

# 사례 발표 | 클라우드 컴퓨팅 기반의 SaaS 플랫폼을 위한 해법, 마이크로소프트의 애저 서비스 플랫폼

## 목 차

- 1. 서 론
- 2. 애저 서비스 플랫폼
- 3. Windows Azure
- 4. Azure Services

신현석 · 장현춘  
(한국마이크로소프트)

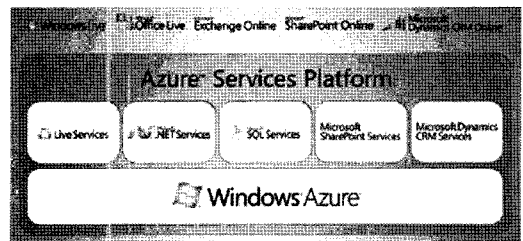
## 1. 서 론

마이크로소프트의 클라우드 컴퓨팅(Cloud Computing) 기술은 “소프트웨어+서비스(Software+Services, 줄여서 S+S)”라는 전략을 바탕으로 하여, (1)대량 가상화 그리드 (Massive Virtualization Grid) 기술, (2)클라우드 클라이언트(Cloud Client) 기술, (3)클라우드 서비스 플랫폼(Cloud Services Platform) 기술, (4)클라우드 서비스로 확장 연동(Scale Out To Cloud Services)하는 기술, 모두 네 가지 서로 다른 기술 시나리오의 총합으로 요약할 수 있다. 본 글에서는 이 가운데 (2),(3),(4)에 해당하는 애저 서비스 플랫폼(Azure Services Platform) 기술에 중점을 두고 여타 클라우드 컴퓨팅 기술과 대비되는 차별성을 논한다.

## 2. 애저 서비스 플랫폼

애저 서비스 플랫폼은 2008년 10월 말경 공식 발표되었다. 애저 서비스 플랫폼은 (그림 1)과 같이 윈도우 애저(Windows Azure)라 명명된 클라우드 운영체제와 다섯 개의 서비스 개발자

블록을 한 데 묶은 통칭이다. 이는 그간 하드웨어 업체를 중심으로 데이터 센터의 고도화된 인프라스트럭처(Infrastructure) 구성에 중점을 두고 클라우드 컴퓨팅의 가치를 해석하던 것과는 달리, 어플리케이션 개발의 편의성과 비용 측면에서 초점을 맞추어 “고성능의 SaaS 플랫폼(High-performance SaaS Platform)”을 실현하는데 클라우드 컴퓨팅을 활용하였다는 점에서 차별성과 의의를 찾을 수 있다. 지금부터는 애저 서비스 플랫폼이 제공하는 서비스 블록들의 기능적 특성을 차례로 상세하게 설명한다.



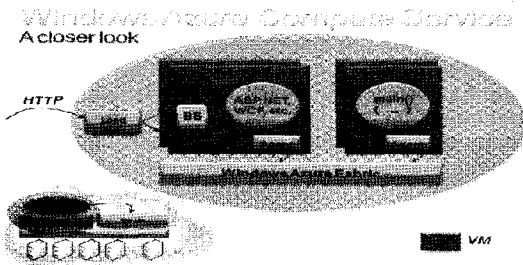
(그림 1) 애저 서비스 플랫폼의 서비스 블록 구성도

## 3. Windows Azure

Windows Azure는 두 가지 중요한 일, 애플리

케이션을 구동(Compute), 데이터를 저장(Storage)하는 역할을 한다. Windows Azure는 마이크로소프트 데이터센터에 존재하는 수많은 서버를 통합 운영하여 강력한 프로세싱 파워를 낼 수 있는데, Fabric이라고 하는 개념을 기반으로 Compute & Storage서비스가 구축 되어 있다.

Compute 서비스는 Windows Server 기반으로 이루어져 있고, 초기 CTP(Community Technology Preview) 버전에서는 .NET Framework 기반으로 구축된 애플리케이션만 구동되지만, Unmanaged 코드 역시 지원할 계획을 가지고 있다. CTP 버전에서는 ASP.NET 애플리케이션과 Windows Communication Foundation (WCF) 서비스 같은 .NET 기반의 소프트웨어를 만들 수 있는데 C#, 기타 다른 .NET 언어를 Visual Studio 2008 같은 개발 도구를 이용할 수 있다.



(그림 2) Web Role, Worker Role Instance로 구성되고, 각 Instance는 독자적인 VM에서 구동됨

### 3.1 애플리케이션 구동(Compute 서비스)

애플리케이션은 보통 여러 개의 Instance를 가지고 있고, 각 Instance는 독자적인 VM 안에서 구동 된다. 이 VM들은 64비트 Windows Server 2008 환경이고, Cloud에서 사용되도록 디자인된 Hypervisor에 의해 서비스 된다. Windows Azure 애플리케이션은 실제로 그 VM을 볼 수 없도록 설계되어 있다. CTP 버전에서는 Web Role, Worker Role Instance를 통해서 .NET 3.5 애플리케이션을 만들도록 구성되어 있다. Web Role Instance는 IIS7 웹서버를 경유하여

HTTP(HTTPs) 요청을 받아 들인다. Web Role 은 ASP.NET, WCF, .NET Framework 기술을 통해 구현하는데 Web Role Instance 간에 요청을 분산시킬 수 있도록 Built-In 로드밸런싱을 지원한다. Worker Role Instance는 외부로부터 요청을 직접 받아들일 수 없게 되어 있기 때문에 외부의 네트워크 커넥션은 허용되지 않고, 웹서버가 VM 내부에 구동 되지 않는다. 입력이 Web Role Instance로 부터 들어와서, Windows Azure Storage 내부의 Queue에 저장된 후 작업 결과 값이 Windows Azure Storage에 저장되거나 외부로 보내진다. Incoming HTTP 요청을 처리하고, 요청이 처리된 후 종료되는 Web Role과 다르게 Worker Role Instance는 배치 작업처럼 무한히 구동될 수 있다.

Windows Azure의 초기 배포 버전은 VM과 물리적 프로세서 코어 간에 1:1 관계를 유지하게 되어 있고 이 원칙을 통해 각 애플리케이션의 성능이 보장된다. 각 Web Role, Worker Role Instance는 자신에게 할당된 프로세서 코어가 있는 것이다. 애플리케이션의 성능을 증가시키기 위해, 소유자는 애플리케이션 설정 파일 안에 운영되는 Instance의 수를 증가시킬 수 있다. 그런 후에 Windows Azure Fabric이 새로운 VM을 생성한 후, 코어에 할당하고 이 애플리케이션에 더 많은 Instance를 구동하도록 한다. Fabric은 또한 Web Role, Worker Role Instance에 오류가 발생했을 때 새로운 Instance를 구동해주는 역할을 한다. Windows Azure의 Web Role Instance는 stateless 형태를 지원한다. 상태 정보를 처리하기 위해서는 Windows Azure Storage를 사용하거나 쿠키를 이용해야 한다. Web Role의 statelessness는 또한 Windows Azure의 Built-In Load Balancer에 의한 특징이다. 요청이 특정 Web Role Instance에게만 보내지는 것이 허용되지 않기 때문이고, 한 사용자가 사용하는 여러 요청들이 동일 Instance로 보내진다는 보장이 없기 때

문이다. Web Role과 Worker Role은 모두 표준 .NET 기술을 사용하여 구현되었다. 그러나, 현재 있는 .NET Framework 애플리케이션을 Windows Azure로 변경 없이 사용하도록 하는 것은 불가능하다. 애플리케이션이 스토리지를 접근하는 방식이 다르기 때문이다. Worker role Instance는 입력을 위해 Windows Azure 스토리지의 큐를 이용한다. 개발의 편의를 위해 제공되는 Windows Azure 소프트웨어 개발 Kit은 개발자의 데스크탑에서 구동할 수 있는 Windows Azure Environment를 포함하고 있다. 이 Windows Azure-in-a-box는 Windows Azure Storage, Windows Azure agent, 그리고 Cloud에서 애플리케이션이 구동되기 위해 필요한 것들 다 포함되어 있다. 개발자는 이 환경에서 애플리케이션 디버깅을 할 수 있고, 테스트가 완료됐을 때 실제 Windows Azure Cloud로 배포할 수 있다. Windows Azure fabric은 애플리케이션 실패를 감지하고, 결과를 보낼 수 있다. 또한, Windows Azure platform은 애플리케이션의 자원 소비량, 프로세서 시간, incoming & outcome 대역폭, 스토리지 등에 대한 상세한 정보를 제공한다.

### 3.2 데이터를 저장(Storage 서비스)

애플리케이션은 데이터와 다양한 방식으로 동작한다(Blob, Queues, Tables).

Blobs은 가장 간단한 방식이다. Windows Azure Storage의 계정은 Container, 50기가 바이트의 크기까지 저장 가능한 Blob, 더 작은 단위인 Block으로 구성되어 있다. 만약 오류가 발생하면, 전체 Blob을 다시 보내는 것이 아니고 가장 최근의 Block을 다시 보내준다. Blobs은 JPEG 사진이 어디서 찍었고, MP3파일의 작곡가가 누구인지 등에 메타데이터 정보를 가질 수 있다.

또 다른 유형인 Tables의 경우 Entity의 계층

구조를 저장할 수 있다. 정해진 스키마를 가지고 있지 않고, 프로퍼티를 통해 다양한 유형의 데이터 타입, 문자열, Int, Bool, DateTime등을 가질 수 있다. 테이블은 여러 서버에 파티션으로 나뉘어 저장되기 때문에 수십억 개의 Entity를 저장할 수 있다.

Queues는 Web Role Instance가 Worker Role Instance와 통신할 수 있는 방법을 제공한다. 사용자가 Windows Azure Web Role을 통해 구현된 웹 페이지에서 컴퓨팅을 많이 필요로 하는 작업을 요청할 경우 Web Role Instance가 어떤 작업이 이루어져야 하는지를 큐에 메시지로 넣고 Worker Role Instance가 이 큐 값을 읽어서 정해진 작업을 수행하고, 다른 큐를 통해 해당 결과 값을 돌려줄 수 있다. Blobs, Tables, Queues에 관계 없이 Windows Azure storage에 저장된 데이터는 Fault Tolerance를 위해 3번 복제 된다.

Windows Azure storage는 REST 방식을 사용한다. 모든 자원은 URI로 이름이 부여되고, 표준 HTTP 방식으로 처리된다. .NET 클라이언트는 ADO.NET 서비스나 LINQ를 사용할 수 있지만, Java 애플리케이션으로는 표준 REST 방식을 사용할 수 있다. Blob이 HTTP GET을 이용해, URI를 호출하는 방법은 아래와 같다.

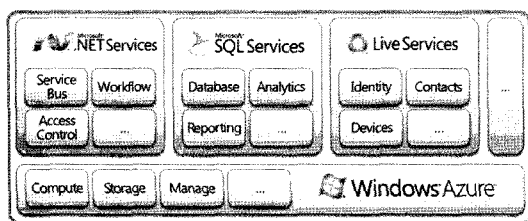
```
http://<StorageAccount>.blob.core.windows.net
/<Container>/<Blobname>
```

<Storage Account>는 새로운 Storage Account가 생성될 때 지정되는 식별자이다. <Container>, <Blobname>은 Container의 이름,이 요청이 접근할 Blob의 이름이다.

Tables, Queues 역시 유사한 방식을 사용한다. Windows Azure 플랫폼은 Compute와 Storage 자원을 개별적으로 비용을 책정하기 때문에 On-Premise 애플리케이션이 RESTful 방식으로 Windows Azure Storage를 사용할 수 있다.

## 4. Azure Services

Windows Azure와 함께 Azure Services Platform을 구성하는 다른 한 축은 개발자가 사용할 수 있는 클라우드 스케일의 빌딩 블록 서비스 모음이라 할 수 있는 Azure Services이다. 아래 (그림 3)에서 보듯이 Windows Azure가 제공하는 클라우드 OS로서의 기본 서비스와는 차별되는 고급 기능을 제공하며, 여기에는 .NET Services, SQL Services, Live Services 등이 현재 제공되고 있으며 향후 Sharepoint Services, Dynamic CRM Services과 같이 현재 마이크로소프트가 SaaS 서비스로 제공하고 있는 기능과 연동할 수 있는 기능을 제공할 예정이다.



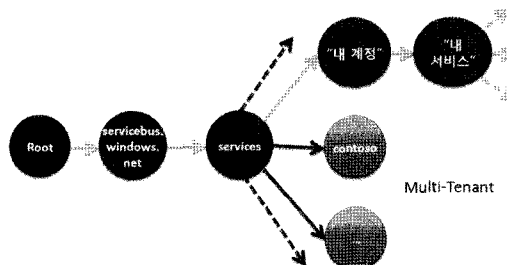
(그림 3) Azure Services 구성 요소

### 4.1 .NET Services

.NET Services는 일반적인 닷넷 애플리케이션 개발에서 .NET Framework이 제공하던 기능을 클라우드 스케일로 확장한 개념이다. 즉, 클라우드 상의 애플리케이션이 서비스를 노출함에 있어서 접근하는 사용자를 어떻게 인증할 것이며, 방화벽으로 막혀 있는 서비스들 사이의 연동은 어떻게 가능하게 할 것이며, 클라우드 상에서 워크플로우를 실행시켜 나의 애플리케이션 상의 워크플로우와 어떻게 연동할 것인가 등에 대한 입증된 빌딩 블록 서비스를 제공하는 것이다.

SOA 기반의 인프라를 갖추기 위해 기업들은 ESB (Enterprise Service Bus)를 도입하여 표준 웹 서비스 기반으로 기업내 다양한 애플리케이션 서비스 연동을 지원하는데, ESB가 클라우드

스케일로 확장되어 기업간 애플리케이션간 서비스 연동을 제공하는 것이 ISB (Internet Service Bus)이며, ISB의 마이크로소프트 버전 구현체가 바로 .NET Service Bus인 것이다.



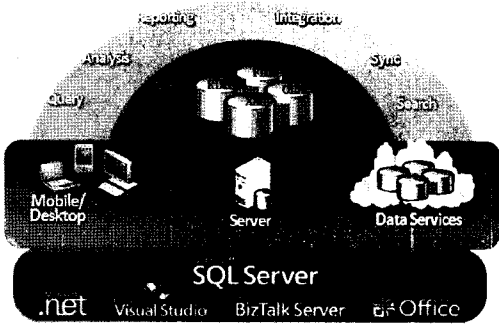
(그림 4) .NET Service Bus 원리

(그림 4)에서 보듯이 .NET Service Bus는 서비스 제공자는 마이크로소프트 데이터 센터에 개설한 자신의 계정에 서비스를 노출하고, 서비스 사용자는 http 프로토콜을 통해 일반 웹 서비스를 호출하는 방식으로 접근할 수 있는 방법을 제공한다. Access Control Services는 다양한 인증 체계를 가진 서비스들 사이에 SAML과 같은 합의된 Federation 표준 기반의 토큰 서비스를 통해 연동할 수 있는 방법을 제공한다.

### 4.2 SQL Services

SQL Services는 마이크로소프트의 SQL Server가 제공하는 각종 Relational한 기능들, 즉 리포팅, 분석, 데이터 동기화, Relational 쿼리 등을 클라우드 서비스로 제공하는 것으로 현재 SQL Data Services (SDS)를 통해 일부 기능을 제공하고 있다. SQL Server 기반으로 구현되어 있는 SDS는 현재 버전의 CTP에서는 일반적인 Relational한 데이터베이스 접근 방식을 제공하지는 않고 REST 및 SOAP 기반의 웹 서비스 인터페이스를 제공하고 있다. 하지만, 향후 SDS의 방향은 Transact-SQL을 지원하여 현재 개발자가 사용하고 있는 Relational한 쿼리를 그대로 활

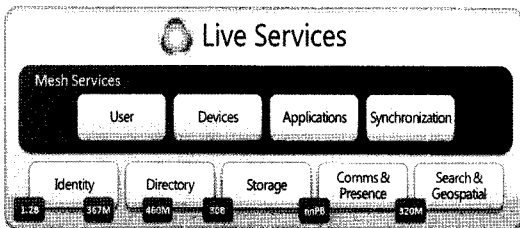
용할 수 있는 클라우드 상의 Relational Database 로 역할을 할 예정이며, 이를 통해 앞서 기술한 Windows Azure내의 Storage 서비스와는 차별화된 서비스를 제공하게 된다.



(그림 5) SQL Data Platform 확장으로서의 SQL Services

### 4.3 Live Services

Live Services는 Windows Live Hotmail, Windows Live Messenger 등 기존에 마이크로소프트가 제공하던 각종 Windows Live 애플리케이션과의 원활한 연동을 제공하며 개발자는 Live Services가 제공하는 Live Framework을 통해 일관된 방식으로 마이크로소프트의 각종 Live 애플리케이션이 관리하는 데이터에 접근할 수 있다. 즉, 아래 (그림 6)에서 보듯이 Hotmail이나 Messenger에서 관리되고 제공되는 엄청난 양의 개인 정보, 지인 연락처 정보, 지인의 로그인 여부, 검색 서비스 등에 대해 접근할 수 있으며 이를 기반으로 애플리케이션을 개발할 수 있고 또한 내 애플리케이션이 사용하는 데이터를 다양한 디바이스간에 동기화 기능도 제공할 수 있다.



(그림 6) Live Services가 제공하는 기능들

현재 Live Services를 이용하여 마이크로소프트가 제공하는 서비스로 Live Mesh (<http://www.mesh.com>) 가 있으며, 다양한 디바이스간 데이터 및 애플리케이션 동기화 기능을 제공하고 있다.

### 참고문헌

- [1] David Chappell, "Introducing The Azure Services Platform," [http://download.microsoft.com/download/e/4/3/e43bb484-3b52-4fa8-a9f9-ec60a32954bc/Azure\\_Services\\_Platform.pdf](http://download.microsoft.com/download/e/4/3/e43bb484-3b52-4fa8-a9f9-ec60a32954bc/Azure_Services_Platform.pdf), 2008.10
- [2] Microsoft, Azure Services Platform, <http://www.microsoft.com/azure/windowsazure.msp>
- [3] Microsoft, Azure Services Platform, <http://msdn.microsoft.com/en-us/azure/default.aspx>
- [4] Microsoft, MSDN, <http://msdn.microsoft.com/en-us/azure/cc994380.aspx>
- [5] Microsoft, Windows Azure Blog, <http://blogs.msdn.com/windowsazure>
- [6] 신현석, "WOA, 전문가기고," IT Today, 2008.12.21
- [7] 신현석, "MS 클라우드 전략의 '코어', 원도우 애저," 마이크로소프트웨어, 2009.01

## 저자약력



**장연순**

1997년 2월 서울대학교 자원공학과(학사)  
1996년 12월 LG CNS 객체기술팀(사원)  
2000년 10월 Summus Technology 창업(기술이사)  
2003년 5월 한국썬마이크로시스템즈(컨설턴트)  
2005년 7월~현재 한국마이크로소프트(에반젤리스트)  
관심분야 : 분산 관련 분야(언어, 미들웨어)  
이 메 일 :



**신연석**

1994년 2월 서울시립대학교 전산통계학과 (이학사)  
2008년 10월~Helsinki School of Economics MBA  
1994년 3월~2006년 1월 삼성SDS, 삼성네트웍스  
(시스템 프로그래머, 엔지니어)  
2006년 2월~2007년 6월 마이크로소프트 Asia Pacific  
Web Platform Architect Lead  
2007년 7월~한국마이크로소프트 개발자 및 플랫폼 사업총괄  
Infrastructure Architect Evangelist (부장)  
관심분야 : 클라우드 컴퓨팅, SaaS 및 WOA(Web Oriented  
Architecture), 그린 IT, IDC(Internet Data Center)  
관련 기술  
이 메 일 :