DEVELOPING APPLICATION FOR CLOUD – A PROGRAMMER'S PERSPECTIVE

Rajeev BV[1], Vinod Baliga[2] and Seshubabu Tolety[3]

[1]Microsoft Competencies, TEC, Siemens Technology Services Bangalore, India
bv.rajeev@siemens.com
[2]Microsoft Competencies, TEC, Siemens Technology Services Bangalore, India
vinod.baliga@siemens.com
[3]Mobile Computing Team, TEC, Siemens Technology Services Bangalore, India
seshubabu.tolety@siemens.com

ABSTRACT

There are many challenges that the developers will come across while developing or migrating applications to cloud. This paper intends to discuss various points that the developers need to be aware of during the development or migration of the application to the cloud in terms of various parameters like security, manageability, optimal storage transactions, programmer productivity, debugging and profiling, etc. The paper provides insights into how to overcome these challenges when developing / migrating the on-premise application on to cloud and the difference in programming when targeting the on-premise data center and cloud. The primary focus area for cloud in this paper would be on Microsoft Windows Azure, Google App Engine and Amazon cloud.

KEYWORDS

Cloud Computing, Cloud Security, Application Scalability on Cloud, Cloud Data Storage, Legacy Applications on Cloud

1. INTRODUCTION

Developing a new web application targeting the cloud or migrating existing web application to the cloud involves certain changes in the programming model. The developer of the application needs to carefully understand some of the other aspects of programming that could be different from legacy on-premise application deployment. The following sections of the paper discuss some of the important technical aspects that the programmer needs to be aware of, while building applications for the cloud.

2. SECURITY

As far as the physical security of a cloud provider's warehouses and hardware are concerned, most of the providers boast of various security process certifications and third party attestations. But developers need not be aware of these since they are controlled directly by the provider.

Confidentiality, Integrity, Identity and Availability are the most important features that every cloud service provider promises to provide. Confidentiality is mainly provided through various account level security techniques such as Identity Management and Access Management. Most

Sundarapandian et al. (Eds) : ITCS, SIP, CS & IT 09,
pp. 13–21, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3102

cloud providers allow access to cloud accounts through encrypted keys and secure certificates which make the cloud service accounts inherently secure. Amazon Web Services for example provides these features through AWS Identity and Access Management and AWS Multi-Factor Authentication. Each request to the storage account requires authentication via encrypted keys which ensures that the data cannot be illegitimately accessed by unintended users.

All transactions that take place between an application and the corresponding storage account happen via secure http. If the data transmission has to be secured using cryptography with authorized key system then it will have to be done by specific applications.

In terms of network security the cloud service providers offer significant protection against traditional network security issues. Distributed Denial of Service Attacks, Man in the Middle Attacks, IP Spoofing and Port Scanning are minimized through various proven techniques employed by the cloud service providers.

Microsoft's Windows Azure platform provides confidentiality through an array of features such as Service Management API authentication, Least Privilege Customer Software which ensures that every application deployed on cloud run with bare minimum privileges by default hence reducing the risk of privilege exploitation by any malicious software attack. Also, every communication that happens between Windows Azure internal components is protected with SSL. Access to Windows Azure Storage services is also secured by means of access control mechanisms.

In a cloud environment it's never guaranteed that a particular application is the only one running on a particular piece of hardware. Since all applications run in a virtualized environment, chances are that multiple virtual hosts will be running on the same physical hardware. But even in this sort of scenario, application developers need not worry about applications intruding into each other's data since all the applications are isolated from each other by design. Microsoft provides this sort of isolation via technologies such as Hypervisor, Packet Filtering and VLAN isolation. AWS provides similar protection with the use of virtualization and firewall solutions.

Microsoft provides users with options to encrypt the data in storage and in transit. While the permanently stored data can be encrypted by using proven techniques that are provided by .NET Cryptographic Service Providers, the data in transit can be protected with the use of SSL. Both Amazon Web Services and Microsoft Windows Azure platforms provide security to their blob storage services both at container and blob level. There are also options provided where the access to each blob can be logged. Similar sort of security options are available for structural data storage and queue storage services provided by various service providers.

Ultimately most applications need to have their own security mechanism, so that only authorized users can make use of the services provided by them. This is traditionally achieved using techniques such as forms authentication or windows authentication. Similar techniques can be employed in cloud environment as well. If the application wants to leverage proven security mechanisms such as Active Directory Services, then the cloud services provide application developers with various options. Windows Azure applications can make use of Active Directory Services through Windows Azure Active Directory services to enable security features such as single sign-on. With Amazon Web Services developers would have to come up with workarounds to make use of Active Directory Services. Windows Azure also provides Access Control Services with which application developers can provide identity and access control to their web applications while integrating with standards based identity providers such as Live ID, Google and Facebook.

Business applications often require industry specific regulatory compliance. AWS is currently PCI DSS 2.0 Level 1 compliant. Microsoft claims to be currently working on getting this compliance for Windows Azure. None of the major cloud service providers are currently HIPAA compliant although guides are available to make use of cloud storage data protection features as a part of an overall strategy to achieve HIPAA compliance.

High availability of an application is something that any organization strives to achieve. But it is also something that is very hard to achieve because it requires investing on highly specialized tools, lots of hardware and specially trained people. But with cloud achieving high availability could be as simple as changing a configuration setting using the management portals to increase the number of application instances. Data storage is also highly replicated so that multiple copies of data are available at any given point in time. For example SQL Azure provides high availability automatically which is quite complex to achieve on premise.

3. DATA STORAGE

When it comes to storing application data, traditionally developers would make use of server storage or network storage to store large files, Microsoft Message Queuing (MSMQ) or other Enterprise Messaging Service (EMS) such as Tibco for queuing services and Relational Database Management Systems like SQL, Oracle or MySQL for storing structured and relational data. Most of the cloud data storage providers provide alternatives to these services. Large files could be stored using Azure Blob Storage or Amazon S3, de-coupled communication between two applications can be achieved using Amazon Simple Queues and relational data can be stored using Amazon RDS or Google Cloud SQL. Table-1 below shows the list of cloud storage services provided by Amazon, Google and Microsoft.

Table-1: Cloud storage services provided by major cloud service providers.

Storage Feature	**Windows Azure**	**Amazon Web Services**	**Google App Engine**
File Storage	BLOB Storage Service	Simple Storage Service	Blobstore
Queuing Service	Queue Service	Simple Queue Service	Task Queue
Structured Data Storage	Table Storage	SimpleDB (beta)	DataStore
Random Read/Write	Azure Drives	Elastic Block Store	-

But the developers need to be aware of a few aspects in which consuming cloud based storage service varies from traditional data storage mechanisms. Cloud based storage services are mostly accessed using REST based APIs. The application developers need to be aware of how REST works and also be familiar with the REST based APIs supported by cloud storage providers.

The next important thing a developer should be aware of is the pricing model of cloud storage services. Although this seems more of a business concern, the application developer must be fully aware of the transaction charges (Cost of each request to the storage), bandwidth charges (Cost of incoming and outgoing data) and the storage charges (Cost per each gigabyte of data stored). Every time a request is made to the cloud storage the transaction and bandwidth usage meter ticks and one will be ultimately charged for it. Table-2 shows the cost model of cloud storage services provided by Amazon, Google and Microsoft. However one has the option of making use of data caching options to avoid frequent hits to the data storage account.

One should also keep in mind that the cloud storage is not local to the application. Hence some amount of latency should be expected by the developer. There are also limitations imposed on all

types of storage be it blob, table or queue storage. These limitations may vary from vendor to vendor.

If the application has to make use of standard file system APIs then the developers will have to make use of special drive storage services provided by the cloud storage provider (Azure XDrive or Amazon Elastic Block Store - EBS). But one has to be aware of the limitations of drive storage services. In Windows Azure only one application instance can have a write access to a particular drive at any given point in time. Other application instances can continue to have read access to the same drive. An application will also have to ensure that the drive is mounted before issuing any command to the drive storage.

Table-2: Cloud storage service charges.

	Windows Azure	**Amazon Web Services**	**Google App Engine**
Blob Storage Charges	$0.14 per GB stored per month	$0.125 per GB per month for first 1TB	$0.13 per GB per month
Storage Transaction Charges	$0.01 per 10000 transactions	$0.01 per 10000 GET transactions, per 1000 PUT, COPY, POST or LIST transactions	Write: $0.10 per 10000 Read: $0.07 per 100000 Small: $0.01 per 100k operations
Data Transfer Charges	$0.12 per GB per month	Data In: Free Data Out: First 1 GB / month - Free, Up to 10 TB / month - $0.120 per GB and so on.	$0.12 per GB per month

A decision for design time of the application would be whether to use structured data storage provided by cloud data providers (Azure Table or Amazon SimpleDB or Google BigTable) or to use cloud RDBMS services. If we look at the cost factor RDBMS services on cloud cost a lot more compared to structured data storage services and also the amount of data storage capacity provided by RDBMS services are pretty low compared to table storage counterparts. But when it comes to data access (using standard data access APIs), Portability (migrating the application and database back to organization premise), Transactions (Cross table and distributed transactions), Type of Data Types supported RDBMS services clearly have the upper hand over the structured non-relational storage services provided by cloud storage providers.

Third party tools such as CloudBerry Backup are available for backing up data from cloud storage accounts. Developers can also implement their own data backup programs.

4. SCALABILITY

When it comes to scaling an application up or down, most cloud providers provide their own scaling solutions. Microsoft's Windows Azure comes with a feature known as Elastic Scale which allows scaling of application via a minor configuration change without having to bring down the existing application. Microsoft also provides APIs through which the application can programmatically scale up or down based on some application logic.

An Amazon Elastic Compute Cloud instance can also be auto scaled up or down as per the demands of the Application that is hosted. An application can also be scaled based on pre-defined schedules. Dynamic scaling is achieved through Amazon Cloudwatch metrics. Amazon

Cloudwatch also has an option where in application can make use of Amazon Simple Notification Service (SNS) to send alerts before initiating auto scale and after completing the auto scale.

Applications hosted on Google App Engine are capable of utilizing technologies that Google applications are built on, things like BigTable and Google File System (GFS).

Since cloud applications are distributed in nature managing user sessions has to be implemented in ways that can support distributed environments. Storing sessions in application memory is not an option so one has to follow state management techniques which may include storing encrypted session state in a dedicated state server or in some other persistence storage. This could result in some form of application latency. Developers have to ensure that the application session objects are serializable so that they could be persisted.

Like state management, logging also differs because of the distributed nature of applications running in cloud environment.

5. DIAGNOSTICS

Tracing and diagnostics is integral part of in the lifecycle of any software. But the way tracing and diagnostics is handled on on-premise applications and cloud hosted applications varies slightly in some aspects.

One of the simplest forms of diagnosing an application hosted in production environment is by having some logging mechanism. In an on-premise application hosting scenarios, we would have the application log errors, exceptions and information to a text file or a database by using our own custom logging mechanism or by using third party logging frameworks such as NLog, JLog or kLogger for .NET, Java and PHP applications respectively.

For out of the box logging Windows Azure provides a diagnostics infrastructure which makes use of the .NET tracing mechanisms to log traces of information and errors. This allows application programmers to choose what gets logged and also gives them the option to transfer these logs to a persistent storage (using Azure storage services) on a timely basis. Similar diagnostics services are provided by Google AppEngine which internally makes use of JLog.

We could use the third party logging frameworks for a cloud hosted application as well. However the way these frameworks are configured within the application would change to some extent. For example if we have to use NLog for an application hosted on Windows Azure, we would have to implement custom NLog targets and integrate them with Windows Azure Diagnostics Infrastructure. Similarly if we have to use NLog in an application hosted on Amazon Web Services to send log reports via email then we would have to configure NLog to make use of Amazon Simple Email Service (SES).

Some of cloud service providers also provide the developers with remote debugging capabilities. For example Windows Azure provides IntelliTrace (Visual Studio 2010 Ultimate only). Azure Connect can also be used to achieve remote debugging in Windows Azure platform. Amazon Web Services ships a toolkit for Eclipse which helps the developers with remote debugging and VMWare is also working on an upcoming CloudFoundry feature that provides capability to remotely debug a Java application.

Almost all major cloud storage service providers give application developers with options to enable storage statistics, analytics and metrics. All of these services will be storing the storage statistics and logs in a predefined structure and these could be read using third party software like AWStats or our own custom APIs. These logs will contain data ranging from time of storage access to IP address of the client who made the request. Storage analytics could be used for audit trails purpose as well.

6. MANAGING RELATIONAL DATA

There may be scenarios where in an application would store data in a relational form into a database such as SQL Server, Oracle or MySQL. Along with highly scalable structured data storage such as Windows Azure's Table Storage or Google App Engine's DataStore, cloud service providers also provider cloud based relational database services.

Windows Azure provides relational database as a service through SQL Azure. SQL Azure is basically SQL Server for the cloud environment and supports majority of the features supported by SQL Server Enterprise Edition. The process of connecting to a SQL Azure database and querying against it remains largely similar to what one would do while making use of a SQL database in an enterprise environment. However SQL Azure does come up with some limitations which are well documented in Microsoft Developer Network (MSDN) Library.

Microsoft also provides tools to migrate an existing on- premise SQL database to SQL Azure which could be helpful in migration of an on-premise application using SQL Server to cloud. An existing on-premise data can be migrated to SQL Azure by using either the migration tool or other options such as SQL Server Integration Services (SSIS). Options are also provided for synchronizing a SQL Azure database with an on-premise database. Although SQL Azure provides a web based management portal, advanced database management can be achieved by connecting to a SQL Azure database via SQL Server Management Studio installed in an on-premise system.

Amazon provides its relational database services through Amazon Relational Database Service (Amazon RDS) where in one has a choice of MySQL or Oracle as his Relational Database Management Server. Amazon RDS takes care of patching and updating the database server software and also provides on demand database instances.

Based on the cloud database service chosen, one has to keep in mind whether each hit to the database is charged in terms of transactions and bandwidth. If it is charged, then it's up to the application developer to keep the database transactions to minimum by making use of macro queries wherever possible. Table-3 shows the cost associated with various cloud based relational database services.

Table-3: Cloud based relational database service charges

SQL Azure	Pricing is based on size of database chosen. For example, a database of size between 1GB to 10GB would cost $9.99 for the first GB and $3.996 for each additional GB
Amazon Relational Database Service (RDS)	Price depends on the type of database chosen (MySQL or Oracle) and the size of the RDS virtual machine.
Google Cloud SQL	Google's database service is not being billed currently

7. MIGRATING LEGACY APPLICATIONS

There are quite a few challenges while trying to migrate an on-premise application to cloud to leverage the benefits that various cloud service providers offer. We can broadly classify these challenges into the following:

The complexity of migrating an application database depends on what sort of cloud data storage service we choose to use. If our on-premise application uses RDBMS like MySQL or MS SQL Server then the migration would range from a minor configuration change to changes in database code such as stored procedures and triggers in case we are using on-premise database features that don't exist in cloud RDBMS service.

In Windows Azure one has the option of migrating a database installation to the cloud using VM Roles. However ports that are commonly used by database servers may not be open in a cloud environment. So the migration has to ensure that database server is configured appropriately. However, if we choose to migrate an existing relational database to one of the structured non-relational data stores, it would require major coding changes in the data access logic.

Most applications would have some sort of authentication and authorization mechanisms built in. Usually an on-premise application would carry out authentication and authorization against an application specific store of user details. If an application is making use of Active Directory Services, then Windows Azure and Amazon Web Services both provide options and workarounds to make the existing Active Directory infrastructure work after the application is moved to cloud environment. Cloud service providers also provide authentication via universal identity providers such as Google, Facebook and Windows Live which could also be an option for authentication in a cloud based application.

Scenarios where an application has to access applications or services hosted on-premise or partner organizations can also be migrated to cloud environment. Windows Azure provides Service Bus as a part of its App Fabric services which enables service calls and messages to pass through firewalls and NAT routers.

Deploying an on-premise application onto cloud environment may include certain challenges. These challenges will mainly depend on whether IaaS services are used or PaaS services are used. With IaaS such as Amazon Web Services, the migration of an existing deployment will be pretty straight forward with minimal effort.

However if we were to choose PaaS services such as Windows Azure, the migration challenges depend on the configuration and dependent applications/libraries that need be installed before the application deployment. An application that does not require any external libraries or OS configurations can be migrated very easily. Applications that require simple OS configuration such as environment variables setting and minor registry modifications can be achieved through start-up scripts which can be run in elevated mode in Windows Azure. However if the configurations are too many and cannot be done through start-up scripts then we would have to make use of Windows Azure VM Roles wherein we would be uploading a Windows Server 2008 R2 image with all the pre- configurations done. VM Role works almost same as other compute roles in Windows Azure however the work of updating OS and applying OS patches will have to be taken care by the cloud service user.

8. USE CASE TO MIGRATE ON-PREMISE APPLICATION TO CLOUD

The use case is about an energy producing plant. Let us assume there are multiple such plants installed in various regions. All the plants need to communicate to a centralized database. The

business layer should be able to scale up or down depending on the demand and save/fetch data from the database. A web application needs to constantly poll for new data from the plant and display onto the UI. The existing design is shown in figure 1.

In the existing design of the application, the User interface of the application communicates with the business logic over HTTP.

Migrating the application to AWS or Google App Engine would involve using different set of tools and techniques. Although migration of the UI components remains largely same, database migration complexity would depend on the RDBMS chosen.

The following steps are needed to migrate the same application and database onto Windows Azure:

1. Upload the existing application and Service to the cloud as a Web Role.
2. Get the Service URL and update the Reference in the application.
3. Migrate the on-premise database to SQL Azure using any of the following techniques.
    a) SSIS – SQL Server Integration Services
    b) SQL Wizard- Copy option.
    c) Data Sync from Azure Management Portal.

Figure-1: On-Premise design of the plant application

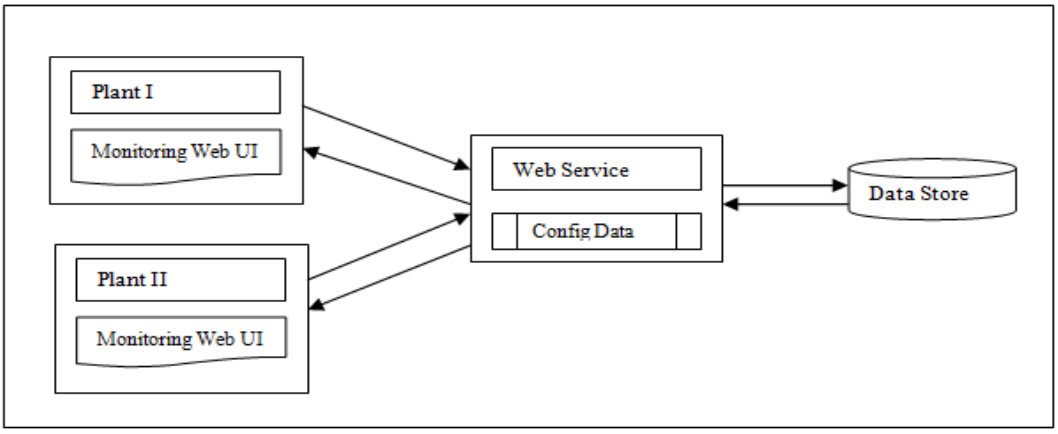
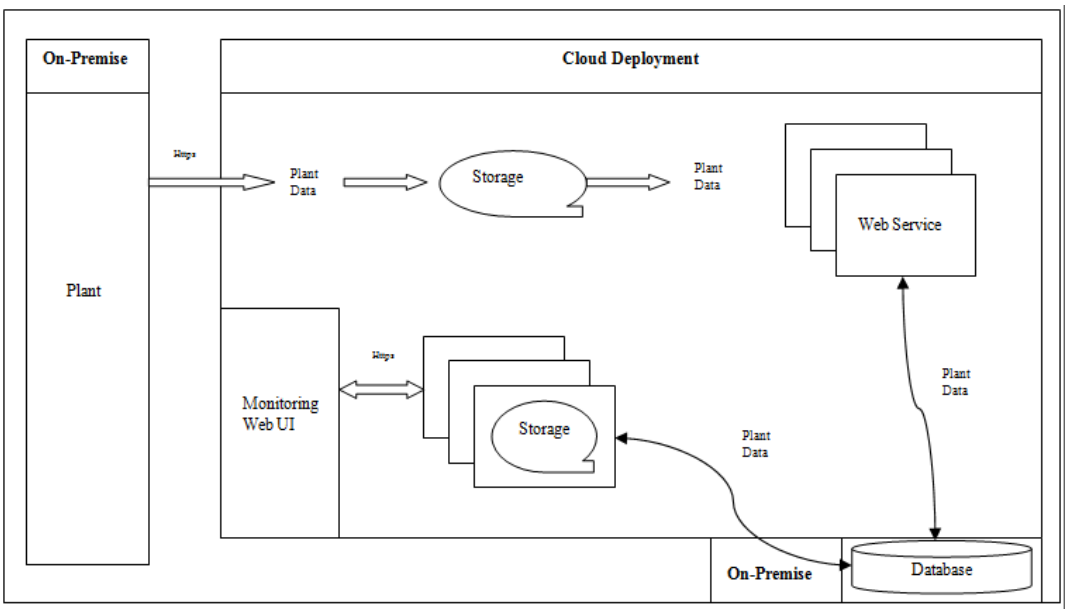


Figure-2 shows the design of the application after it was modified for migration to cloud.



9. CONCLUSION

Developing an application for cloud environment is not too different from the traditional on-premise application development. It's just the nuances of cloud computing platforms that the developers and architects need to be aware of. We have discussed several points in the course of this paper which shed light on issues that a developer or an architect faces while adapting to the latest advances in cloud computing.

REFERENCES

[1] Charlie Kaufman and Ramanathan Venkatapathy, "Windows Azure Security Overview".

[2] Jinesh Varia, "Architecting for the Cloud: Best Practices". [Online]. Available: http://media.amazonwebservices.com/AWS_Cloud_Best_Practices.pdf

[3] J.D. Meier, "Azure Security Notes". [Online]. Available: http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-48-03/0572.AzureSecurityNotes.pdf

[4] Jinesh Varia, "Migrating your Existing Applications to the AWS Cloud" [Online]. Available: http://media.amazonwebservices.com/CloudMigration-main.pdf

[5] "Integrating Applications with the Cloud on the Windows Azure Platform". [Online]. Available: http://wag.codeplex.com/

[6] David Chappell & Associates, "The Windows Azure Programming Model". [Online]. Available: http://www.davidchappell.com/writing/white_papers/The_Windows_Azure_Programing_Model_1.0--Chappell.pdf

[7] Creating HIPAA compliant Medical Data Applications [Online]. Available: http://awsmedia.s3.amazonaws.com/AWS_HIPAA_Whitepaper_Final.pdf