HPC
In the Cloud

Mastering Windows Azure Application Development

Dmitry Yakovlev, Senior Vice President, DataArt

Cloud Landscape

Before mastering cloud application development is discussed, it's important to go over the existing cloud landscape. There are four major cloud computing platforms available on the market today: Amazon EC2, Windows Azure, Google App Engine and Force.com. Each of these platforms is a more or less successful attempt to commercialize internally-crafted virtualization technology. While Amazon and Microsoft succeeded in building general-purpose cloud environment, Google and SalesForce remain niche players due rudiments of internally-grown technology and use of programming languages like Python and APEX.

In the years to come, cloud computing will take a substantial piece of the market from traditional deployment models. This implies growing demand for applications that can operate in a cloud environment, and for software engineers skilled in cloud computing technologies. Since commercial software development is driven by enterprises which prefer mainstream technologies, Amazon EC2 and Windows Azure are likely to be the two platforms of choice for software developers. Considering the fact that both platforms can host applications written in different programming languages, one should denote Java and .NET as primary development platforms for Amazon EC2 and Windows Azure respectively.

In this article we will focus on mastering application development for Windows Azure which is a valuable investment for a .NET programmer.

Required Knowledge

Let's discuss specific skills required for Windows Azure application development, things to start with and areas to be studied.

Mandatory Skills

A programmer looking to dive into Windows Azure application development should have a working knowledge of Microsoft .NET technologies. Particular skills are:

- .NET Framework 4.0
- ADO.NET Data Services
- LINQ
- Windows Communication Foundation (WCF)
- ASP.NET MVC 3.0
- Multi-threading

Developers should be familiar with RDMS concepts and MS SQL 2008. Additionally, a solid understanding of HTTP protocol and REST concept is very desirable as it helps to assess the implications of network topology (load balancers, proxy servers, CDNs) on RESTful web services. Knowledge of Service Oriented Architecture (SOA) design principles is essential as cloud applications strongly rely on services.

Cloud Concepts

1. The first logical step is to become familiar with cloud-related concepts and to adopt the principles of cloud application development. There is a lot of information on the Web about cloud computing. From a software developer's perspective, cloud can be treated as a way to get on-demand access to two types of scalable resources: compute (CPU) and storage which are available via services provided by the cloud platform.

2. The second step is to learn how Windows Azure hosting environment works in detail. There is a good presentation at Channel9 describing platform infrastructure and application lifecycle. As result of this step, a developer should recognize and adopt the following ideas:
    - **Cloud application runs in a bare Windows 2008 operating system**
      Don't assume that Windows Azure hosting environment has any preinstalled software; it's a bare operating system. Any functionality, usually supported by preinstalled software, should be instead implemented within the application hosted on Windows Azure.
    - **The application instance can be recycled by the platform at any point of time**
      Everything stored on a local disk drive memory will be deleted once the instance is recycled. To preserve the data and make it available to other instances, use Windows Azure Storage services.
    - **Cloud application runs in a concurrent environment**
      Services provided by Windows Azure platform are designed to operate in concurrent environments with the use of "try and correct" pattern. The application should follow this pattern and properly handle cases in which access to a service is declined by repeating the operation later. Another aspect to keep in mind in that a web application under Windows Azure always runs behind load balancer.

Getting Started with Windows Azure Development

I would recommend beginning by reading a book by Tejawsi Redkar "Windows Azure Platform" (second edition) which gives a good introduction to Windows Azure for beginners.

Setting Up the Development Environment

First, one has to setup the development environment. Windows Azure development environment requires Windows Vista SP2, Windows 7 or Windows 2008 operating system. The following software should be installed:

- Visual Studio 2010 Professional or above
- Windows Azure Tools & SDK (latest version is available at http://www.microsoft.com/windowsazure/sdk/)
- Windows Azure Storage Explorer – a convenient GUI tool to explore Azure storage (available at http://azurestorageexplorer.codeplex.com/)
- Code samples (http://code.msdn.microsoft.com/windowsazure)

At this stage, one should be able to open Visual Studio and create a blank Windows Azure solution. As an exercise, I would suggest implementing a simple online photo storage application.

Web, Worker and VM Roles

Learn three types of application roles supported by Windows Azure:

- Web role
- Worker role
- Virtual Machine (VM) role

The first two roles are analogs of traditional web application and Windows services. VM role is somewhat special and shouldn't be used unless a customized version of a guest operating system is required. Using VM roles puts the burden of OS support on the system administrator.

Pay attention when logging in to the Windows Azure application. Remember that debugging a cloud application is rather hard, if at all possible. Therefore, the application should emit and store enough debugging information to allow for discovering and tracing problems in the code.

Storage Services

There are three types of storage supported by Windows Azure platform:

- Blob storage
- Table storage
- Queue storage

The services are exposed via REST API and available outside of Windows Azure hosting environment as well, so one can create an application for a mobile device which interacts directly with the storage. Windows Azure SDK comes with a managed library providing access to storage services via an object model.

There are several important things to know when working with Azure storage:

- Storage objects are addressed by URL, so certain restrictions are applied to the object name.
- There are three types of blobs: single blob, block blob and page blob. Each of them has a minimum and a maximum size.
- Table storage is not a relational database. There are no relationships, indexes and constraints. It's more like an Excel spreadsheet highly scalable in the number of rows.
- A table always includes two properties (PartitionKey and RowKey) forming a primary key, the total length of the key can't exceed 1024 symbols. Only 256 symbols of the primary key can be used to address the record.

- Table always includes Timestamp field used to resolve conflicts
- Table has limits of 1M per entity (row) and 64K per property (field)
- Sorting is not supported by Table storage, so it's always done on the client side.
- Maximum number of records returned by a query against Table is limited to 1,000 entities per request. A continuation token should be used to retrieve subsequent data.

Windows Azure storage services should be studied in detail, with particular attention paid to addressing a scheme, size limits, and operation restrictions. This knowledge will prevent you from making wrong decisions in the design of your cloud application.

Among other things worth taking a look at, I would recommend a comprehensive study of Azure storage performance http://azurescope.cloudapp.net/BenchmarkTestCases/ which gives a good idea of storage throughput under different scenarios.

Azure SQL

Azure SQL is a cloud version of regular MS SQL database. It looks like a complete replacement of regular MS SQL database with minor restrictions on T-SQL syntax. However, the fundamental restriction of Azure SQL is size limit of 50Gb per database, so it's not entirely scalable. Recognizing this fact is important for application architecture. You should store in Azure SQL *only* data which shouldn't grow substantially, e.g., a list of user accounts. Quite often developers try to employ Azure SQL to store things like pictures, documents, logs, etc., so the storage space gets exhausted quite soon and the system fails.

Azure SQL databases are available outside of Windows Azure hosting environment. It is possible to setup access restrictions based on an IP address.

AppFabric

Windows Azure AppFabric is a set of middleware services designed to facilitate development of enterprise applications on top of Windows Azure. Currently, the SDK and services are available as a CTP release. Learning AppFabric SDK is not required to develop Windows Azure applications, however one service highly demanded by developers to take note of: Cache service which provides fast access to in-memory data storage. Here I refer to the fact that Windows Azure doesn't provide a way to store web session data, so developers have to implement a custom version session provider relying on Windows Azure storage. The new Cache service addresses this issue. There is a good up-to-date reading about AppFabric by Alan Smith available for free at http://www.cloudcasts.net/devguide/

Deployment

You have arrived at the stage when your first Windows Azure application is implemented and tested in the development environment. To deploy the application to the Windows Azure environment, you need to sign up for the service. There is a free trial available for 90 days (valid credit card is required).

Once the account is set, you can create a storage account and a hosted service. Each hosted service supports two environments: staging and production. Deploying the application is quite simple and requires uploading a package and a configuration file. The application in the staging environment is available under a private URL for testing purposes. Once the staging environment is tested, it can be switched to production in a single click. Now you have your Windows Azure application running.