지 않아도 되는 자동화된 방법으로, 무선랜과 셀룰라 이동통신을 통하여, 보안이 제공되는 상태에서 안전하게 배포하는 기술

Mobile Device Management : 기업이 관리하는 다량의 기기들에 대해서 무선을 통하여 기기 제어 정보를 조정하고, 환경 설정을 업데이트하고, 회사 정책을 지키고 있는지를 모니터링하고, 특정 프로그램을 지우거나, 혹은 특정 프로그램에 대해서 사용자가 임의로 지우려는 작업을 금지하는 등의 무선 통신을 통한 단말 제어 기술

이외에도 iPhone OS 4.0은 기기 패스코드(passcode)와 암호화 키를 이용한 강화된 데이터 방어기술, 복수의 Exchange ActiveSync 계정과의 동기화를 지원하는 메일 기술, Secure Socket Layer (SSL) Virtual Private Network (VPN) 지원 기술 및 멀티태스크 기술 등을 도입하여, 모바일 비즈니스 분야에서의 iPhone OS 기반 모바일 기기의 확산을 적극적으로 추진하고 있다.

III. 모바일 클라우드 서버 SW 기술

모바일 클라우드 환경에서의 서버 기술은 현재의 유선 클라우드 컴퓨팅 환경에서의 서버 기술과 크게 다르지는 않을 것으로 예상한다. 콘텐트의 경우 무선 환경에 맞도록 재편될 것이며, 4장에서 설명할 콘텐트 가속화 기술이 서버에 내재되어 모바일 통신 환경에 최적의 성능을 제공할 수 있는

형상으로 개선될 것으로 예상한다. 클라우드 컴퓨팅 환경에서의 서버 기술은 다수의 자료들[7]에서 이미 다루어진 분야이기에, 본 고에서는 상용 플랫폼과 개방형 플랫폼으로 구분하여 간략하게 정리하고자 한다.

3.1 개방형 소프트웨어 플랫폼

Open-source 혹은 free-license의 개방형 플랫폼 소프트웨어가 클라우드 컴퓨팅의 서버 환경을 위하여 제공 되고 있다. 본 절에서는 대표적인 플랫폼인 Hadoop과 Eucalyptus, 그리고 EU RESERVOIR 플랫폼에 대해서 간략하게 알아본다.

3.1.1 Hadoop

Hadoop은 Apache 프로젝트에 포함된 free-license의 Java 기반 소프트웨어 프레임워크로서 data-intensive한 분산 응용 프로그램 환경을 목표로 한다. 수천개의 노드들과 petabyte의 데이터를 처리할 수 있도록 설계되었다. 처음 Hadoop을 개발한 Doug Cutting은 Google의 MapReduce와 Google File System (GFS)에서 영감을 받고 Hadoop을 개발한 것으로 알려져 있다. 현재 Hadoop은 다수의 대학뿐 아니라 Yahoo!, IBM, Google, Amazon, Adobe, AOL, Facebook, The New York Times, Microsoft, Twitter 등에서 널리 사용하고 있다[8].

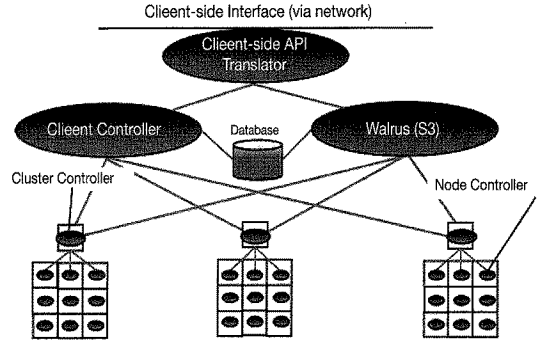
Hadoop은 현재 다수의 sub-project로 분화되어, 다양한 분야에 대한 연구 및 개발로 확산되고 있다. 현재 Hadoop에서 연구/개발하고 있는 분야는 다음의 〈표 1〉과 같다.

〈표 1〉 Hadoop 서브 과제 개요

서브과제	추진 내용
Common	Hadoop 서브 과제에 대한 common utility 기술
Avro	스크립트 언어와의 동적 통합(integration)을 지원하는 데이터 Serialization 시스템
Chukwa	대용량 분산 시스템을 관리하기 위한 데이터 수집 시스템
HBase	대규모 테이블로 구성된 구조적 데이터 저장장치를 지원하는 scalable한 분산 데이터 베이스
HDFS	어플리케이션 데이터에 대한 고성능 액세스를 지원하는 분산 파일 시스템
Hive	데이터 Summarization과 Ad-hoc 쿼리를 지원하는 데이터웨어하우스 인프라스트럭쳐
MapReduce	Compute 클러스터 상에서 대규모 데이터 셋의 분산 처리를 지원하는 소프트웨어 프레임워크
Pig	병렬 계산을 위한 상위 레벨 데이터 플로우 언어 및 수행 프레임워크
ZooKeeper	분산화된 어플리케이션을 위한 고성능 coordination 서비스

3.1.2 Eucalyptus

Eucalyptus는 Elastic Utility Computing Architecture Linking Your Programs To Useful Systems의 약자로서, Linux 기반의 Infrastructure as a Service (IaaS) 스타일의 open source 플랫폼으로서 캘리포니아 산타바바라 대학에서 클라우드 컴퓨팅에 대한 연구를 목적으로 만들어 졌다. 2008년 5월에 Release 1.0이 발표되었으며, 현재 Eucalyptus Systems 회사에서 관리하고 있으며, Eucalyptus기반 유료 솔루션을 서비스 중이다. 최근NASA에서 Eucalyptus기반으로 클라우드 컴퓨팅 시스템을 구축하기도 하였다. Eucalyptus의 플랫폼 구조도가 (그림 4)에 나타나 있다[9][11].



(그림 4) Eucalyptus 플랫폼 구조도

Node Controller (NC)는 호스트 상에서 동작하고 있는 VM 인스턴스들의 수행, 관찰 및 중단을 제어한다. Cluster Controller (CC)는 특정 NC에서 동작할 Virtual Machine (VM)의 수행을 스케줄링하고, 관련 제어 정보를 수집하는 역할을 한다. 또한 가상 인스턴스 네트워크의 관리도 함께 수행한다. Storage Controller (Walrus)는 Amazon의 S3 인터페이스를 구현하여 put/get storage 서비스를 지원한다. 이를 통하여 VM 이미지(image)와 사용자 데이터를 액세스하고 저장하는 메커니즘을 지원한다. Cloud Controller는 클라우드에 대한 사용자와 관리자의 entry-point 역할을 수행한다. 자원에 대한 정보를 노드 매니저에 문의하여 관리하고, 상위 레벨의 스케줄링 결정을 내리며, CC들에게 스케줄링 된 정보를 전달하여 수행하도록 지시하는 역할을 한다. 마지막으로 Client-side API Translator는 Amazon EC2 SOAP 인터페이스와 같은 외부 클라이언트의 인터페이스를 내부

의 Eucalyptus 시스템 인터페이스로 변환하는 기능을 수행한다.

3.1.3 RESERVOIR

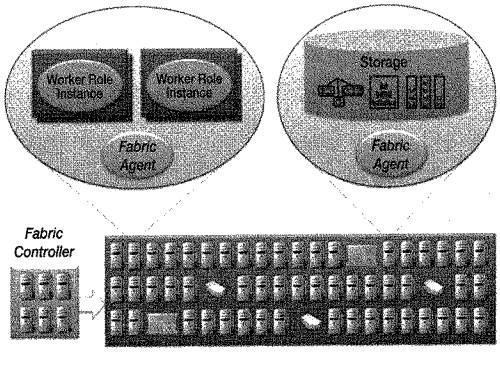
RESERVIOR은 European Union (EU) Framework Program 7 (FP7) 프로젝트로 진행 중이다. 다른 플랫폼과 다른 특징으로서 서로 다른 관리자의 통제를 받는 복수 도메인의 클라우드 간을 협력을 고려한 환경에서, 복잡도가 높은 IT 서비스의 수행과 관리를 용이하게 하는 목적으로 진행 중이다[10].

3.2 상용 소프트웨어 플랫폼

개방형 소프트웨어 플랫폼이 주로 각 기업의 서버들을 클라우드화 하는데 반하여, 상용 소프트웨어 플랫폼은 각 회사에서 컴퓨터, 운영체제 및 저장장치를 가지고 클라우드 환경을 구축한 후, 필요한 사용자에게 해당 환경을 필요한 만큼, 필요한 시간 동안 임대해 주고, 이에 대한 사용료를 받는 모델로 지원하고 있다.

3.2.1 Microsoft Azure

Microsoft는 (그림 5)와 같은 클라우드 컴퓨팅 환경을 구축하고 있으며, 사용자가 원하는 만큼의 컴퓨팅 프로세서와 저장장치를 필요한 시간만큼 사용하고, 사용량에 대하여 사용료를 내는 모델의 Azure 플랫폼을 제공하고 있다. Azure는 기존의 Microsoft의 .NET 서버, Visual Studio 개발 환경 등을 재활용할 수 있기에 Microsoft의 환경에 익숙한 사용자에게 장점이 있다[12].



(그림 5) Microsoft Azure 구조도

Microsoft가 확보한 서버들과 저장장치에 접근하기 위하여, 사용자는 본인이 필요한 컴퓨팅 프로세서와 저장장치에 대한 요구사항을 온라인 인터페이스를 통하여 Microsoft에 요청하면, Fabric Controller가 물리적인 컴퓨터와 저장장치를 묶어서 사용자가 요청한 논리적인 가상 서버 환경을 제공하여 준다. 이를 위하여 Fabric Controller는 클라우드 자원인 컴퓨터와 저장장치에 위치한 Fabric Agent와 통신하여 각 자원을 모니터링 한다. 사용자가 요청한 컴퓨팅 프로세서는 역할에 따라서 웹 어플리케이션 타입의 Web Role과 .NET 기반의 어플리케이션 타입인 Worker Role로 나뉘어 지원된다.

3.2.2 Google App Engine

Google App Engine은 2008년부터 제공된 클라우드 컴퓨팅 서비스로서 Google 인프라 상에서 사용자가 자신의 웹 서비스를 만들 수 있도록 되어 있다. 이를 위하여 Google은 〈표 2〉와 같은 다양한 API를 사용자에게 제공하고 있다[13].

〈표 2〉 Google App Engine API

API	주요 기능
Python 런타임 API	응용프로그램이 실행되는 Python 환경 정보 (CGI, 샌드박스 기능, 응용 프로그램 캐싱, 로깅)
데이터 저장소 API	확장 가능한 데이터저장소 및 이를 효율적으로 사용하는 방법에 대한 정보
이미지 API	이미지 데이터 조작 서비스
메일 API	응용프로그램에서 이메일 전송
Memcache API	분산 메모리 캐시
URL 가져오기 API	응용프로그램에서 다른 인터넷 호스트에 액세스
사용자 API	사용자 응용프로그램을 Google 계정과 통합

Software Development Kit (SDK)가 제공되어 자신의 로컬 환경에서 소프트웨어의 개발과 시험을 할 수 있으며, 테스트가 완료되면 해당 서비스를 Google로 업로드 할 수 있다. 일단 업로드된 서비스는 관리자 인터페이스를 통하여 언제 어디서든 서비스 상태, 시스템 및 데이터 로그, 방문자 이력 등을 조회할 수 있다. 실행환경은 Python에 기반하고 있으며, 추후 Java의 지원이 계획되어 있다. 인프라 내부는

Google이 확보한 서버 클러스터 상에, GFS를 통한 분산 파일 시스템이 동작하고 있으며, 이 위에 Bigtable 분산 데이터베이스를 구동하고 있다. 글로벌 메모리 캐쉬로는 Memcache가 지원된다.

IV. 모바일 클라우드 네트워크 기술

모바일 클라우드 컴퓨팅 환경은 기본적으로 유선대비 많은 문제점이 있는 무선랜 혹은 셀룰라 이동통신 기술을 활용한다. 따라서 무선 환경에서는 유선 환경에서 발생하던 대역폭, 전송 지연, 서비스 품질, 보안 등으로 인한 문제의 영향이 더 크게 발생하며, 무선에서는 없었던 이동성 등으로 인한 IP 주소 변경과 트래픽 유실 등의 문제가 발행한다. 따라서 모바일 환경에서의 클라우드 컴퓨팅 환경을 구축하는 경우에는 언급한 문제점을 해결하기 위한 네트워크 기술이 반드시 필요하다.

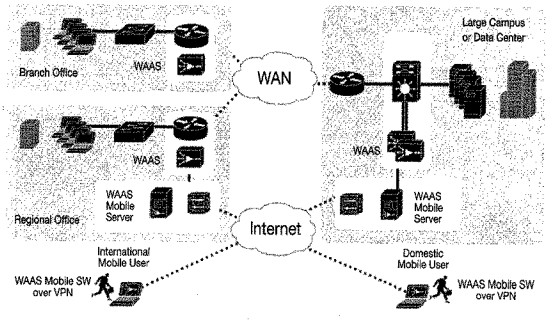
현재 클라우드 컴퓨팅 환경을 위하여 제공되고 있는 네트워크 기술/제품은 Cisco가 대표적이며, 소프트웨어 혹은 하드웨어와 소프트웨어의 결합된 제품으로 제공된다. 본 장에서는 모바일 클라우드에 특화된 기술은 아니지만, 현재 클라우드 컴퓨팅 환경을 위하여 제시되고 있는 기술을 살펴봄으로써 향후 모바일 클라우드 환경에서 고려되어야 하는 기술들에 대하여 살펴보고자 한다. 향후 모바일 클라우드가 일반화되고 활성화되면, 단지 클라우드 서버와 단말의 소프트웨어뿐만 아니라, 네트워크에서도 성능 개선과 향상을 위한 기술이 핵심 분야로 부상할 것으로 본다.

참고로 가상화된 컴퓨터 자원을 이용하는 클라우드 컴퓨팅과 더불어 네트워크를 가상화 하는 이슈에 대해서도 많은 연구가 이루어 지고 있으며, 향후 모바일 클라우드 컴퓨팅 환경과 함께 연계되어 기술 발전이 활발하게 이루어 질 것으로 예상한다. 이에 대한 사항은 참조문헌 [14]를 참조하기 바란다.

4.1 Wide Area Network (WAN) 가속화 기술

클라우드 컴퓨팅 환경은 필연적으로 원격지에 위치한 서버를 WAN 기술을 통하여 thin-client가 액세스 하는 형상을 가진다. 지금까지의 인터넷 서비스를 살펴보면 고속의 유선 통신 기술과 Local Area Network (LAN) 수준의 성능 제약을 고려한 응용 프로그램 개발이 이루어져 왔다. 즉 넓은 대역폭에서 고속의 통신을 지원하면서, 낮은 지연 시간과 매우 높은 통신 품질(낮은 에러율 등)을 보장 받았다. 하지만 클라우드 컴퓨팅의 경우는 thin-client가 원격지에 위치한 클라우드 서버를 액세스함으로써, 원격지의 클라우드 서버와 thin-client를 연결하는 WAN의 품질에 따라서 (LAN 대비 상대적으로) 작아진 대역폭과 저속 통신, 높은 지연, 그리고 높아진 에러율의 특성을 가진다. 특히, 셀룰라 이동통신을 활용하는 모바일 클라우드 환경에서는, 이동통신의 특성상 peak/mean/minimum 전송 속도에서 통신 속도가 동적으로 변하며, 동적인 통신 속도와 이동통신 네트워크에서의 처리 부하로 인한 지연이 증가하고, 무선 상황에 따라서 데이터의 유실이 유선과 비교할 수 없을 정도로 발생한다.

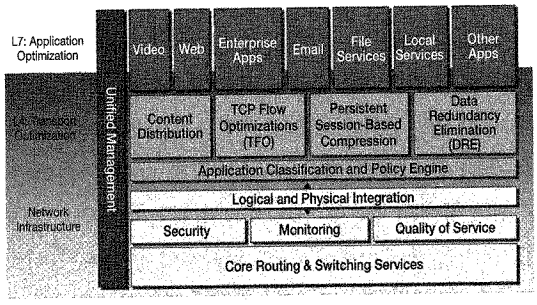
Cisco는 이러한 환경을 위하여, Wide Area Application Service (WAAS) 솔루션을 제공하고 있다. (그림 6)에 나타나듯이, WAAS는 클라우드 서버와 thin-client 혹은 복수의 장소에 분리된 클라우드 서버들간의 통신 속도를 향상하는 기술로서, 기존 장비에 독립적으로 동작하도록 되어 있다.



(그림 6) Cisco WAAS 구성도

WAAS는 (그림 7)과 같이 Transport 계층에서의 성능 개선을 주 목적으로 한다.

가장 중요한 기능은 TCP의 성능을 개선하는 것으로서 TCP Proxy와 Flow Optimization을 통하여 원격지점간의 TCP 통신에서 TCP 윈도우를 스케일링하고, 고속의 전송이 가능하도록 초기 윈도우 값을 큰 값으로 유지하게 하고, WAN에서의 혼잡구간과 손실 발생시 클라우드와 서버가 이

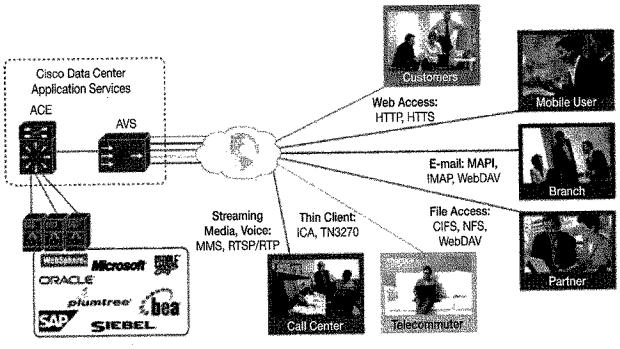


(그림 7) Cisco WAAS 내부 계층 구조도

를 인식하지 못하도록 함으로서 성능 저하를 방지하는 기능 등을 제공한다. 다음으로 중요한 기능은 Data Redundancy Elimination (DRE) 기술로서 네트워크 상에서 응용 프로그램에 독립적으로 redundant한 정보를 제거하여 TCP와 같은 전송 계층의 트래픽을 압축하는 기술이다[15].

4.2 어플리케이션 가속화 기술

앞서 4.1에서는 TCP를 중심으로 하는 전송 계층에 대한 성능을 개선하는 것을 소개하였다. TCP 보다 높은 상위 계층에 대한 성능 개선 기술에 대해서 Cisco는 (그림 8)에 나타난 것과 같이, Application Control Engine (ACE)과 Application Velocity System (AVS)이라는 장치를 제공하고 있다[16].



(그림 8) Cisco ACE/AVS 장비 적용도

주요 기능은 Web 페이지에 대하여 업데이트가 이루어진 부분만 전송하는 Delta encoding, 다운로드 속도 개선을 위한 Dynamic browser caching, 이미지 파일을 네트워크와 단말에 최적화 하는 Smart image optimization, 전송하는 콘텐트의 크기를 transparent하게 줄여주는 Compression 등이다. 이외에도 TCP connection multiplexing, High performance caching, Adaptive and configurable caching, Load-based dynamic caching, Lazy request evaluation, SSL acceleration, URL mapping, Single-sign-on optimization, XML transformation 등의 기능을 제공하여 응용 프로그램이 원격지의 클라우드 서버와 thin-client에서 동작하는 경우의 성능을 향상시키고 있다.

V. 결 론

현재 클라우드 컴퓨팅 환경은, 기업이 자체적으로 확보하고 있는 인프라를 기반으로 구축하는 private cloud와 Microsoft/Google 등 외부 기업의 인프라를 임대하여 사용하는 public cloud로 나뉘어 있다. 이미 개인 사용자들의 public cloud 서비스에 대한 이용은 웹 기반 메일/일정관리 등으로 일반화 되어 있으며, 웹 기반 하드디스크와 온라인 워드프로세서, 온라인 스프레드 쉬트 프로그램 등의 사용도 웹기반 메일과 연계하여 증가하고 있는 추세이다. 기업의 경우도 보안과 운용비용의 감소를 고려하여 private cloud 형태 중심으로 점차 시스템을 변화시키고 있는 상황이다.

이러한 추이를 반영할 때, LTE/LTE-Advanced 및 Mobile-WiMAX 16m과 같은 4세대 무선 통신 기술의 상용화와 함께, 노트북/iPhone/iPad와 같은 스마트 모바일 기기의 확산에 힘입어, 향후 모바일 환경에서의 클라우드 컴퓨팅 기술에 대한 관심은 급성장할 것으로 예상한다. 이를 위하여 본고에서는 모바일 클라우드 컴퓨팅 환경에의 단말, 서버 및 네트워크 소프트웨어 핵심 기술에 대하여 알아보았다.

참 고 문 헌

[1] Marinelli, Eugene E., "Hyrax: Cloud Computing on Mobile Devices using (Master Thesis)", September 2009.

(http://oai.dtic.mil/oai/oai?&verb=getRecord&metadataPrefix=html&identifier=ADA512601)

[2] The Chromium Projects, "Software Architecture", November 2009. (http://sites.google.com/a/chromium.org/dev/chromium-os/chromiumos-design-docs/software-architecture)

[3] Apple Inc., "iPhone OS Technology Overview - General", 19 October 2009. (http://developer.apple.com/IPhone/library/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechOverview.pdf)

[4] Apple Inc., "iPhone OS Enterprise Deployment Guide Second Edition", April 2010. (http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdf)

[5] Apple Inc., "iPad in Business Deployment Scenarios and Device Configuration Overview", April 2010. (http://images.apple.com/ipad/business/pdf/iPad_Deployment_Scenarios.pdf)

[6] Apple Inc., "New Enterprise Features in iPhone OS 4". (http://www.apple.com/iphone/business/).

[7] 한재선, "클라우드 컴퓨팅 플랫폼과 오픈 플랫폼 기술", 한국정보처리학회 정보처리학회지, 제16권 제2호, pp.39~50, 2009년 3월.

[8] Apache Hadoop, (http://hadoop.apache.org/#What+Is+Hadoop%3F)

[9] Eucalyptus Systems. (http://www.eucalyptus.com/)

[10] RESERVOIR - Resources and Services Virtualization without Barriers. (http://62.149.240.97/)

[11] Daniel Nurmi, and etc., "The Eucalyptus Open-source Cloud-computing System", Cloud Computing and Its Applications 2008, 22~23 October 2008. (http://www.cca08.org/papers/Paper32-Daniel-Nurmi.pdf)

[12] David Chappell & Associate, "Introducing Windows Azure", December 2009. (http://download.microsoft.com/download/0/C/0/0C051A30-F863-47DF-BC53-9C3CFA88E3CA/Windows Azure David Chappell White Paper March 09.pdf)

[13] Google App Engine, (http://code.google.com/intl/ko-KR/appengine/)

[14] 김영화, "미래인터넷의 네트워크 가상화 기술 동향", 한국전자통신연구원 전자통신동향분석 제25권 제1호 2010년 2월. (http://ettrends.etri.re.kr/PDFData/25-1_132_147.pdf)

[15] Cisco Systems, "Cisco Wide Area Application Services (WAAS) Software Version 4.1 Technical Overview", 2008. (http://www.cisco.com/en/US/prod/collateral/contnetw/ps5680/ps6870/prod_white_paper0900aecd8051d5b2.pdf)

[16] Cisco Systems, "Application Infrastructure Control, Performance, Security, and Infrastructure Simplication with Cisco Application Control Engine and Application Velocity System", 2006. (http://www.ciscosystems.net/en/US/prod/collateral/modules/ps2706/ps6906/prod_white_paper0900aecd80458698.pdf)

약 력

1998년 경희대학교 박사
1999년 ~ 2008년 삼성전자 정보통신총괄 네트워크사업부/통신연구소
2008년 ~ 현재 경희대학교 컴퓨터공학과 교수
관심분야 : 무선광대역통신, 컴퓨터네트워크, 미래인터넷, M2M/MTC 통신시스템

이 성 원

2009년 경희대학교 학사
2009년 ~ 현재 경희대학교 컴퓨터공학과 석사과정
관심분야 : 미래인터넷, 모바일클라우드컴퓨팅

김 현 준