ISSN:0975-9646

Windows Azure Platform: an Era for Cloud Computing

Madhurima[1], Vandana[2], Madhulika[1]

[1]*School of Engineering & Technology, Apeejay Stya University Sohna, Gurgaon-122103, India*
[2]*Computer Science Department, Skyline Institute of Engineering & Technology Greater Noida, India*

***Abstract*— Windows Azure platform is the Microsoft implementation of cloud computing. This paper covers detailed introduction to Windows Azure Platform. Windows Azure provides resources and services for consumers. The next part describes the five main components of Windows Azure: Hardware is abstracted and exposed as compute resources. Physical storage is abstracted as storage resources and exposed through very well defined interfaces. A common Windows Fabric abstracts the hardware and the software and exposes virtual compute and storage resources. Windows Azure also introduces some new storage options over and above the usual file system. Windows Azure programming model helps developers create applications that are easier to administer, more available, and more scalable than those built in the traditional Windows Server environment.**

***Keywords*— Windows Azure, Cloud, Components, Storage Model, Programming Model.**

I. INTRODUCTION

Windows Azure is a foundation for running applications and storing data in the cloud [2]. Rather than providing software that Microsoft customers can install and run themselves on their own computers, Windows Azure today is a service: Customers use it to run applications and store data on Internet-accessible machines owned by Microsoft. Those applications might provide services to businesses, to consumers, or both [1]. Microsoft's Windows Azure Platform is an internet-scale cloud platform. Azure's flexible and interoperable platform can be used to build new applications to run from the cloud or enhance existing applications with cloud-based capabilities [3]. Windows Azure is a cloud services operating system that serves as the development, service hosting and service management environment for the Windows Azure Platform. For example, physical hardware resources are abstracted away and exposed as compute resources ready to be consumed by cloud applications. Physical storage is abstracted with storage resources and exposed through well-defined storage interfaces. A common

Windows fabric abstracts the physical hardware and software platform and exposes virtualized compute and storage resources. In addition, each instance of the application is monitored for availability and scalability, and automatically managed. Windows Azure runs on machines in Microsoft data centers. The goal of Windows Azure is to provide developers with an on-demand compute and storage platform to host, scale, and manage internet or cloud applications [5]. The platform itself provides and maintains its own copy of Windows. Developers focus solely on creating applications that run on Windows Azure. Windows Azure supports a consistent development experience through its integration with Visual Studio. Windows Azure is an open platform that supports both Microsoft and non-Microsoft languages and environments. Windows Azure welcomes third party tools and languages such as Eclipse, Ruby, PHP, and Python [3].

II. WINDOWS AZURE COMPONENTS

Windows Azure has five main parts: Compute, Storage, the Fabric Controller, the CDN, and Connect.

Compute: Windows Azure compute can run many different kinds of applications. Whatever an application does, however, it must be implemented as one or more roles. Windows Azure then typically runs multiple instances of each role, using built-in load balancing to spread requests across them [1]. This is made up of two kinds of "Roles". The first is a "Web Role", which basically means ASP.NET. A Web Role is the front end code, screens and so on that we expose to our users. The other Role is a "Worker Role". This is basically like the Windows Services or DLL's we use in typical .NET programming. Worker Roles are the programs that don't have a front end to the user [2].

Storage: The second component in Windows Azure is the. We have three types here – Blobs, which are like files, Tables, which are key-value-pair type storage, and Queues, which let Web Roles and Worker Roles, communicate to each other [2].

Fabric Controller: The third component in Windows Azure is the Fabric controller or Application Fabric. This component handles authentication and transport – not only between Windows Azure applications, but even from servers. We can say that we could have that large SQL Server or Oracle system and expose that to an Azure application, and we wouldn't have to allow the users of the application into our network [2].

Content Delivery Network: The CDN stores copies of a blob at sites closer to the clients that use it. The Windows Azure CDN actually has many more global caching locations than it shows, but the concept is correct. The first time a particular blob is accessed by a user, the CDN stores a copy of that blob at a location that's geographically close to that user. The next time this blob is accessed, its contents will be delivered from the cache rather than from the more remote original. [1]

For example, suppose Windows Azure is used to provide videos of a day's sporting events to a far-flung audience. The first user who accesses a particular video won't get the benefit of the CDN, since that blob isn't yet cached in a closer location. All other users in the same geography will see better performance, since using the cached copy lets the video load more quickly [1].

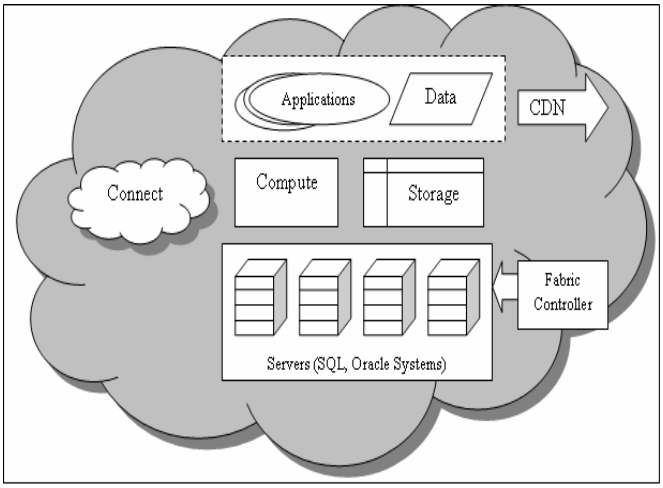

Figure 1: Windows Azure has five main components: Compute, Storage, Fabric controller, CDN and Connect.

Connect: Running applications in the Microsoft cloud is useful. But connecting to the on-premises environments (the applications and data we use inside our organizations) with Windows Azure is important. Windows Azure Connect is designed to help do this. By providing IP-level connectivity between a Windows Azure application and machines running outside the Microsoft cloud, it can make this combination easier to use.

III. Azure Cloud Storage Model

Windows Azure Storage "provides persistent, redundant storage in the cloud". Microsoft's goal is to create storage that is durable and secure, scalable and efficient. Windows Azure Storage allows us to store data for any length of time and to store any amount of data [4].

We can store our data in Windows Azure in four different ways:

- Blobs
- Tables
- Queues
- SQL Azure databases

Blobs: These are used for binary data and are like files on our hard drive. There are two types – block and page. Block blobs are used for streaming, like when we want to start watching a movie before it even completes the download. We can store files up to 200GB at a pop. And they parallelize well. Page blobs are used when we need a LOT of storage (up to a terabyte) and access a "page" directly, with an address [3].

Tables: To allow applications to work with data in a more fine-grained way, Windows Azure storage provides tables. These aren't relational tables. A table has no defined schema; instead, properties can have various types, such as int, string, Bool, or DateTime. A single table can be quite large, with billions of entities holding terabytes of data, and Windows Azure storage can partition it across many servers if necessary to improve performance [6].

Queues: This storage is used to transfer messages between blocks of code. If we think of the stateless-programming web-world, we need a way to tell one application something that isn't event-based. Since it's a queue, it helps us with something called "idempotency", which means that a single message on the queue will get processed once, and only once [3].

SQL Azure Databases: We can use SQL Azure Databases, when we need relational storage and leverage Transact-SQL code we already have [3]. SQL Azure provides a large subset of SQL Server's functionality, including reporting, as a managed cloud service. Applications can create databases, run SQL queries and there's no need to administer the database system or the hardware it runs on. A SQL Azure database can be accessed using the Tabular Data Stream (TDS) protocol, just as in the on-premises version of SQL Server. This lets a Windows Azure application access relational data using familiar mechanisms like Entity Framework and ADO.NET [6].

IV. Windows Azure Programming Model

The Windows Azure programming model is based on Windows, and the bulk of a Windows developer's skills are applicable to this new environment. [2]. The Windows Azure programming model imposes three rules on applications [4]:

Rule 1: Built from one or more roles: Web roles, Worker roles and VM roles.
Rule 2: Runs multiple instances of each role.
Rule 3: Behaves correctly when any role instance fails.

The Windows Azure programming model helps improve application availability. It provides protection against hardware failures. Because every application is made up of multiple instances of each role, hardware failures—a disk crash, a network fault, or the death of a server machine won't take down the application. Along with hardware failures, the fabric controller can also detect failures caused by software. If the code in an instance crashes or the VM in which it's running goes down, the fabric controller will start either just the code or, if necessary, a new VM for that role. An application built using the Windows Azure programming model can be updated while it's running; there is no need to take it down. To allow this, different instances for each of an application's roles are placed in different update domains. When a new version of the application needs to be deployed, the fabric controller can shut down the instances in just one update domain, update the code for these, then create new instances from that new code. Once those instances are running, it can do the same thing to instances in the next update domain, and so on. While users might see different versions of the application during this process, depending on which instance they happen to interact with, the application as a whole remains continuously available. The model also has the ability to update Windows and other supporting software with no application downtime. The fabric controller assumes that every Windows Azure application follows the three rules listed earlier, and so it knows that it can shut down some of an application's instances whenever it likes, update the underlying system software, then start new instances. By doing this in chunks, never shutting down all of a role's instances at the same time, Windows and other software can be updated beneath a continuously running application [4].

V. CONCLUSION

Windows Azure provides a Windows-based computing and storage environment in the cloud. Running applications and storing data in the cloud is the right choice for many situations [7]. The various parts of Windows Azure work together to make this possible. Windows Azure Tables provide scalable, available, and durable structured storage in the form of tables. The tables contain entities, and the entities contain properties. The tables are scalable to billions of entities and terabytes of data, and may be partitioned across thousands of servers. Windows Azure Blobs provide a simple interface for storing named files along with metadata for a file. Windows Azure Queues provide reliable storage and delivery of messages for an application and are the most common basis for Web and Worker role communication. The Windows Azure programming model can be useful for anybody who wants to create easier to administer, more available, and more scalable applications.

REFERENCES

[1] David Chappell, A white paper on "Introducing Windows Azure", Oct 2010.

[2] Dong Xu, "Cloud Computing: an Emerging Technology", International Conference On Computer Design And Applications (ICCDA 2010), Volume-1, Pgs (100-104).

[3] Windows Azure Platform: Overview.http://www.azurepilot.com.

[4] David Chappell, A white paper on "The Windows Azure Programming Model", Oct 2010.

[5] Wei Lu, Jared Jackson, and Roger Barga, "AzureBlast: A Case Study of Developing Science Applications on the Cloud", Proceedings of the 1st Workshop on Scientific Cloud Computing (Science Cloud 2010), Association for Computing Machinery, Inc., 21 June 2010

[6] David Chappell, A white paper on "Introducing The Azure Services Platform: An Early Look At Windows Azure, .Net Services, Sql Services, And Live Services", Oct 2008.

[7] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg , Ivona Brandic "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing " Journal of Elsevier- Future Generation Computer Systems (2009).