

특집  
03

## 클라우드 컴퓨팅 플랫폼과 오픈 플랫폼 기술

## 목 차

1. 서 론
2. 클라우드 컴퓨팅 플랫폼 요소 기술
3. 클라우드 컴퓨팅 플랫폼 사례
4. 오픈 플랫폼 기술
5. 결 론

한 재 선  
(NexR)

## 1. 서 론

전기, 운송, 컴퓨팅 등 범용기술의 초기 단계에서는 낮은 기술 수준과 유통망 부재 등의 이유로 필연적으로 분산된 형태로 공급되는게 일반적이다. 즉, 범용기술이 필요한 기업이 직접 기술을 공급하는데 필요한 설비를 구축하고 운영해야 한다. 하지만 분산 공급은 많은 자본 투자와 큰 고정비가 소요되는 만큼 비경제적이다. 따라서 범용기술의 발전과 함께 중앙 집중형의 공급이 가능해지면 “규모의 경제”를 극대화할 수 있는 대규모 유틸리티 방식으로 산업의 패러다임이 바뀌게 된다. 전기 산업의 변화가 그 역사적 증거이며 IT 기술의 성숙과 함께 컴퓨팅 산업 역시 유사하게 진화할 것으로 예상된다. 이러한 비전과 함께 최근 주목받고 있는 기술이 클라우드 컴퓨팅이다[1].

기존 범용기술 사례에서도 알 수 있듯이 결국 클라우드 컴퓨팅의 핵심 경쟁력은 “규모의 경제”를 얼마나 더 효율적으로 실현하느냐 문제로 귀결되고 그 기반이 되는 것이 플랫폼 기술의 경쟁력이라 할 수 있다. 클라우드 컴퓨팅 플랫폼이

라 함은 개인이나 기업에게 컴퓨팅 자원을 서비스 형태로 제공해 줄 수 있게 해 주는 기반 HW 및 SW를 말한다. 이미 Amazon.com EC2, S3 서비스[2]와 Google App Engine[3] 등이 입증하고 있듯이 탄탄한 플랫폼 기술이 뒷받침되어야 안정적인 서비스와 함께 저렴한 가격정책으로 경쟁 우위를 점할 수 있다. Microsoft 역시 최근 Windows Azure 서비스 플랫폼[4]을 오픈하면서 클라우드 컴퓨팅 플랫폼 경쟁력을 강화하고 있는 상황이다.

클라우드 컴퓨팅 플랫폼[5]은 다양한 요소 기술이 유기적으로 결합된 형태로 구성된다. 자원 가상화 기술(Server Virtualization, Storage Virtualization, Network Virtualization 등), 대용량 분산 시스템 기술(분산 파일 시스템, 분산 병렬 처리, 분산 데이터 저장소 등), SOA(Service Oriented Architecture) 기반의 클라우드 인터페이스 기술, AC(Autonomic Computing) 및 통계 기법 기반의 클라우드 자원 운영 관리 기술, 클라우드 SLA(Service Level Agreement), 클라우드 보안 및 프라이버시 기술 등이 핵심적인 요소 기술에 해당하며 이들을 어떻게 하나의 플랫폼

폼으로 통합하느냐 역시 중요한 기술 경쟁력이라 할 수 있다.

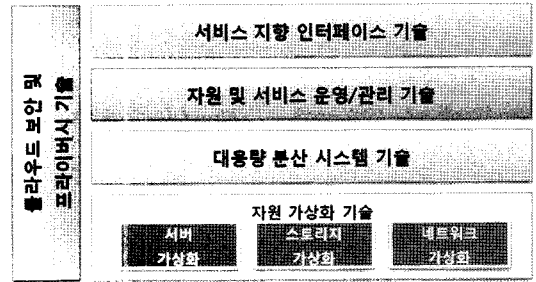
이러한 플랫폼 경쟁력을 확보한 대표적인 사례가 Amazon과 Google로서, Amazon.com의 Amazon Web Services 플랫폼은 2002년부터 쌓아온 실제 운영 경험이 강점이며 Google App Engine 플랫폼은 Google 검색엔진의 검증된 플랫폼을 공유함으로써 안정성과 확장성을 보장할 수 있었다. 최근 등장한 Microsoft의 Windows Azure 플랫폼 역시 Microsoft의 플랫폼 기술력의 총합으로 앞으로 주목해 볼 만하다. 이들 플랫폼이 클라우드 컴퓨팅 기업들 소유의 단허진 플랫폼이라면 오픈소스 커뮤니티 주도로 오픈 플랫폼 형태의 클라우드 컴퓨팅 플랫폼들도 속속 등장하고 있다. Apache 오픈소스 프로젝트인 Hadoop [6]이 가장 대표적인 예로서 대규모 서버 클러스터 환경에서 대용량 데이터의 분산 저장 및 병렬 처리를 지원하면서 Yahoo!, Face-book 등 많은 기업들의 기본 플랫폼으로 채택되고 있다. 이외에도 대학 및 연구소를 중심으로 Eucalyptus[7], OpenNebula[8], Reservoir[9], Enomaly ECP [10] 등이 쏟아져 나오고 있으며 다양한 클라우드 컴퓨팅 서비스의 기반이 되고 있다.

본 고에서는 Amazon, Google, Microsoft 세 기업의 클라우드 컴퓨팅 플랫폼 기술과 특징에 대해 살펴보고 오픈 플랫폼의 현황과 특징에 대해 분석해 본다.

## 2. 클라우드 컴퓨팅 플랫폼 요소 기술

컴퓨팅 기술 발전의 관점에서 보면 클라우드 컴퓨팅은 전혀 새로운 개념이 아니다. 클러스터 컴퓨팅부터 시작하여 최근의 그리드 컴퓨팅, 유틸리티 컴퓨팅, 서비스 지향 컴퓨팅 등의 핵심 철학이 하나로 녹아들어 클라우드 컴퓨팅을 탄생시킨 것이다. 필연적으로 클라우드 컴퓨팅 플랫폼의 기술적 토대는 현존하는 여러 기반 기술로 구성되었음을 쉽게 짐작할 수 있다. (그림 1)

은 주요 클라우드 컴퓨팅 플랫폼의 공통적인 기술 구조를 나타낸다.



(그림 1) 클라우드 컴퓨팅 플랫폼의 기술 구조

### 2.1 자원 가상화(Virtualization) 기술

컴퓨팅 자원의 활용률을 극대화하고 대규모 데이터센터의 관리 편의성을 확보하기 위해서 자원을 가상화하여 운용하고 관리하는 자원 가상화 기술은 가장 기본적인 기술이라 할 수 있다. 그래서 VMware와 Citrix 등 서버 가상화 솔루션 업체가 클라우드 컴퓨팅 솔루션 시장에 적극적으로 뛰어들고 있다.

### 2.2 대용량 분산 시스템 기술

수많은 서비스와 대규모 사용자를 기본 가정으로 하고 있으므로 대용량 저장능력과 고성능 컴퓨팅 파워를 제공하고 신속한 확장성을 보장할 수 있는 분산 시스템 기술이 필요하다.

### 2.3 자원 및 서비스 운영/관리 기술

서비스 및 사용자의 동적인 변화에 능동적으로 대응하여 자원을 관리하고 글로벌 규모로 서비스를 쉽게 배포 및 관리하는 기술이 필요하다. Autonomic Computing, SLA, Workflow Management, Global Service Provisioning 등이 여기에 포함된다.

### 2.4 서비스 지향 인터페이스 기술

클라우드 컴퓨팅은 “as a Service” 형태의 기능 접근 방법을 제공한다. 따라서 자원에 대한

접근부터 플랫폼 자체에 대한 제어까지 모두 서비스 지향 인터페이스로 정의되고 구현되어야 한다. 추후 클라우드간 연동을 고려할 때 반드시 처음부터 고려해야 할 요소이다.

## 2.5 클라우드 보안 및 프라이버시 기술

클라우드 컴퓨팅으로 인한 데이터와 서비스의 중앙집중화는 악의적인 공격에 더욱 치명적인 결과를 초래할 수 있으며 여러 서비스에 의해 자원이 공유되는 환경은 Intra-Cloud에서 보다 강력한 보안 및 프라이버시 기술을 필요로 한다.

## 3. 클라우드 컴퓨팅 플랫폼 사례

### 3.1 Amazon EC2 플랫폼

Amazon EC2는 사용자에게 가상의 컴퓨팅 자원을 제공하고 사용한만큼 비용을 청구하는 서비스다. 비용은 한 시간 단위로 계산되며 가장 기본 단위의 컴퓨팅 인스턴스를 한 시간 사용할 경우 단지 \$0.10가 청구된다. EC2의 세 가지 기술 컴포넌트는 다음과 같다.

#### 3.1.1 EC2 인스턴스(Instance)

OS와 어플리케이션이 실행되는 최소 컴퓨팅 자원 단위로서 Xen 기반의 가상머신이다. 인스턴스의 종류는 웹 서비스와 같이 보통 어플리케이션에 적합한 사양의 표준 인스턴스와 복잡한 계산 응용을 위한 High CPU 인스턴스로 나뉘며 각 인스턴스의 크기는 가상머신 생성시 CPU, 메모리, 디스크 등 자원 할당을 조절하면서 결정된다. 가장 기본적인 인스턴스는 1개의 가상 코어(1.0-1.2 GHz 2007 Opteron 혹은 2007 Xeon 프로세서에 해당), 1.7 GB 메모리, 160 GB 디스크 용량을 가진 32 bit 머신에 해당한다.

#### 3.1.2 Amazon Machine Image (AMI)

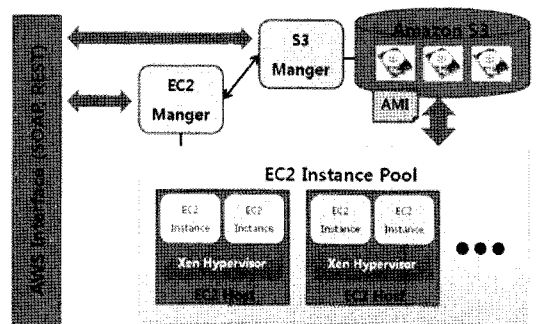
AMI는 OS와 어플리케이션을 포함한 부팅 가능한 루트 디스크 이미지다. AMI는 Amazon에

의해 미리 패키징된 것들과 사용자와 파트너사에 의해 필요에 따라 적절한 어플리케이션들로 패키징된 것들이 존재한다. AMI에서 지원하는 OS는 각종 Linux 배포판부터 Windows Server 2003, OpenSolaris 등 다양하며 이미 제공되고 있는 AMI에는 Apache 웹 서버부터 데이터베이스, WAS, 개발환경, Hadoop 등 다양한 어플리케이션 등이 패키징되어 있다.

#### 3.1.3 Simple APIs

“As a Service”는 표준 인터넷 프로토콜을 통해 제공된 모든 기능에 접근할 수 있음을 의미한다. EC2 역시 EC2 인스턴스와 AMI를 컨트롤하고 관리하는 대부분의 기능이 SOAP API와 HTTP Query API로 구현되어 있다. 따라서 기본적으로 제공되는 Command Line 도구 외에 외부 웹 서비스나 어플리케이션을 통해 EC2 서비스에 접근 가능하다.

이와 같은 컴포넌트들은 Amazon 플랫폼 위에서 서로 유기적으로 결합되고 운영된다. Amazon 플랫폼은 다음 (그림 2)와 같다.



(그림 2) Amazon EC2 플랫폼 구조

각 EC2 호스트는 Xen Hypervisor[11]가 기본으로 설치되어 있고 그 위에 EC2 인스턴스로서 두 개의 가상머신이 실행된다. EC2 인스턴스들은 풀(Pool)을 형성하여 EC2 매니저에 의해 관리된다. 수천 수만의 인스턴스를 관리하기 위해선 효율적인 가상머신 라이프사이클 관리 기능

이 필요하며 개개 EC2 호스트의 호스트 프로그램과 긴밀한 협조 하에 모니터링 및 자원 관리 기능이 필요하다. EC2 매니저는 인스턴스 관리, AMI 관리, Security Group 및 Key-pair 관리, 사용자 요청 처리 등의 핵심적인 기능을 담당한다. AMI는 영속적인 저장소인 Amazon S3에 저장 및 관리되고 인스턴스를 생성할 때 EC2와 연동하여 사용된다. S3는 수 만대의 서버에 대량의 데이터를 안전하게 저장해 주는 서비스를 제공한다. Amazon 플랫폼의 강점은 서버 가상화 기술을 활용한 효율적인 컴퓨팅 자원 관리와 대용량 데이터 저장 시스템, 표준 웹 API를 통한 인터페이스, 그리고 이들을 유기적으로 결합하는 관리 및 운영 시스템이라 할 수 있다.

### 3.2 Google App Engine 플랫폼

Google App Engine은 2008년 4월 시작한 클라우드 컴퓨팅 서비스로서 사용자 개발 웹 서비스를 구글 인프라 위에서 실행할 수 있는 인프라 자원을 제공할 뿐 아니라 웹 서비스를 개발할 수 있는 SDK와 서비스 관리 도구 등도 함께 제공하

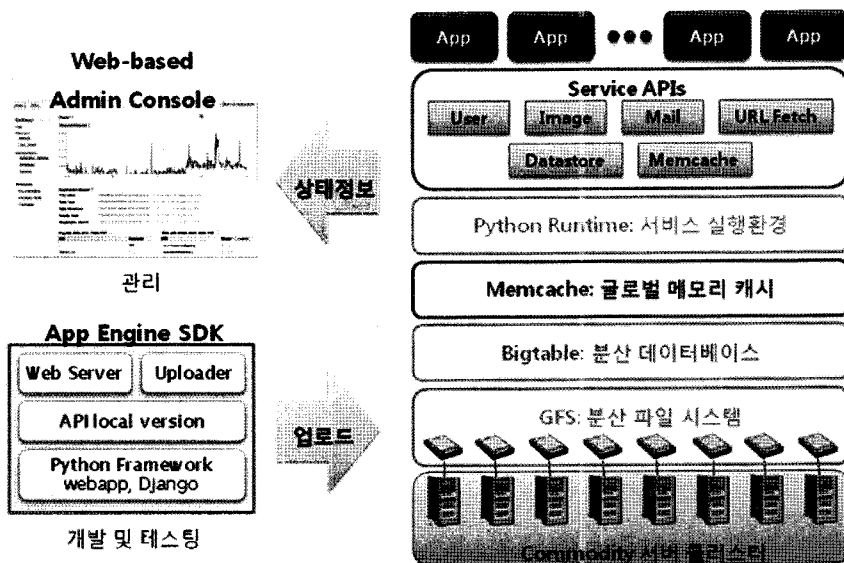
는 클라우드 플랫폼이다. 현재 500 MB의 저장공간과 월 500만 페이지뷰 정도의 Bandwidth를 무상으로 제공하고 있으며 추후 추가적인 자원을 구매할 수 있는 모델을 도입하겠다고 한다. 이미 다양한 서비스를 통해 검증된 Google 인프라를 활용하므로 확장성과 안정성 측면에서 개발자는 부담을 덜 수 있게 되고 더욱이 웹 서비스 개발 환경을 제공하기 때문에 서비스 개발부터 배포, 운영까지 전 과정을 Google App Engine에서 처리할 수 있다. Google App Engine 플랫폼은 다음 (그림 3)과 같고 5가지 핵심적인 특징을 가지고 있다.

#### 3.2.1 확장성 있는 서비스 인프라스트럭처

사용자 개발 서비스들은 Google 서비스와 동일한 인프라 기술 위에서 실행되기 때문에 서비스의 확장성과 안정성은 걱정할 필요가 없다. (분산 파일 시스템으로 GFS[12], 분산 데이터 저장소로 Bigtable[13], 글로벌 메모리 캐시로 Memcache[14] 등 제공)

#### 3.2.2 Python 런타임 환경과 다양한 서비스 APIs

현재 Python 실행 환경을 제공하고 있으며



(그림 3) Google App Engine 플랫폼 구조

MVC 모델을 비롯하여 보다 효율적인 개발을 돕기 위해 Python 웹 프레임워크인 webapp와 Django를 제공하고 있다. 추후 Java 등의 다른 언어에 대한 지원도 약속하고 있다. 그리고 서비스에 필요한 여러 가지 기능들(인증, 이미지 처리, 메일, 데이터 접근 등)을 미리 구현하여 API 형태로 제공하고 다양한 Google API와 연동을 지원하여 서비스의 확장을 돕고 있다.

### 3.2.3 Software Development Kit (SDK)

SDK에는 Google App Engine을 접속할 필요없이 로컬에서 모든 개발과 테스트를 할 수 있도록 내장 웹 서버, API들의 로컬 버전과 웹 프레임워크 등이 기본적으로 탑재되어 있다. 또한 개발된 서비스를 App Engine 사이트에 올리기 위한 업로더도 제공한다. 이는 개발자에게는 편리한 개발환경의 장점을, Google 입장에서는 개발 및 테스트 부하를 개발자쪽으로 넘기는 장점을 준다.

### 3.2.4 웹 기반 서비스 관리 도구

웹 서비스의 전체 라이프사이클을 지원하기 위해서는 서비스 운영을 관리하는 기능이 필수적이다. 이를 위해 App Engine은 웹 기반의 Admin Console을 제공하여 언제 어디서나 웹 브라우저를 통해 서비스 상태, 사용자 방문 이력, 시스템 로그, 데이터 관리 등의 관리 작업을 처리할 수 있게 한다.

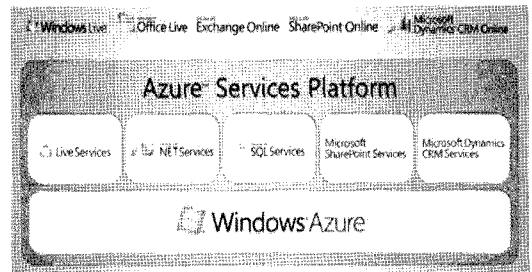
### 3.2.5 확장성 있는 데이터 저장소

수 많은 서비스들이 동일한 인프라를 공유하는 환경에서는 확장성 있는 데이터 저장소가 반드시 필요하다. 이를 위해서 App Engine은 기존 관계형 데이터베이스 대신 Google의 분산 데이터 저장소인 Bigtable을 기본 저장소로 사용하여 확장성 문제를 해결하였다. 대신 개발자는 SQL 인터페이스 대신 datastore API와 GQL(Google Query Language)의 새로운 데이터 인터페이스를 사용해야 한다.

정리해 보면 Google App Engine의 강점은 이미 검증된 Google의 서비스 인프라를 최대한 활용하여 확장성과 안정성을 확보하였다는 것과 로컬 환경에서 개발과 테스트 작업을 마칠 수 있는 개발 도구를 제공하고 있는 점, 그리고 Google의 다양한 API와 통합을 통해 시너지를 극대화시킨 것이라 할 수 있다.

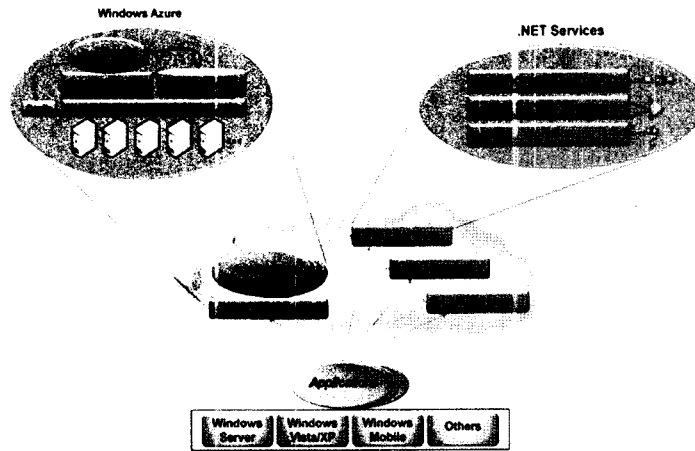
## 3.3 Microsoft Azure 서비스 플랫폼

Azure 서비스 플랫폼은 2008년 10월 Microsoft의 기술 컨퍼런스인 PDC에서 처음 발표된 클라우드 컴퓨팅 플랫폼이다. Azure 플랫폼이 목표로 하는 것은 Amazon 스타일의 "Infrastructure as a Service"[15]가 아니라 Google App Engine처럼 "Platform as a Service"[16] 시장을 바라보고 개발되었다. 현재 Preview 단계이며 신청자에 한해 베타 테스트 할 수 있는 권한을 제공하고 있으며 2009년 후반쯤 정식 오픈 예정이다. 웹 어플리케이션의 개발과 운영을 지원하는 Web Role 서비스 타입을 지원한다는 점에서 Google App Engine과 유사하지만 추가로 .NET 기반의 어플리케이션을 클라우드 환경에서 제공하기 위해 Worker Role 서비스 타입을 지원한다는 차이점이 있다.



(그림 4) Microsoft Azure 서비스 플랫폼

(그림 4)은 Azure 서비스 플랫폼의 구성[4]을 나타낸다. Windows Azure는 Microsoft 데이터 센터의 수많은 서버를 통합된 단일 시스템으로 묶어 컴퓨팅 서비스와 스토리지 서비스를 제공



(그림 5) Windows Azure 서비스 플랫폼의 구조

해 주는 클라우드 컴퓨팅 운영체제라 할 수 있다. 그리고 그 위에 클라우드 서비스의 개발을 돕는 다섯 가지 서비스들이 제공된다. 또한 클라이언트 라이브러리와 Visual Studio 개발 도구를 제공하여 클라이언트 환경에서 클라우드 서비스를 개발할 수 있게 지원하고 있다. (그림 5)는 Azure 서비스 플랫폼의 구조[17]를 나타낸다.

### 3.3.1 Windows Azure Compute 서비스

Windows Azure는 클라우드에 적합하도록 설계된 Hypervisor 위에 각 어플리케이션 인스턴스를 위해 가상머신(VM)을 제공한다. 현재 웹 어플리케이션을 위한 Web Role 인스턴스와 .NET 기반 어플리케이션을 위한 Worker Role 인스턴스, 두 가지 인스턴스를 지원하고 개발자의 VM 이미지를 구동하는 것은 지원하지 않는다. Windows Azure의 초기 버전은 VM과 물리적인 프로세서 코어 간 1:1 관계를 유지하여 어플리케이션에게 일관된 성능을 보장한다.

### 3.3.2 Windows Azure Storage 서비스

스토리지 서비스는 클라우드 서비스를 위해 간단하고 확장 가능한 스토리지를 제공한다. 세 가지 타입의 데이터구조, 즉 이미지, 동영상 등의 바이너리 데이터를 위한 Blob(Binary Large

Object), Windows Azure 어플리케이션 간의 커뮤니케이션을 위한 Queue, 엔티티의 계층구조를 표현할 수 있는 Table을 제공한다. Azure 스토리지 서비스에서 제공하는 테이블은 SQL을 사용하지 않고 LINQ[18] Syntax의 직관적인 쿼리 언어를 사용한다. 또한 어플리케이션들은 HTTP RESTful 접근을 통해 스토리지 서비스를 이용할 수 있다.

### 3.3.3 Access Control

사용자의 어플리케이션 접근 제어에 대한 통일된 방법을 제공하고 회사간 Identity 영역을 넘어 접근 제어 기능을 사용할 수 있도록 Identity Federation을 지원한다. 이를 위해 복수의 claim으로 구성된 SAML(Security Assertion Markup Language)[19] 기반의 token 기능을 구현하고, Rule 기반의 claim transformation을 통해 Identity Federation을 지원하며 이들 token의 생성과 관리를 담당하는 STS(Security Token Service)를 제공한다.

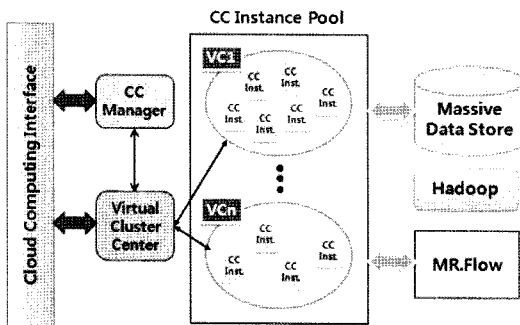
### 3.3.4 Service Bus

어플리케이션의 서비스를 외부에 노출시켜 다른 회사나 조직에서 접근 가능할 수 있는 간편한 방법을 제공한다. Service Bus는 Registry 서비

스를 통해 어플리케이션이 Endpoint를 등록하여 외부에서 찾을 수 있도록 하며 각 요청은 Service Bus에 의해 받아져 어플리케이션에게 전달된다.

### 3.4 NexR 데이터 클라우드 플랫폼

NexR[20]은 플랫폼 개발 및 구축 전문회사로 대용량 데이터 저장 및 처리 플랫폼과 클라우드 컴퓨팅 플랫폼을 전문으로 하고 있다. NexR 플랫폼의 특징은 Xen 기반의 서버 가상화 인프라와 Hadoop 분산 시스템을 유기적으로 통합하여 데이터 처리 및 분석에 최적화된 데이터 클라우드 플랫폼을 구성한다는 것이다. (그림 6)는 NexR 데이터 클라우드 플랫폼의 구조를 나타낸다.



(그림 6) NexR 데이터 클라우드 플랫폼 구조

#### 3.4.1 Virtual Cluster Center(VCC)

각 클라우드 컴퓨팅(CC) 인스턴스들은 하나의 가상머신으로 맵핑되며 사용자나 어플리케이션의 요구사항에 따라 복수의 인스턴스들이 가상 클러스터(VC)를 구성하여 마치 하나의 독립적인 서버 클러스터처럼 운영된다. VCC는 CC 인스턴스 관리와 가상 클러스터의 운영을 담당하며 CC Manager의 요청에 따라 인스턴스 풀을 제어한다.

#### 3.4.2 CC Manager

사용자와 시스템 사이의 기능적인 인터페이스를 담당하여 사용자의 요청에 따라 VC 생성 및

해제를 하고 Metering & Billing 등을 처리한다.

#### 3.4.3 MR.Flow

Hadoop MapReduce 모듈들을 Drag-and-Drop 방식으로 조합하여 복잡한 데이터 처리를 가능케 해 주는 서비스로 분산 프로그래밍에 익숙치 않은 데이터 분석가들이 쉽게 데이터 처리 워크플로우를 개발할 수 있게 해 준다.

#### 3.4.4 HadoopSource

Hadoop MapReduce 프로그램을 등록하여 공유할 수 있게 하는 일종의 디렉토리 서비스로서 외부 개발자에 의해 개발된 데이터 처리 프로그램을 활용할 수 있게 해주며 프로그램의 실행과 관련된 메타데이터를 표준 XML 형태로 기술, MR.Flow 뿐 아니라 다른 데이터 처리 서비스에서 접근 가능하도록 지원한다.

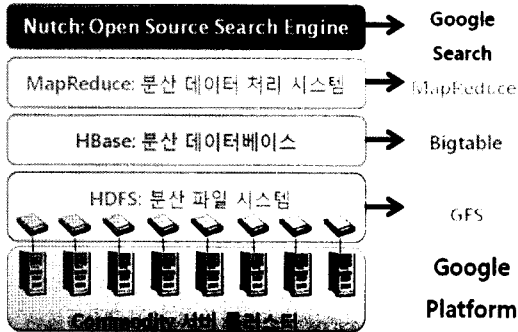
NexR 데이터 클라우드 플랫폼은 로그 분석, 데이터 마이닝, 데이터웨어하우징, BI(Business Intelligence), Knowledge Discovery 등의 데이터 처리 응용을 클라우드 서비스화하는데 최적화되어 있으며 특히 대규모 컴퓨팅 자원이 요구되는 대용량 데이터를 저장하고 처리할 수 있도록 지원하고 있다.

## 4. 오픈 플랫폼 기술

### 4.1 Hadoop 플랫폼

Hadoop은 오픈소스 클라우드 컴퓨팅 플랫폼의 대표주자로서 이미 Yahoo!, Facebook, Amazon, IBM, NexR 등 많은 기업들에서 활용하면서 그 가치를 인정받고 있다. Google의 분산 플랫폼이 검색엔진을 분산화하는 과정에서 개발되어 그 응용범위가 넓혀진 것처럼 Hadoop 역시 Lucene과 Nutch 등의 오픈소스 검색엔진의 분산화를 위해 시작되었고 최근엔 그 활용범위가 대용량 시스템으로 확대된 케이스다. 초기 개발 단계에서부터 Google GFS, MapReduce[21] 등

을 모델로 했기 때문에 Google 플랫폼과 거의 유사한 방식으로 동작한다. (그림 7)은 Hadoop 플랫폼의 구조를 나타낸다.



(그림 7) Hadoop 플랫폼 구조

4.1.1 Hadoop Distributed File System(HDFS)[22]

대용량 데이터를 저장할 수 있는 분산 파일 시스템으로 수천대 규모의 저가 서버 클러스터를 묶어 단일 파일 시스템 이미지를 제공하여 비용 절감 효과와 함께 뛰어난 확장성을 보장한다. 특히 데이터 안정성을 보장하기 위해 최소 세 개의 복사본을 유지하며 대용량을 커버하기 위해 64 MB의 큰 블록 단위를 가지고 있는 것이 특징이다.

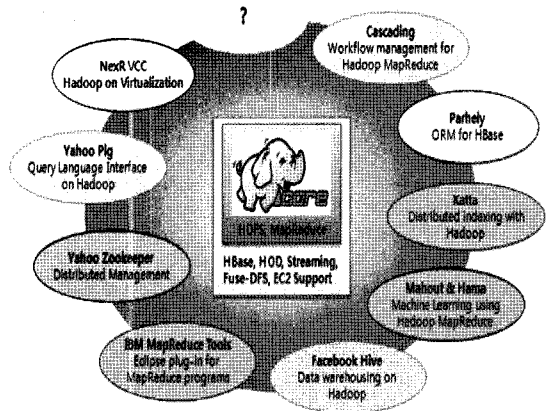
4.1.2 HBase[23]

HDFS 기반의 분산 데이터 저장소로서 기존 관계형 데이터베이스와 달리 컬럼 기반의 Key-Value 방식의 저장방식을 채택하고 있으며 메모리와 디스크를 동시에 활용하는 하이브리드 구조로 구현되어 있다.

4.1.3 MapReduce

분산 데이터 처리 시스템으로 HDFS에 분산 저장되어 있는 데이터를 map() 과 reduce()라는 간단한 분산 프로그래밍 방식을 통해 병렬 처리해 준다. 분산 병렬 처리에 필요한 작업 스케줄링, 부하 분산, 장애 대책 등을 시스템에서 처리해 주기 때문에 쉽게 data parallel 스타일의 병렬 처리를 가능케 한다.

Hadoop 플랫폼은 Google 플랫폼과 매우 유사하지만 오픈소스 진영에 의해 개발된 오픈 플랫폼이라는 점에서 의미하는 바는 매우 크다. 여러 기업들에 의해 Production 환경에서 안정성을 검증받으므로써 도입 사례가 급격히 증가하고 있고 이것은 Hadoop 개발에 긍정적인 피드백으로 작용하여 개발 속도를 증가시키고 릴리즈 주기를 단축시키고 있다. 또한 외부에서 Hadoop의 부족한 부분을 개발하는 서브 프로젝트들이 수십개씩 생겨 나면서 일종의 Hadoop Ecosystem (그림 8)을 형성하고 있다. 오픈 플랫폼을 중심으로 형성된 이러한 개발자 및 사용자 생태계는 기업들의 폐쇄적인 플랫폼이 따라오기 힘든 큰 장점이다. 국내에서도 한국 Hadoop Community [24]가 Hadoop Ecosystem 활성화를 위해 활동 중에 있다.



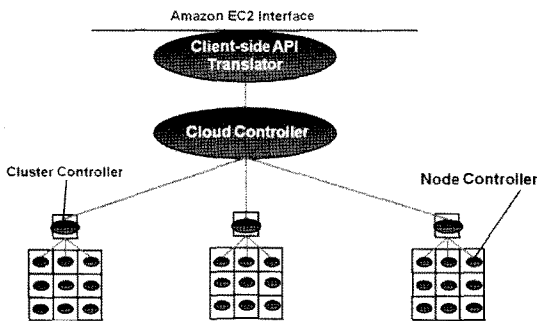
(그림 8) Hadoop Ecosystem

4.2 Eucalyptus: Elastic Utility Computing 플랫폼

Eucalyptus[7]는 캘리포니아 산타바바라 대학 (UCSB)에서 클라우드 컴퓨팅 연구를 위해 만든 오픈 소스 플랫폼이다. 연구 목적을 위해 만든 것이기 때문에 상업적인 플랫폼보다 설치 및 관리가 용이하고 플랫폼의 수정 및 확장이 쉽도록 설계되어 있다. 컴퓨팅 자원들에 대한 단순한 계층 구조와 모듈형 디자인을 통해 확장성을 달성



하였고 Virtual Networking과 Web Services에 의한 느슨한 연결로 기존 인프라 환경에 영향을 주지 않고 설치가 가능하다. 그리고 설치의 편의성을 위해 오픈 소스 클러스터 설치 도구인 Rocks Cluster[25]를 활용한다. 또한 상업적으로 성공적인 Amazon EC2와 인터페이스 호환성을 보장하여 기존 EC2 클라이언트 툴을 사용하여 제어할 수 있다. 다양한 오픈 소스 소프트웨어를 활용하고 있으며 대표적으로 Xen hypervisor와 Axis2, JiBX, Rampart 등의 산업계 표준 Web Services 소프트웨어들을 이용하고 있다.



(그림 9) Eucalyptus 플랫폼 구조

#### 4.2.1 Node Controller(NC)

VM Instance를 호스팅하는 물리적 노드를 관리하기 위한 컴포넌트로 각 노드에 설치되어 runInstance, terminateInstance, describeInstance 등의 VM Instance를 관리하고 제어한다.

#### 4.2.2 Cluster Controller(CC)

노드 클러스터에서 NC들을 관리하는 역할을 하며 일반적으로 클러스터의 헤드 노드에 설치된다. NC들에서 상태 정보를 수집하고 Instance 제어 명령을 NC에게 전달하며 Virtual Networking을 관리한다.

#### 4.2.3 Cloud Controller(CLC)

클라이언트가 플랫폼에 접속하는 entry point이며 클라우드 전체에 대한 설치 및 제어를 담당한다. 사용자 요청 및 관리자 요청에 대한 처리,

VM Instance 스케줄링, Service Level Agreements(SLA) 처리 및 메타데이터 관리 등을 수행한다.

#### 4.2.4 Client-side API Translator

내부의 Eucalyptus 시스템 인터페이스와 몇몇 외부 클라이언트 인터페이스간 변환을 책임진다. 예를 들어 Amazon EC2 SOAP 인터페이스는 Eucalyptus 내부 오브젝트로 변환되어 처리된다.

### 4.3 Enomaly ECP(Elastic Computing Platform) 플랫폼

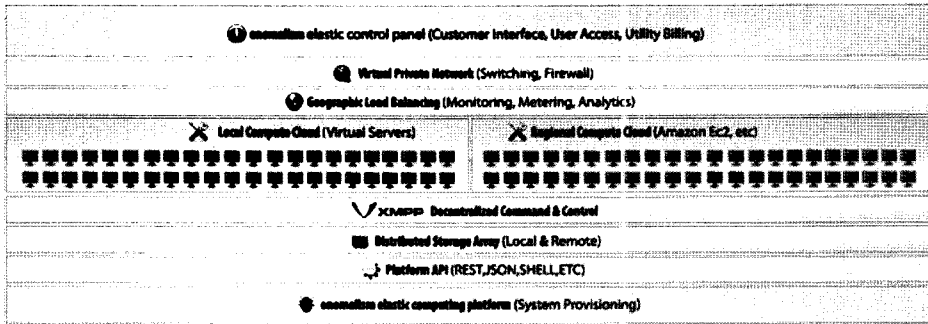
Enomaly ECP[10]는 로컬 및 리모트 컴퓨팅 노드들을 가상 클라우드 인프라스트럭처 환경으로 구성해서 Virtual Application을 실행하고 관리할 수 있도록 하는 오픈 소스 소프트웨어이다. 특히 서버 가상화 관리 소프트웨어가 확장된 형태여서 클라우드 컴퓨팅 서비스보다 클라우드 컴퓨팅 관리 기능에 중점을 두고 있는 것이 특징이다. 이를 위해 웹 기반의 Management Dashboard를 제공하여 VM 배치 플래닝, 자동 VM 스케일링, 부하 분산 등의 기능을 제어할 수 있다. (그림 10)는 Enomaly ECP 플랫폼의 구조를 나타낸다.

#### 4.3.1 서버가상화관리

다양한 hypervisor와 인터페이스할 수 있는 libvirt[26] 오픈 소스 소프트웨어를 사용하여 Xen, KVM, VMware 등의 서버 가상화를 지원하며 Virtual Application Wizard로 가상 어플리케이션을 쉽게 배포할 수 있게 한다.

#### 4.3.2 Hybrid Cloud Computing

갑작스러운 자원 요구 증가에 대처하기 위해 Private Cloud와 Public Cloud를 결합한 Hybrid Cloud Computing 모델을 채택하고 있다. 이를 위해 Virtual Private Cloud(VPC)라는 개념을 도입, 로컬 자원과 리모트 자원을 단일 homogeneous computing 환경으로 묶어준다.



(그림 10) Enomaly ECP 플랫폼 구조

#### 4.4 EU Reservoir Cloud Computing Project

Reservoir[9]는 IT 서비스를 유틸리티로서 효율적이고 안정적으로 배포 및 운영하기 위한 차세대 컴퓨팅 클라우드를 개발하는 프로젝트다. 특히 단일 클라우드 컴퓨팅의 제한된 확장성과 클라우드 컴퓨팅 서비스간 Interoperability의 부재를 핵심 이슈로 지적하고 이를 해결하기 위해 Open Federated Cloud Computing 플랫폼을 연구하고 있다. 또한 기업 비즈니스 요구에 맞추기 위해서 Service Level Agreement(SLA) 관리와 이에 기반한 동적 자원 제어 알고리즘 개발에 초점을 맞추고 있다. (그림 11)은 Reservoir 플랫폼의 구조를 나타낸다.

##### 4.4.1 Service Manifest

서비스 어플리케이션에 관한 모든 사항을 담고 있다. 마스터 이미지(OS, 미들웨어, 어플리케이션, 데이터, 설정 등)와 이를 이용한 VM Instance 생성 규칙, Virtual Networking 컴포넌트 등의 서비스 어플리케이션 구조 정보를 가지고 있으며 이는 Open Virtual Format(OVF) [27]를 확장한 형태로 기술된다. 또한 자원 할당 요구사항을 명시하여 이를 기반으로 플랫폼은 자원을 제어하여 SLA를 만족시킬 수 있도록 한다.

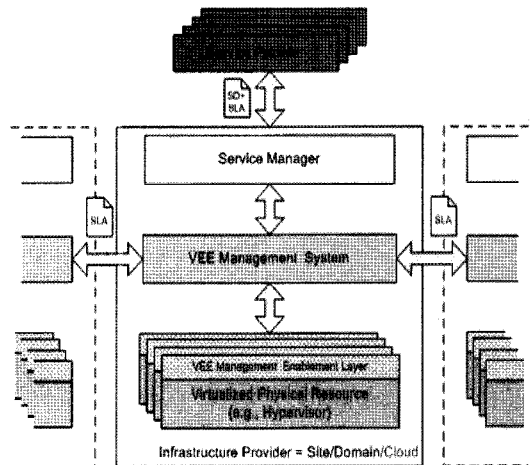
##### 4.4.2 Service Manager

Service Manifest를 받아 비용을 협상하며 지

불처리를 관리하는 등 서비스 제공자와의 커뮤니케이션을 담당한다. Service Manifest의 정보에 따라 서비스 어플리케이션을 배포 및 설치하고 이들을 감시하며 SLA에 맞게 자원 할당량을 조절한다.

##### 4.4.3 VEE Manager

Virtual Execution Environment(VEE)를 Service Manifest에 명시된 조건에 맞는 물리적 노드로 적절히 배치하는 역할을 하며 Remote Cloud와의 Federation을 책임진다. 이를 위해 Remote Cloud에 VEE를 이동시키고 관리 및 모니터링 하는 기능을 구현한다.



(그림 11) Reservoir 플랫폼 구조

## 5. 결론

앞서 살펴 본 것처럼 현재는 오랜 기간 기술적 노하우와 운영 경험을 보유하고 있는 글로벌 IT 기업의 클라우드 컴퓨팅 플랫폼이 기술 성숙도 면에서 높은 경쟁력을 가지고 있다. 이 때문에 클라우드 컴퓨팅으로 사업을 전환하려는 여러 기업이나 새로운 클라우드 컴퓨팅 벤처들이 클라우드 컴퓨팅 시장에서 기술 우위를 점하기는 당분간 쉽지 않을 전망이다. 하지만 Hadoop 사례처럼 기술적으로 뛰어난 오픈 플랫폼은 시장에서 인정받고 많은 사용자 및 개발자를 확보하면서 점점 더 경쟁력 있는 플랫폼으로 성장할 수 있다. 또한 특화된 기술을 보유한 다양한 오픈 플랫폼이 등장하고 있으므로 이들과 유기적인 협력 관계를 유지하면서 그들의 기술을 잘 활용한다면 플랫폼 경쟁력을 확보하는 시간을 단축시킬 수 있을 것이다. 이와 같은 협력관계는 오픈 플랫폼 생태계를 형성하여 기존 선두 클라우드 컴퓨팅 플랫폼과 경쟁할 수 있는 강력한 기술 기반을 제공해 줄 수 있다. 다소 뒤쳐진 국내 클라우드 컴퓨팅 플랫폼 기술을 단기간에 선두 사업자들의 수준까지 발전시키려면 오픈 플랫폼을 최대한 활용하는 전략이 필요하다.

## 참고문헌

- [1] Nicholas Carr, "The Big Switch", W.W. Norton & Co., Jan 2008
- [2] Amazon Web Services, <http://aws.amazon.com/>
- [3] Google App Engine, <http://code.google.com/appengine/>
- [4] Windows Azure, <http://www.microsoft.com/azure/windowsazure.mspx>
- [5] 한재선, "사례 위주로 살펴 본 클라우드 컴퓨팅 플랫폼 기술", 마이크로소프트웨어, Jan 2009
- [6] Hadoop, <http://hadoop.apache.org/core/>
- [7] Daniel Nurmi, Rich Wolski, Chris Grzegorzczak, Graziano Obertelli, Sunil Soman, Lamia Youseff, Dmitrii Zagorodnov, "The Eucalyptus Open-source Cloud-computing System", Cloud Computing and Its Applications, Oct 2008
- [8] OpenNebula, <http://www.opennebula.org/>
- [9] L.M. Vaquero, L. Rodero-Merino, J. Caceres, M. Lindner, "A Break in the Clouds: Towards a Cloud Definition", ACM Computer Communication Reviews, 2009
- [10] Enomaly ECP, <http://www.enomaly.com/>
- [11] Paul Barham, Boris Dragovic, Keir Fraser, Steven H. Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Warfield, "Xen and the Art of Virtualization", ACM Symposium on Operating Systems Principles, Oct 2003
- [12] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google File System", ACM Symposium on Operating Systems Principles, Oct 2003
- [13] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber, "Bigtable: A Distributed Storage System for Structured Data", Symposium on Operating System Design and Implementation, Nov 2006

- [14] memcached, <http://www.danga.com/memcached/>
- [15] Infrastructure as a Service, [http://en.wikipedia.org/wiki/Infrastructure\\_as\\_a\\_Service](http://en.wikipedia.org/wiki/Infrastructure_as_a_Service)
- [16] Platform as a Service, [http://en.wikipedia.org/wiki/Platform\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Platform_as_a_service)
- [17] David Chappell, "Introducing the Azure Services Platform", David Chappell & Associates, Oct 2008
- [18] LINQ, <http://msdn.microsoft.com/en-us/vcsharp/aa904594.aspx>
- [19] Security Assertion Markup Lanaguage, <http://en.wikipedia.org/wiki/SAML>
- [20] NexR, <http://www.nexr.co.kr>
- [21] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Symposium on Operating System Design and Implementation, Dec 2004
- [22] Hadoop Distributed File System, [http://hadoop.apache.org/core/docs/current/hdfs\\_design.html](http://hadoop.apache.org/core/docs/current/hdfs_design.html)
- [23] HBase, <http://hadoop.apache.org/hbase/>
- [24] 한국 Hadoop Community, <http://www.hadoop.or.kr>
- [25] Rocks Cluster, <http://www.rocksclusters.org/>
- [26] Libvirt, <http://libvirt.org/>
- [27] OpenVirtualFormat, [http://en.wikipedia.org/wiki/Open\\_Virtual\\_Machine\\_Format](http://en.wikipedia.org/wiki/Open_Virtual_Machine_Format)

## 저자약력



**한 개 선**

1998년 부산대학교 전자공학과 학사  
 2000년 KAIST 전기 및 전자공학과 석사  
 2005년 KAIST 전자전산학과 박사  
 2005년~2007년 KAIST 전산학과 선임연구원  
 2008년~현재 한국 Hadoop Community 회장  
 2007년~현재 KAIST 정보미디어 경영대학원 겸직교수  
 2007년~현재 NexR 대표이사  
 관심분야 : 클라우드 컴퓨팅, 플랫폼, 대용량 분산 시스템,  
 Hadoop, 데이터마이닝, 데이터웨어하우징, BI,  
 웹 2.0  
 이 메 일: [jshan@nexr.co.kr](mailto:jshan@nexr.co.kr)  
 Blog: <http://www.web2hub.com/>