Fall 2010

A Clouded Future: Analysis of Microsoft Windows Azure As a Platform for Hosting E-Science Applications

Ian Blair
Regis University

A CLOUDED FUTURE: ANALYSIS OF MICROSOFT WINDOWS AZURE AS A PLATFORM FOR HOSTING E-SCIENCE APPLICATIONS

A THESIS

SUBMITTED ON 15 OF NOVEMBER, 2010

TO THE DEPARTMENT OF INFORMATION TECHNOLOGY OF THE SCHOOL OF COMPUTER & INFORMATION SCIENCES

OF REGIS UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF MASTER OF SCIENCE IN SYSTEMS ENGINEERING

BY

Ian Bla

Ian Blair

APPROVALS

James A Lupo

James A. Lupo, Thesis Advisor

Stephen D. Barnes

Douglas I. Hart

Douglas I. Hart

Abstract

Microsoft Windows Azure is Microsoft's cloud based platform for hosting .NET applications. Azure provides a simple, cost effective method for outsourcing application hosting. Windows Azure has caught the eye of researchers in e-science who require parallel computing infrastructures to process mountains of data. Windows Azure offers the same benefits to e-science as it does to other industries. This paper examines the technology behind Azure and analyzes two case studies of e-science projects built on the Windows Azure platform.

Acknowledgements

I would like to thank Dr. James Lupo for all his help and guidance with preparing this thesis. I would also like to thank all of the Professors at Regis University. Over the past two years I’ve seen a lot of growth in my understanding and appreciation for Computer and Information Sciences. Most of my development is due to mentoring and guidance I received while attending courses at this University. My heartfelt thanks go out to all the members of the University staff who have been my instructors and mentors.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

Cloud computing has been generating a lot of interest. It promises to revolutionize the way applications are hosted over the internet. Experts have even made predictions foreseeing a future where companies no longer house applications in their own datacenters. Instead, house them on cloud based server farms owned and managed by a hosting provider. In Cloud Security and Privacy (2009), author Tim Mather sees similarities between the cloud computing of the current “information age” and the “industrial age”. During the early industrial age, companies provided their own power through steam, windmills, and water generators. In the late industrial age this changed, because, he says, “with electrification, however, organizations no longer provide their own power; they just plug in to the electrical grid… cloud computing is really the beginning of the same change for information technology."

This is especially true for "e-science" applications. Cloud computing may usher in a new age of scientific discovery. In the past, scientific discovery has been hampered by the high cost of having to build or lease parallel computing infrastructures. Scientific applications that require the parallel processing muscle of stacks of networked servers were out of the reach of modestly funded research projects. The advent of cloud computing could lead to a new level of democratization in parallel computing, because cloud computing services offer scientists the power of parallel computing at significantly lower cost than having to build solutions using a traditional Windows HPC Server infrastructure. The ability to scale up or down based on demand further lowers costs for researchers, because processing power isn't wasted or left idle. It can be decommissioned or added as needed.

Windows Azure makes a suitable platform for hosting small and mid-size e-science applications. This study will look at two examples of how cloud computing reduces the cost and complexity of e-science applications. Through research, we will assess the value of the cloud infrastructure in terms of cost, scalability, and ease of maintenance over the traditional infrastructure.   This study argues that hosting parallel processing e-science applications on the Windows Azure Cloud is superior to hosting on the traditional infrastructure. The cloud infrastructure is cheaper and more cost effective. The cloud infrastructure is also better at adapting to changes and can scale more quickly.

After the introduction, there will be a review of related literature and a brief discussion of the methods used to build this thesis. In section 4.1, includes a brief discussion of the value Windows Azure could provide as a platform for hosting e-science applications, and a discussion of some of the costs associated with a traditional application infrastructure. In section 4.2, technical overviews are provided for Windows Azure, SQL Azure and Internet Information Server. Section 4.2 also includes a discussion of the cost benefits of Windows Azure. Additionally, section 4.2 includes detailed instructions of how to set up a Windows Azure development environment, migrate databases to SQL Azure as well as a production deployment on Windows Azure. These walk-throughs show the ease of creating a development environment and performing production deployments.

In section 4.3, two case studies are examined. We will first look at Azure blast which is an application designed for bioinformatics research. Windows Azure was responsible for reducing the cost and complexity when producing this research application. Then we'll examine an application designed to process the large amounts of data produced by MODIS satellite data re-projection and reduction pipeline. Section 4.4 will conclude chapter 4 with a comparison to Amazon’s EC2 cloud.

Cloud computing has a lot of advantages over the traditional infrastructure, and it may be the infrastructure of the future. It is the natural progression from infrastructures of the past. The distributed model similarly had its advantages over mainframe architectures, and over time distributed systems have become the primary platform for hosting applications. The same may happen with hosting applications in the cloud.

Chapter 2 – Review of Literature and Research

Using the cloud for e-science is in its infancy, and because of this, there are only a handfull of studies related to this subject. It’s an area that has an enormous amount of potential and can open HPC computing to a larger community. “Defining Future Platform Requirements for e-Science Clouds” (Ramakrisnan, 2010) closely matches the content in this study. However, they focus on a broader range of cloud computing offerings, and this study’s primary focus is Windows Azure. They begin by describing e-science projects, and explaining how cloud computing could be used to host these applications. They discuss the positives and negatives of ~~a~~ cloud computing offerings from Amazon and Yahoo. They conclude by offering suggestions on how to improve cloud systems to better serve the needs of the e-science community.

In “Bridging the Gap between the Cloud and an e-Science Application Platform” (Simmhand, 2010), the researchers create an application using Windows Azure to perform genome sequencing, and evaluate it as a platform for e-science applications. This study provides a good case study on how easily applications written for .net can be moved or integrated with services provided by Windows Azure. The authors describe Azure’s services and provide a detailed description of their application’s design.

Chapter 3 – Methodology

Cloud computing is relatively new technology. Very little has been written about Microsoft's Cloud operating system, Windows Azure. The research used for this thesis is a combination of interpretivist quantitative epistemology and case study analysis. From the point of view of the System Architect, this study researches the underlying technology behind Windows Azure. Through exhaustive examination of technical libraries, whitepapers, and other technical documentation a technical roadmap was created for implementing Windows Azure. I argue that Windows Azure is a suitable platform for hosting e-science applications, is less expensive and easier to maintain than the traditional HPC server infrastructure.

Interpretivist epistemology frees the researcher from the limitations associated with proving there is only one solution to a given problem. Instead, the interpretivist believes that his solution is more accurate than others. In a number of ways Windows Azure is more suitable for hosting e-science applications than the traditional Windows HPC infrastructure because it requires very little administration, has an abundance of scalability, and is less expensive. Azure requires no purchase of hardware, software licenses, or support staff to maintain the infrastructure. Two examples are provided within this study that solve real world science problems using the Windows Azure platform at far lower cost than building a traditional HPC infrastructure.

Chapter 4 – Project Analysis and Results

4.1 E-Science

4.1.1 E-Science in the Cloud

Moving e-science applications to the cloud opens up opportunities for scientific discovery to a wider community. Small or mid-range projects could build cloud based parallel computing applications that could scale to the size of large super-computing centers if needed. There are three ranges of e-science applications, bulk-scale large synchronous computation applications, bulk-scale synchronous mid-range applications, and asynchronous massively independent applications. (Ramakrishnan, 2010)

Bulk-scale large synchronous applications run on large scale clusters, with thousands of cores, at super-computing centers. These applications run long batch jobs and make use of large storage systems and networks. The communication requirements of applications within this range may be beyond the capabilities of most cloud infrastructures that have been primarily designed for hosting web applications. The network speeds may just be too slow and create performance bottle necks. Mid-range systems run similar jobs but on a smaller scale. These systems run on clusters that have tens to hundreds of cores, and are often privately owned and funded either by research groups or universities. Finally, asynchronous massively independent applications run on workstations or small clusters owned by scientists, groups, or universities for processing of sensor data or biometric analysis, and they often perform pre or post processing of data or create visualizations for analysis and reporting. (Ramakrishnan, 2010)

The second and third ranges would see the greatest benefit from a move to cloud computing. Cloud computing has several advantages for applications within these ranges.

Cloud computing requires no up-front costs or investment other than the monthly hosting fees. There is no need to purchase hardware or networking equipment. It doesn't require a staff of systems and network administrators to support the infrastructure. Also, it offers the ability to scale up supercomputing levels based on need. In section 4.4 Case Studies, we'll examine two case studies of asynchronous massively independent applications that have been built on Windows Azure. In the next section we'll discuss some of the types of applications Windows Azure is suitable for.

4.1.2 Windows Azure for E-Science

Windows Azure wasn't designed with HPC computing in mind, but because it has the ability to provide significant value for e-science projects, it has gotten the attention of the scientific community. As of this writing, Microsoft is providing grants to projects certified by the National Science Foundation (NSF) to use Azure for their research (Microsoft, 2010d). Because it wasn't specifically designed for HPC computing, it lacks many of the libraries and tools HPC researchers have at their disposal on a traditional HPC infrastructure.

There is no current port for Dryad/Linq, a popular HPC framework available on Windows HPC Server, nor does the Azure architecture have the bandwidth to support applications that rely on heavy MPI messaging. These applications are better suited for server farms with higher bandwidth interconnectivity. It's more suited for applications that run ensemble or parameter sweep studies. Azure is suitable for the following applications (Microsoft, 2010):

- Evaluation of cloud computing for a spectrum of algorithms, computational models, and programming patterns.

- Programming abstractions for cloud platforms that offer higher level abstractions and guarantees to application developers.
- Scaling conventional algorithms to cloud scale, coping with load and dealing with faults.
- Research into scientific data management, data analysis, and new services in the cloud to support data intensive research.
- Data parallel program frameworks, going beyond the map reduce model.
- Experimental platforms for distributed agent technology.
- Compiling declarative and/or functional languages for cloud platforms.
- Sustainable science gateways and virtual organizations in the cloud for web-based application and collaboration frameworks.
- Research to support digital data libraries in the cloud that contain scientific data, not just the metadata, and support integration with published literature.
- New techniques for data visualization of cloud scale data sets, including visual analytics.
- Frameworks to support sensor-based science in the cloud, from services to process data streaming from sensors to the storage and curation of raw and derived data products.
- Machine learning techniques for data understanding and data generation.
- Research to support intelligent interactions leveraging web data and domain knowledge.

4.2 Windows/SQL Azure

4.2.1 Windows Azure

Windows Azure is Microsoft's new cloud computing platform. The "cloud" in cloud computing refers to the internet, but more specifically, it refers to internet connected datacenters. In cloud computing, applications are hosted on virtual machines that run on a compute cluster made up of many commodity hardware systems. Resources can be added and subtracted as needed. There is a total abstraction between the virtual machine and its underlying hardware.

One of Windows Azure's benefits is having the ability to deploy applications without any purchase of equipment. All the hardware, monitoring, networking, and security required to host an application are provided by Microsoft. Applications are developed using a development version of the Windows Azure cloud "Fabric", and then deployed on the cloud. In Figure 1, it shows how the Windows fabric handles hardware resources and makes them available to services running in the cloud. This new architecture greatly reduces the cost of hardware and support for an application.

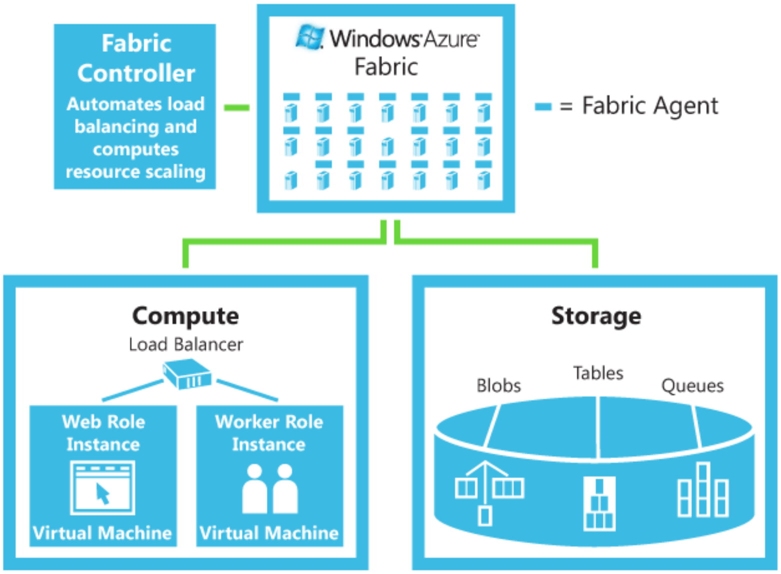


Figure 1, Azure Fabric Architecture

Windows Azure runs IIS 7.0 and is backwards compatible with most existing .net applications allowing for the easy migration of existing ASP/.NET applications to the Windows Azure cloud. In addition, Azure resources can be scaled quickly. This allows for quick responses to demand, and when resources are no longer needed, those resources can be decommissioned rapidly. All the management, capacity monitoring, and fault detection is done by Microsoft and is included in the pricing.

Windows Azure hosts applications in services running as virtual machines. These services can be scaled to multiple instances to provide redundancy or to meet demand. These services can assume one of two roles. The first, which is the web role, is designed to host IIS 7.0/ASP.net applications. The second role is a web worker role. This role cannot accept external incoming connections, but is designed, instead, to provide background processing for the web role. Web roles are fully secured and protected by Microsoft Trust Policies. These polices provide a policy based firewall for protecting applications running on the Windows Azure Cloud. (Microsoft, 2009a)

Lately, there has been much concern over the security of cloud architectures. “Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds by Ristenpart, Tromer, Shacham, and Savage, the authors state that “cloud infrastructures can also introduce non-obvious threats from other customers due to the subtleties of how physical resources can be transparently shared between virtual machines (VM).” They go on to suggest that by exploiting side-channels between virtual machines hosted on the same server, they were able to snoop into others running VMs. While this has been proven on Amazons EC2 cluster, they haven’t presented any evidence that the same could be done in the Windows Azure cloud.

Windows Azure requires two configuration files, the service definition file and the service configuration file. The service definition file, figure 2, provides the information for the configuration of the service, its storage, and SSL security certificates. This file cannot be modified or changed in a running service. The service must be decommissioned before any changes can be made and then re-commissioned.

```
<ServiceDefinition name="<service-name>" upgradeDomainCount="<number-of-upgrade-domains>">
  <WebRole name="<web-role-name>">
    <ConfigurationSettings>
      <Setting name="<setting-name>" />
    </ConfigurationSettings>
    <LocalResources>
      <LocalStorage name="<local-store-name>" cleanOnRoleRecycle="[true|false]" sizeInMB="<size-in-
megabytes>" />
    </LocalResources>
    <InputEndpoints>
      <InputEndpoint name="<input-endpoint-name>" protocol="[http|https|tcp]" port="<port-number>" />
    </InputEndpoints>
    <InternalEndpoint name="<internal-endpoint-name>" protocol="http" />
    <Certificates>
      <Certificate name="<certificate-name>" storeLocation="<certificate-store>" storeName="<store-name>" />
    </Certificates>
  </WebRole>
  <WorkerRole name="<worker-role-name>">
    <ConfigurationSettings>
      <Setting name="<setting-name>" />
    </ConfigurationSettings>
    <LocalResources>
      <LocalStorage name="<local-store-name>" cleanOnRoleRecycle="[true|false]" sizeInMB="<size-in-
megabytes>" />
    </LocalResources>
    <Endpoints>
      <InternalEndpoint name="<internal-endpoint-name>" protocol="[http|tcp]" />
      <InputEndpoint name="<input-endpoint-name>" protocol="[http|tcp]" port="<port-number>" />
      <InputEndpoint name="<input-endpoint-name>" protocol="https" port="<port-number>"
certificate="<certificate-name>" />
    </Endpoints>
    <Certificates>
      <Certificate name="<certificate-name>" storeLocation="<certificate-store>" storeName="<store-name>" />
    </Certificates>
  </WorkerRole>
</ServiceDefinition>
```

Figure 2, Service Definition File (Microsoft, 2009b)

The service configuration file, figure 3, provides values that the service can use while running on Windows Azure’s “fabric.” This file includes the number of instances deployed for each of the service roles. This file can be redeployed without affecting the running service.

```
<ServiceConfiguration serviceName="<service-name>">
  <Role name="<role-name>">
    <Instances count="<number-of-instances>" />
    <ConfigurationSettings>
      <Setting name="<setting-name>" value="<setting-value>" />
    </ConfigurationSettings>
    <Certificates>
      <Certificate name="<certificate-name>" thumbprint="<certificate-thumbrint>"
thumbprintAlgorithm="<algorithm>" />
    </Certificates>
  </Role>
</ServiceConfiguration>
```

Figure 3, Service Configuration File, Retrieved from (Microsoft, 2009c)

Windows Azure also includes storage in the cloud for blobs, tables, and queues. These data storage features pale in comparison to SQL Azure which is a fully relational database based on Microsoft's SQL Server 2008 product. Windows Azure's storage doesn't support relations between tables or referential integrity of any type, but allows you to store data close to your application. Azures blobs, queues, and tables are used very heavily in the case study e-science applications. They provide close, scalable storage for data. The data is fully redundant and secured by policy. Windows Azure offers more reliability, due to redundancy, than an application hosted on a single Windows Server. It also offers the advantage of using Microsoft's security instead of having to build and pay for a local security solution.

4.2.2 IIS

IIS is the web server integrated into Windows Azure. E-science platforms require the use of a front end to submit jobs to the scheduling engine. IIS is often used to host the portal that the researcher uses to submit jobs, monitor jobs, and retrieve results. IIS is a robust web server, and is fully capable of scaling to any number of users. Its tight integration with the Windows operating system gives it good performance, security, and reliability.

The architecture of IIS consists of three major components, two of which run under the local system account as a user mode process and the third, http.sys, is a kernel mode process. The HTTP.sys listener, the W3SVC service, and the Windows Process Activation Service comprise the web server. HTTP.sys, IIS's kernel mode listener process, handles requests made from clients to the web server, and it passes these requests to the W3SVC service. The W3SVC service works closely with the WAS service who spawns processes to process these requests. HTTP.sys is a fast kernel mode process that can handle requests quickly without having to switch to user mode, which would be much slower. It provides caching of requests, request prepossessing, and security filtering.

The W3SVC and WAS both run under the svchost.exe process. The W3SVC service provides HTTP administration and configuration, process management, and performance monitoring. This service reads the configuration information from the IIS meta database and uses it to configure the listener service. It provides counters for Windows integrated performance monitor to monitor IIS processes and behavior. It also manages the IIS worker processes and is responsible for starting, stopping, and recycling them.

Lastly, the Windows Process Activation Service (WAS) monitors and manages application pools and worker processes. Application pools allow you to isolate IIS applications running on the same server. Application pools give you control over starting, stopping, and recycling a pooled application without affecting other applications running under the same process. The WAS service allows you to integrate Windows IIS with other listeners such as Windows Communication Foundation (WCF) or other products. (Phar, 2007)

The complete IIS is process diagramed in figure 4. HTTP requests are handled by HTTP.sys in kernel mode. These requests are then passed to the W3SVC service which asks the

WAS service to check the configuration information for the requested application. Using this information, it spawns a worker process within the SVCHOST.exe process for the request.

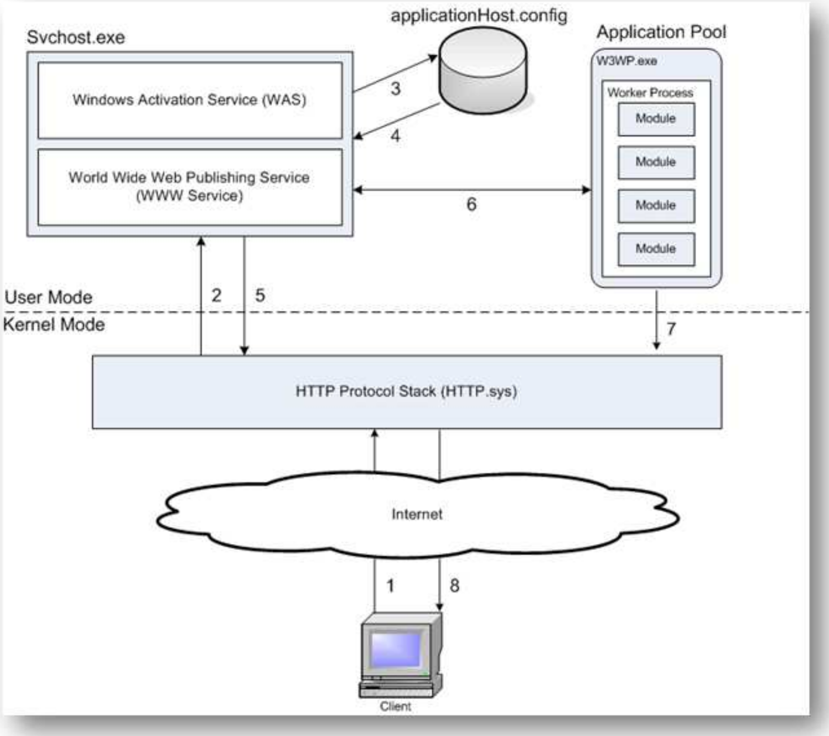


Figure 4, Overview of a HTTP Request, (Phar, 2007)

4.2.3 Development Environment

Building a Windows Azure development environment is far simpler and more cost effective than building a Windows Server based development environment. Microsoft provides an Azure SDK that allows the installation of a local fabric on the developer's workstation or on a Windows Server. Using the SDK, applications can be run on a development version of the Windows Azure "fabric." Applications built and run on the development fabric behave as if they were running on the production "fabric" hosted by Microsoft. It requires a computer running either Windows Vista/7 or Windows 2008 server and Visual Studio 2008 SP1 or 2010 beta. (Microsoft, 2009i)

SQL Azure doesn’t allow the creation of a local development version of their product. Testing requires the creation of a service on the SQL Azure Cloud or local SQL Server instance. However, it is possible to create multiple services for development and/or staging in addition to the production service if needed. A production SQL Azure service would include a service URL, database name, username, and password. The Application, hosted either on the Windows Azure cloud or on the development cloud can access the data via HTTPS using this URL.

If an application’s database was built and tested on a local Microsoft SQL 2008 server, using SQL Server migration tools to perform a migration is quite simple. See section 4.2.6 Migrating Database. Using the SQL server Management studio’s Import and Export data feature, data and schema can be exported from a SQL Server 2008 server directly to SQL Azure.

Table 2 has a listing of all the virtual machine sizes.

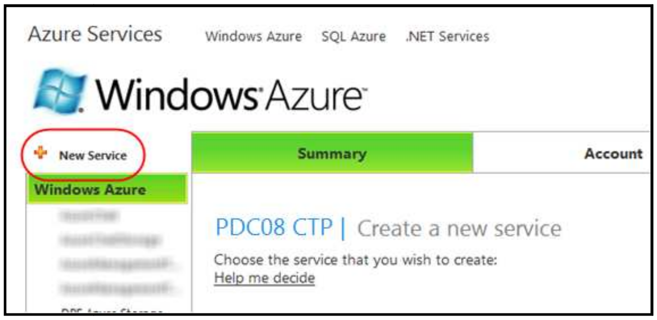
Table 1 Virtual Machine Sizes (Microsoft, 2009m)

VM Size	CPU Cores	Memory	Disk Space for Local Storage Resources
Small	1	1.7 GB	250 GB
Medium	2	3.5 GB	500 GB
Large	4	7 GB	1000 GB
ExtraLarge	8	15 GB	2000 GB

4.2.4 Deploying in Production

Deploying production applications on Windows Azure is even simpler than creating a development environment. The production deployment of a Windows Azure application is simple and elegant. Visual studio offers tight integration between the IDE development environment and the development "fabric". In order to release an application in production it must first be built. Visual Studio's build option creates two files, one with a .cspkg extension and one with a .cscfg extension. These two files are all that is needed to deploy applications on the cloud.

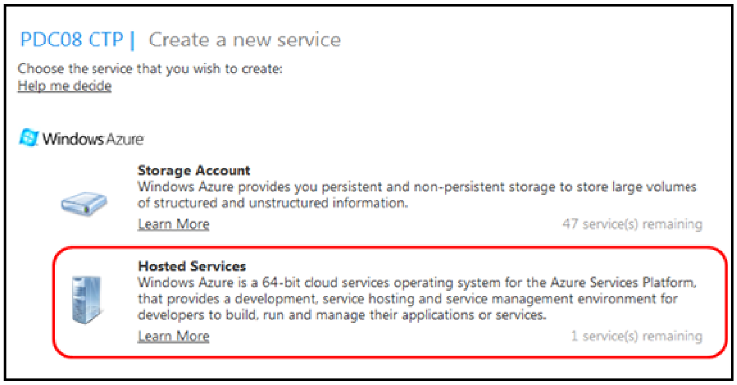
Deploying an application on the cloud requires a login to the windows Azure account portal, and selecting new service, as shown in figure 5. The next page allows the management of hosted services and storage accounts. If the application requires blob, tables, or other simple storage services they can be managed by selecting Storage Account. Hosted services tasks such as creating Web and Worker Role services are managed by selecting Hosted services, see figure 6. It then requires that the name of the service and the service description are entered, Microsoft, 2009I.



PDC08 CTP | Create a new service

Choose the service that you wish to create:
Help me decide

Figure 5, New Azure Service (Microsoft, 2009l)



PDC08 CTP | Create a new service

Choose the service that you wish to create:
Help me decide

Windows Azure

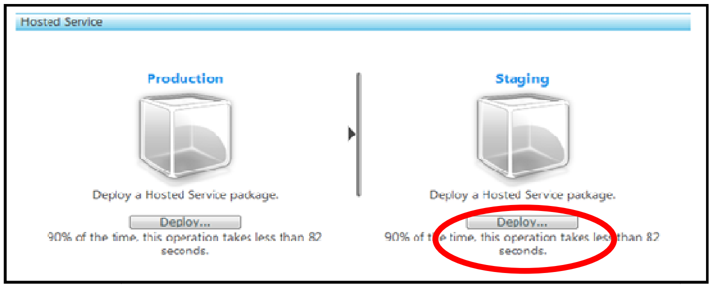
Storage Account
Windows Azure provides you persistent and non-persistent storage to store large volumes of structured and unstructured information.
Learn More 47 service(s) remaining

Hosted Services
Windows Azure is a 64-bit cloud services operating system for the Azure Services Platform, that provides a development, service hosting and service management environment for developers to build, run and manage their applications or services.
Learn More 1 service(s) remaining

Figure 6, Azure Hosted Services (Microsoft, 2009l)

After entering the name and description of the service, a URL must be entered for the hosted service. The application can be assigned to an affinity group. Assigning the application to an affinity group ensures that the web role VM and storage are created within the same geographic area, Microsoft, 2009I. This will reduce the amount of distance between an application's web role and its storage. Keeping them local provides better application performance.

After completing the service, configuration applications must be uploaded. An application can be deployed in the staging environment by clicking deploy under the staging icon as shown in **Error! Reference source not found.**. This will open the staging deployment page. Enter the path of the .cscfg and .cspkg as well as a version number for the application, as shown in **Error! Reference source not found.**. (The .cscfg and .cspkg files are created by Visual Studio when building a package for a Windows Azure Project, discussed in the Development Section) Next, run must be selected to start the staging VM, Microsoft, 2009I.



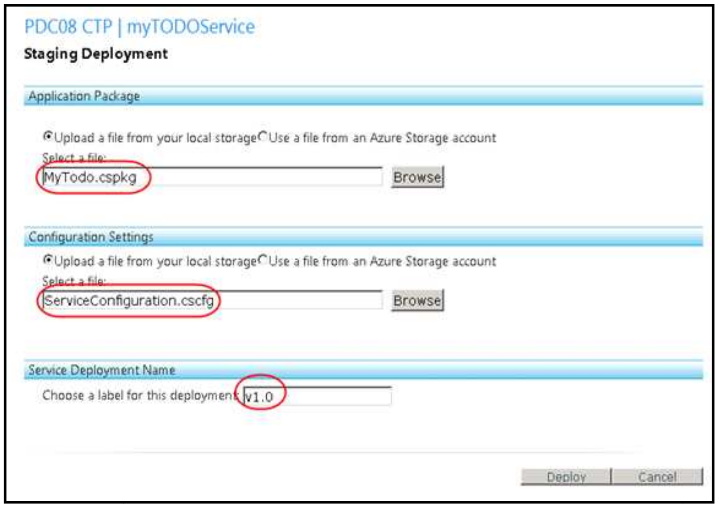
Deploy a Hosted Service package.

90% of the time, this operation takes less than 82 seconds.

Deploy a Hosted Service package.

90% of the time, this operation takes less than 82 seconds.

Figure 7, Hosted Service (Microsoft, 2009l)



PDC08 CTP | myTODOService

Staging Deployment

Application Package

● Upload a file from your local storage ○ Use a file from an Azure Storage account

Select a file: MyTodo.cspkg

Configuration Settings

● Upload a file from your local storage ○ Use a file from an Azure Storage account

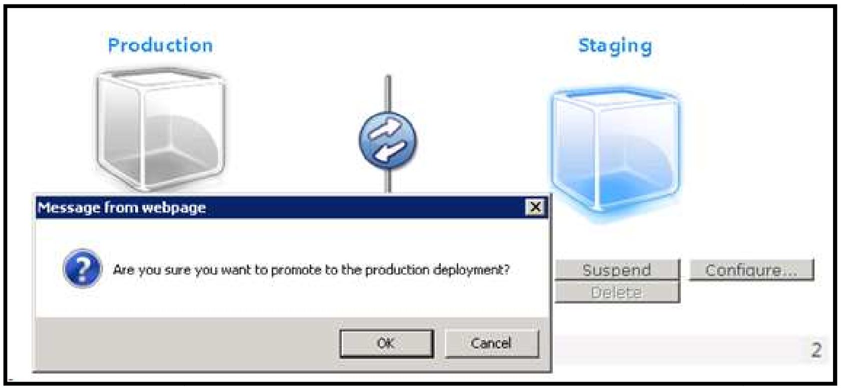
Select a file: ServiceConfiguration.cscfg

Service Deployment Name

Choose a label for this deployment: v1.0

Figure 8, Staging Deployment (Microsoft, 2009l)

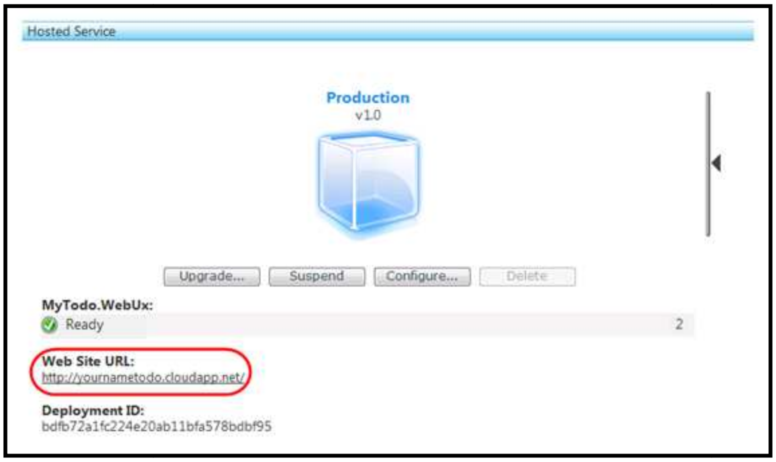
After successfully testing the application in staging, it can then be deployed into production by pressing the switch button in the center of the hosted application page. A message box will pop up, asking if the application should be moved to production, figure 9, Microsoft 2009I. Selecting OK will release the application onto the production Azure Fabric. Microsoft 2009I. The next page presents the user with a URL for the production website, figure 10. The deployment process is simple, and leaves little room for error. This package deployment method is far simpler and less error prone than deploying using traditional methods.



Message from webpage

Are you sure you want to promote to the production deployment?

Figure 9, Production Deployment 1 (Microsoft, 2009l)



MyTodo.WebUx:
Ready

Web Site URL:
http://yournametodo.cloudapp.net/

Deployment ID:
bdfb72a1fc224e20ab11bfa578bdbf95

Figure 10, Production Deployment 2 (Microsoft, 2009l)

4.2.5 SQL Azure

SQL Azure is Microsoft's database in the cloud. It's a fully relational, SQL database, based on their SQL Server 2008 DBMS. It is entirely a different product to Windows Azure, and it has a different architecture and pricing model. It can be integrated with a Windows Azure application on the backend allowing for the creation of a multi-tier application with the data hosted by Microsoft in their datacenters.

The difference between Microsoft's SQL Server 2008 product and SQL Azure is that SQL Azure is a service. The DBA or customer is totally abstracted from the underlying hardware and storage of the database. All database maintenance and storage administration is done by Microsoft. Customers, however, do have full control over logical databases, users, and roles, but Microsoft's SQL Azure team will manage the servers. This may simplify the administration and support costs for e-science customers who may lack the services of a professional DBA. SQL Azure can be accessed through Microsoft's policy based firewall for external clients or internally by Web roles running on Windows Azure.

SQL Azure does lack some of the features included with Microsoft's full SQL Server 2008 DBMS such as analysis services, replication, reporting services, and the service broker. Applications that require these features would be more at home on the traditional infrastructure. SQL Azure is, however, easier to use, support, and maintain. New database services are provisioned and decommissioned online relatively easily without the costs and administrative overhead of dealing with physical hardware, storage, and the Windows Server operating system. (Microsoft, 2009d)

Microsoft SQL Azure is hosted on SQL Server technology running in Microsoft's datacenter, and is comprised of four key layers. **Error! Reference source not found.**11, has a breakdown of the 4 key layers which are Client Layer, Services Layer, Platform Layer, and Infrastructure Layer. The client layer runs either a Windows Azure application or on-site local application server. The same connection tools used with Microsoft's traditional database products can be used to connect to SQL Azure.

The services layer is a gateway between the hosted application and Microsoft's SQL Azure service. It provides the billing, metering, hosting, provisioning, and connection routing for SQL Azure services. Because the client is totally abstracted from the hardware hosting the database, this layer provides a single point of access to the data. On the other side of this layer is the platform layer. The platform layer consists of instances of Microsoft SQL 2008 servers that make up the "SQL Azure fabric". Lastly, the infrastructure layer consists of the physical servers, Windows Server 2008 operating systems, and the network interconnects that support the SQL Azure platform. (Microsoft, 2009d)

SQL Azure's capacity may be too small for the needs of some e-science applications which would be better served by using the table or blog services to store data. SQL Azure is very useful for data smaller than 10GB, such as user accounts, application queues, and reports. The case study applications discussed with this thesis make heavy use of Windows Azure's blog and table storage to store application related data.

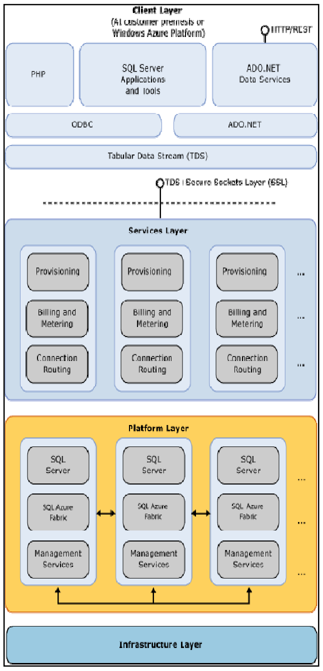


Figure 11, SQL Azure Architecture (Microsoft, 2009e)

Windows Azure and SQL Azure can be easily integrated to create a single solution. **Error! Reference source not found.**12, is a diagram showing the architecture Windows Azure and SQL Azure. The client can connect to the data directly via ADO.NET, ODBC, or PHP from an application installed on their local workstation or server. If an application is running on a Web role, the web role can connect directly to SQL Azure within Microsoft's cloud infrastructure using HTTP or HTTPS. As an option, the web role service can connect via SSL or TDS to the SQL Azure service internally for increased security. Both the Windows Azure web role service and the SQL Azure service are protected externally by configurable Microsoft firewall policies.

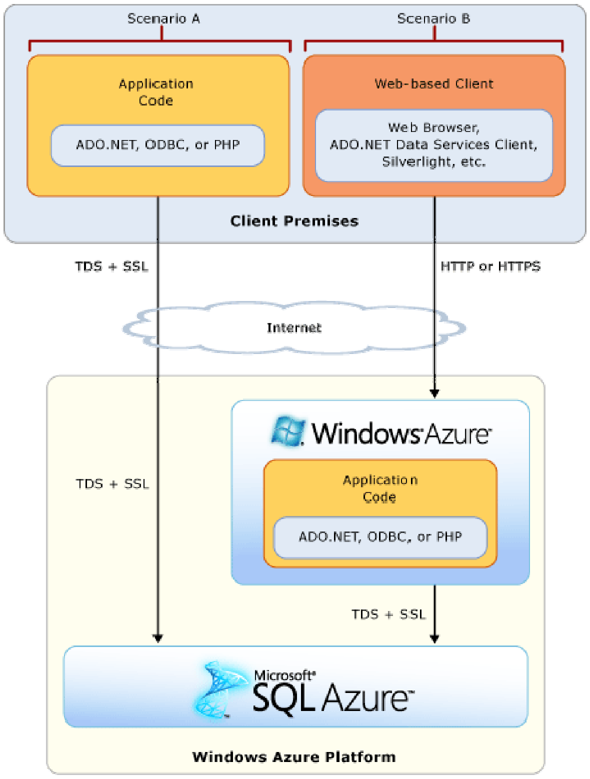


Figure 12, Windows, SQL Azure Integration (Microsoft, 2009g)

4.2.6 Migrating Databases

E-science applications may require that databases be moved from their laboratories or other scientific organization sites to the cloud or vice versa. There are several ways to migrate existing databases to SQL Azure. SQL Azure uses the same administration tools as Microsoft SQL Server 2008. One of the simplest ways of migrating data would be to use the “Generate Scripts” option in SQL Server Management Studio (SSMS). This feature will generate a single script that includes the database schema and data. Alternatively, you can use SSMS’s Import Export Wizard or the bcp.exe utility. These utilities require a few more steps and are more complicated.

The Generate Scripts wizard is by far the simplest approach. By right clicking the database object in SSMS’s “Object Explorer” and selecting tasks, then “Generate Scripts”, it will open a generate scripts dialog box. All objects should be selected. SQL Azure does not support user-defined data types, extended properties, Windows authentication, or the USE statement. The following options should be configured (Microsoft, 2010e):

- Convert UDDTs to Base Types = True
- Script Extended Properties = False
- Script Logins = False
- Script USE DATABASE = False
- Script Data = True

In the resulting script all instances of “SET_ANSI_NULLS ON” should be deleted. All instances of the following clauses should also be deleted, "WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]". Also, for any create table statements the "ON [PRIMARY]" clause should also be deleted.

To re-create the database on SQL Azure, connect using SQL Server Management Studio by selecting Query -> Connection ->Change Connection. Fill out the appropriate connection and authentication information for the SQL Azure instance. Create the database using a "CREATE DATABASE" SQL statement. Then run the generated script.

4.3 Cost of Azure

4.3.1 Azure Costs

The cost of Windows Azure depends on the size and number of the instances. Windows Azure supports four different virtual machine sizes, see table 2. The small (default), medium, large, and extra large instances are 12, 24, 48, 96 cents per hour respectively. The storage costs 15 cents per GB of storage per month, and additionally 1 cent per 10,000 transactions per month.

SQL Azure has two different options. The Web Edition allows the storage of up to a 1 GB relational database. The Business Edition allows storage of up to a 10 GB relational database. The Web Edition is $9.99 per month, and the Business Edition is $99.99 per month. Data transfers for North America are 10 cents per GB in and 15 cents per GB out. For the Asia Pacific region data transfers are 30 cents per GB in and 45 cents per GB out.

The cost of the development environment for Windows Azure is free. A development fabric can be downloaded from Microsoft and run on either a Windows desktop or server. It's also a free addition to Microsoft Visual Studio 2008 and 2010. Windows Azure includes a staging instance with a separate URL, see **Error! Reference source not found.**, at no additional cost. Applications can be staged and tested on a different staging virtual machine, and then later promoted to the production virtual machines. Microsoft provides a service level agreement guaranteeing 99.95% uptime for Windows Azure and 99.96% for SQL Azure.

4.3.2 Traditional Server Infrastructure Support and Maintenance

The traditional Windows Server 2008 infrastructure requires servers as well as the required network infrastructure to interconnect them. Often the traditional infrastructure is mostly centralized and housed within datacenters equipped with cooling, UPS systems, and high speed WAN connectivity. According to Microsoft's TCO report (Microsoft, 2010b), the traditional infrastructure has the following shortcomings:

1. It requires the purchase, leasing, and management of assets both hardware and software.
2. Network infrastructure devices must be purchased and/or leased, supported, and managed by internal staff or managed services providers.
3. Bandwidth must be purchased and managed for WAN and internet connectivity.
4. Systems and network management tools must be purchased and implemented to monitor network and server performance, availability, security, and configurations.
5. Servers require internal staff or managed service providers to assure that assets are properly installed, configured and optimized. These tasks include managing physical server installs and changes, operating system installs and patches, virtualization management, security management, systems monitoring and configuration. When a hardware device has an issue, technicians need to quickly mitigate the issue and affect a repair.
6. Storage needs to be managed to assure that data is available, high performance, recoverable and secure.
7. A security infrastructure must be in place to prevent network intrusions, viruses, and to ensure data security.

8. Datacenters require physical support and maintenance including managing physical security / access, managing air handlers / cooling systems, managing power, and managing change.

4.4 Case Studies

4.4.1 Azure Blast

In the paper "Azure Blast: A Case Study of Developing Science Applications on the Cloud", Windows Azure is used to provide the processing and infrastructure for an e-science application. Azure Blast is a web portal that performs bio-informatics analysis. It is designed to search for similarities between nucleotide or peptide sequences. BLAST uses an algorithm to search against a database of sequences and looks for similarities between the query sequence and subject sequence. Using the database, supplied by the National Center for Biotechnology Information (NCBI) or another scientific organization, it can identify the function of the query sequence.

The Azure Blast application is very computationally intensive because it has to compare millions of sequences. "A Blast query over a normal size genomics database can take several hours or even days." (Lu, 2010) The application must compare millions of sequences stored in a very large database, close to 10GB. Azure handles the load by dividing the workload amongst many worker roles that are responsible for processing only part of the data and sending their results back to the management worker role where the results will be combined.

Applications that could provide these services were, in the past, out of the reach of smaller projects. Azure allows modestly funded research projects the ability to use almost unlimited processing power without the purchase of expensive servers and network equipments.

Jobs submitted to Azure Blast run in batches. Jobs are submitted to a portal where the researcher can monitor the job and view the results, see figure **Error! Reference source not found.**. Jobs are stored within a table and are queued to prevent conflict or loss due to role failure. The scheduler retrieves the job from the table, schedules its execution and maintains the jobs state.

The job scheduler maintains a global queue called a "dispatch queue" in which the worker roles pole against for available work. The cloud fabric will handle all resource management and health management for all roles. The application is totally unaware of these functions. This system allows for greater flexibility and scalability. The application is unaware of how many worker roles are performing the batch processing. The number of roles determines the performance of the batch process. Adding additional roles will shorten the processing time. This flexibility allows a research project to scale based on the funding for each project. A low funded project may be equipped with just a handful of roles, while a highly funded project can be scaled to thousands of worker roles.

The Azure blast application tasks are forked to child tasks. Each worker process is responsible for its own child task, and upon successful completion of its task, the data is returned to the parent role and kept in a completion queue. Waiting can be synchronous or asynchronous depending on the needs of the batch job. A synchronous execution is preferred for quick tasks and asynchronous for long tasks. A failure of the child processes doesn't affect the parent process because of the queue. A failed child task will be re-issued to another child until the parent process has been completed.

The BLAST application first splits the input data and stores it in an Azure blob. It then assigns a blast task to each partition of data. Each of these tasks runs the NCBI blastall binary against the data, sends the results to Azure blob storage, and then sends a completion message to the queue. After all child tasks have completed their jobs, a merging task gathers and processes the results. The merging task then returns the merged results to the blob queue and the scheduler is notified of the completed job.

The researchers were able to test the application scaled up to 64 large (8 core) instances. They saw a marked improvement in processing time of sequences when scaling to multiple instances. On a database of 2.8GB they saw an increase of 56x when comparing one instance to 64. A ten day batch job could be completed in just over four hours. One of the best values of using Azure for this type of application is that the processing power is only needed during the batch operation. Developers could conceivably build this application using just a couple of instances for testing, and scale up prior to running a larger job. After the completion of the job retire unused instances. Traditional HPC infrastructures lack this flexibility, and would require purchased hardware to remain idle until needed.

The effect of using cloud computing services on e-science projects could be profound. In the long term, by reducing the costs associated with parallel computing and increased access to raw CPU power in the form of multiple VMs, it may lead to greater scientific discovery. The lower cost should give a wider community access to resources that were only reserved for the very few. "This offers the promise of "democratizing" research as a single researcher or small team can have access to same large-scale compute resources as large, well-funded research organization without the need to invest in purchasing or hosting their own physical infrastructure." (Lu, 2010)

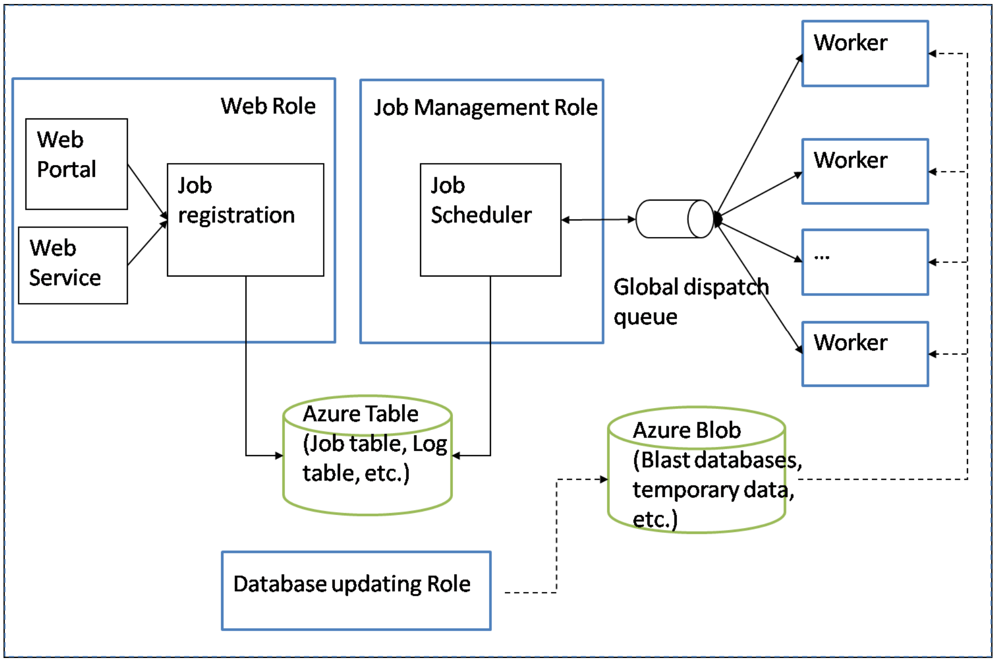


Figure 13, Azure Blast Application

4.4.2 A MODIS Satellite Data Re-projection and Reduction Pipeline

In the study "eScience in the Cloud:A MODIS Satellite Data Reprojection and Reduction Pipeline in the Windows Azure Platform", Windows Azure is used to produce an e-science application to provide processing of MODIS satellite data. MODIS data comes from the Terra and Aqua satellites. They circumnavigate the globe and record changes in atmosphere, land, and ocean using sensors that record electromagnetic waves. The data is sent back and made available via several FTP sites. Scientists from geology, environmental, oceanographic, and many other sciences use this data in their studies.

The MOIS system has major problems. The data collected is published to a number of different FTP sites in three main groups, air, land, and water. Each day hundreds of different files are sent to these sites making data collection an enormous task. The data published from the different groups are in different formats. Before making use of this data, it requires resource intensive filtering and transformation. The development skills required to transform this data is complex enough that it would require the skills of a software developer. Lastly, if a study requires the collection of data over time, 10 years for example, it would require substantial amounts of storage. For this study up to 2TB.

Scientists today have the ability to collect large amounts of data, but there are several hurdles when it comes to turning this data into useful forms. "A scientist's commodity workstation will quickly be overwhelmed for ~~a~~ non-trivial computations over large datasets. Purchasing larger computational platforms (i.e. clusters) - although arguably the norm - is generally not cost-effective even if the money is available, as they quickly become outdated and especially if additional system administrators must be hired by the domain scientists just to manage the infrastructure." (Li, 2010)

As part of their study they created an application called the MODIS data re-projection and reduction pipeline. It's responsible for collecting data from the various FTP sites, transformation of the files, and finally production of analysis/reports. A researcher logs in to a research portal, submits jobs, monitors the job, and then collects the data from the results. The operation of the MODIS application is very similar to Azure Blast discussed in the last section.

The architecture of the application is also very similar to Azure Blast. A researcher submits a request through a web portal. The applications service manager fetches the request and begins the data retrieval process. The data retrieved from the FTP sites is stored in Windows Azure blob storage. The application then processes the data based on the requester's specifications which include format and filtering options. When the job is finished, the researcher is notified by email.

Like the Azure Blast application, this application leverages the power of the Azure platform allowing scalability in the number of worker VMs. Each worker fetches tasks from a queue, when complete, they return their portion of the larger job to a completion queue. A service manager VM maintains the status of each job and coordinates the work of each worker process. This application has been tested for performance on a dual core windows desktop and 1, 50, 100, and 150 Windows Azure instances. The results for the test are listed in *Table 2 Application Performance Chart*.

Again, there is evidence of Windows Azure's value for e-science projects. This applications' services are designed to be resold to a scientific community who use this data in their studies. Instead of building an application, they can use this service to more quickly produce reports, even if they lack the expertise or funding to store or translate this data into a usable form on there own. The e-science application provider sees a cost reduction due to not having to procure the hardware and networking infrastructure required to host this service. They can also scale this service based on demand, up or down, to ensure they are able to provide the best price. When demand is high they may provide 200 VMs to process requests in the queue, and when demand is down they can equip their application with only 50 VMs. The overall effect is a reduction in cost for science projects.

Table 2 Application Performance Chart

	MOD04_ L2 (hrs.)	**MOD06_L2 (hrs.)**	**MYD11_L2.005 (hrs.)**
Desktop	16.29	72.62	33.45
Single instance	34.21	116.19	63.56
50 instances	0.76	2.25	1.12
100 instances	0.40	1.20	0.61
150 instances	0.30	0.85	0.44

4.5 Alternatives to Azure

4.5.1 Amazon EC2

While Windows/SQL Azure is considered to be a platform as a service (PAAS), Amazons EC2 service is considered an infrastructure as a service (IAAS). Windows Azure provides one platform, the IIS/.NET/SQL Server software stack, for hosting applications. Amazon's service provides more options and flexibility. It allows you to use a number of different operating systems, application frameworks, and DBMS's to build your infrastructure on their cloud. The flexibility they offer comes with a higher price tag.

Amazon's EC2 acronym stands for Elastic Compute Cloud. Its elasticity refers to its ability to scale. Like Windows Azure, it allows the provisioning of as many instances or VMs as needed which can be scaled up or down with little effort. The EC2 cloud allows the running of pre-configured Amazon Machine Images (AMIs) that run many of the major operating systems, application frameworks, and database management systems. It also provides an option to build and store custom AMIs with customized software. Pre-built AMIs for Windows 2003/2008, Open Solaris and many flavors of Linux such as Redhat, SUSE, Ubuntu, Oracle, Gentoo, and Debian are available. The available DBMSs include IBM DB2/Informix, Microsoft SQL Server 2005, MySQL, and Oracle 11g. The available application servers include JBoss Enterprise, Ruby on Rails, IBM WebSphere, Java Application Server, and Oracle Web Logic.

Amazon EC2 VM can be attached to additional storage using Amazon's Elastic Block Store service at a pay per size/utilization fee. VMs can be hosted in multiple geographic locations to reduce the latency and provide redundancy. VMs can run in the US East (Northern Virginia), US West (Northern California), EU (Ireland), or Asia Pacific (Singapore) availability zones. Amazon secures internet connected instances using a web-interface configurable firewall, and provides an Amazon Virtual Private Cloud service that uses IPsec VPN to graft a virtual infrastructure on to an existing local infrastructure.

Amazon promises an uptime of 99.95% which is comparable to Windows Azure. Amazon's other services include Elastic IP Addresses, which provides load balancing for VMs running on the cloud, and Amazon Cloud Watch, which provides a monitoring solution for AWS. Amazon Cloud Watch provides important metrics, such as CPU utilization, disk reads/writes, and network usage. Like Windows Azure these can be accessed through web service APIs for integration. In addition, EC2 provides auto scaling to automatically adjust for demand. Based on pre-set conditions, EC2 will automatically spawn more instances. When demand decreases, it will scale back down. Windows Azure doesn't currently have the ability to scale automatically.

Amazon's EC2 is priced very similarly to Windows Azure except for one key difference, the Database. Amazon's 32-bit small instance (default), with 1GB of memory, 1 processor core, and 160GB of local storage is 12 cents per hour. The 64-bit large instance, with 7.8GB of memory, 2 processor cores, and 860GB of storage is 48 cents per hour. The 64-bit extra large instance, with 15GB of memory, 4 processor cores, and 1,690GB of storage, is 96 cents per hour. These prices are similar to Windows Azure. Amazon, however, doesn't provide a service like SQL Azure. Amazon allows you to provision a VM with SQL Server 2005. The lowest price for a SQL server instance is $1.08 per hour. Yearly, this would cost $9,461, which is significantly higher than SQL Azures largest instance which has a yearly cost of $1,199. (Amazon, 2010)

While Amazon allows the use of more platforms, middleware, and DBMS's it is more complex and requires more administration than Windows Azure. Windows Azure uses a single platform with a single software stack and all maintenance of the individual VMs are done by Microsoft. The customer is totally abstracted from the VM architecture. Amazon is a VM hosting service, and VMs require some of the same maintenance as a full server implementation, see section 4.3.2 Traditional Server Infrastructure Support and Maintenance, for a complete list. Because of Amazons flexibility in giving the customer the ability to use a variety of platforms; it makes a good choice for e-science applications. Many existing e-science software applications and frameworks could be more easily ported to Amazon, because of the implementation of fully accessible VMs.

4.5.2 Windows HPC Server

The Windows 2008 HPC (High Performance Computing) infrastructure is very similar to any business network. It consists of Windows servers configured as a cluster. A High speed network consisting of switches, routers, and caballing is required to connect the workstations and servers. Routers are required to connect the entire system to other networks if needed.

Microsoft HPC server is a version of the Windows operating system that has been tailored for HPC computing. It consists of the Windows Server 2008 operating system and a special tool set designed to assist developers and administrators to create and support HPC applications (Microsoft, 2010c). The license price for Windows HPC is $479USD for each node. The standard version of Windows 2008 server is $1,029 (Enterprise $3,999), and may be required if Active Directory, hosting SQL Server, or other network services are needed. Using SQL 2008 database servers would require even higher upfront costs. Microsoft SQL Server 2008 Standard licenses are $7,171 (Enterprise $27,495) per CPU. (Microsoft, 2010d) Using

Windows/SQL Azure requires no license costs, no building of facilities, or hiring of support staff.

Chapter 5 – Conclusions

Scientific discovery in data intensive fields depend on HPC computing. Traditionally, researchers had two options. Option 1, build expensive, difficult to manage cluster systems. These systems require the purchase of network equipment and servers; in addition, they require staff to maintain and support them. When not in use these investments would remain idle, and after a few years would become obsolete requiring the purchase new hardware.

Option 2, lease the services of a supercomputing center. This option is less costly than option 1, but may require long waits in a queue and/or lack of flexibility when it comes to platform or software. The processing center may have restrictions, like platform or framework, that prevent or limit the ability to use their infrastructure. Windows Azure presents a third and very viable option.

Using Windows Azure, HPC applications can be built using the cloud infrastructure requiring no up-front costs and are easily scalable up or down. While Windows Azure is in its infancy, and doesn't allow the use of all the tools and applications that a Windows HPC Server would, it still is a viable option for HPC applications. Its ability to scale allows researchers to provision more VMs when needed. After a research project, VMs can be returned instead of sitting idle incurring charges.

This paper described the features of Windows Azure as well as provided walk-throughs of development and production deployment. We discussed features SQL Azure. In addition, using two case studies we showed the value of Windows Azure for hosting e-science applications. Azure allowed them to develop e-science applications without building or having to support an parallel computing infrastructure. Azure provided a low cost, zero maintenance application platform that could grow or shrink based on demand or budget.

References

Vecchiola, C. & Chu, Xing Chen (2009). Aneka: A Software Platform for .NET-based Cloud Computing. Melborne, Austiralia: The Univeristy of Melborne, Australia

Armburst, M. (2009). Above the Clouds: A Berkeley View of Cloud Computing. Berkeley, CA: UC Berkeley Reliable Adaptive Distributed Systems Laboratory

Maenpaa, J. (2009). Thesis: Cloud Computing with the Azure platform. Hesinki, Finland: Helsinki University of Technology

Mather, T., Kumaraswamy S. & Latif, Shahed (2009). Cloud Security and Privacy. Sebastopol, CA: O'reily Media.

Phar, R. (2009). Introduction to IIS 7.0. Retrieved on November 3, 2009 from http://learn.iis.net/page.aspx/101/introduction-to-iis-70-architecture/.

Microsoft (2009a). About Windows Azure. Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/dd179442.aspx.

Microsoft GFS (2009). Securing the Cloud. Redmond, WA: Microsoft.

Microsoft (2009b). Service Definition Schema. Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/ee758711.aspx

Microsoft (2009c). Service Configuration Schema. Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/ee758710.aspx.

Microsoft (2009d). SQL Azure Overview**.** Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/ee336241.aspx.

Microsoft (2009e). SQL Azure Architecture. Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/ee336271.aspx

Microsoft (2009g). SQL Azure Data Access. Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/ee336239.aspx

Microsoft (2009f). SQL Azure Firewall. Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/ee621782.aspx.

Microsoft (2009h). Pricing. Retrieved on November 3, 2009 from http://www.microsoft.com/windowsazure/pricing/.

Microsoft (2009i). Installation and Troubleshooting. Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/ee460770.aspx

Microsoft (2009j). How to: Connect SQL Azure Through ASP.NET. Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/ee621781.aspx

Microsoft (2009k). Creating a New Cloud Service Project. Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/ee405487.aspx.

Microsoft (2009l). Deploying an application using the Windows Azure Developer Portal. Retrieved on November 3, 2009 from http://channel9.msdn.com/learn/courses/Azure/Deployment/DeployingApplicationsinWindowsAzure/Exercise-1-Deploying-an-Application-Using-the-Windows-Azure-Developer-Portal/.

Microsoft (2009m). Virtual Machine Sizes. Retrieved on November 3, 2009 from http://msdn.microsoft.com/en-us/library/ee814754.aspx.

Microsoft (2010). FAQ About Azure for Research. Retrieved on October 10th, 2010 from http://research.microsoft.com/en-us/projects/azure/faq.aspx

Microsoft (2010b). Microsoft Windows Azure Platform TCO Analysis. Retrieved on October 10th, 2010 from http://www.cloudave.com/1207/windows-azure-tco-calculator/

Microsoft (2010c). Windows Server HPC Server 2008 R2. Retrieved on October 10th, 2010 from http://www.microsoft.com/hpc/en/us/product/windows-hpc-server-2008.aspx.

Burrows, J. (2008). Azure: Can Microsoft Meld Windows with the Web? Retrieved on December 6th, 2009 from http://www.businessweek.com/technology/content/oct2008/tc20081027.

Dreyfuss, J. (2008). Microsoft Unveils Cloud Strategy Called Azure. Retrieved on December 6th, 2009 from http://www.redherring.com/Home/25346

Hardy, Q. (2009). Azure thing? Retrieved on December 6th, 2009 from http://www.forbes.com/global/2009/1116/technology-microsoft-azure-system-a-zure-thing.html?feed=rss_author

Hunter, P. (2009). Cloud Computing: Clouds Aloud. Retrieved on December 6th, 2009 from http://kn.theiet.org/magazine/issues/0916/cloud-computing-0916.cfm (Engineering and Technology Magazine).

Ristenpart, T., Tromer, E., Shacham, H., Savage, S. (2009). Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds. San Diego, CA: Dept. of Computer Science and Engineering, University of California and Cambridge, MA: Computer Science and Artificial Intelligence Laboratory Massachusetts Institute of Technology

Lu, W., Jackson, J., & Barga, R. (2010). AzureBlast: A Case Study of Developing Science Applications on the Cloud. Redmond, WA: Microsoft Research.

Li J., Agarwal, D., Humphrey, M., Van Ingen, C., Jackson, K., and Ryu, Y., (2010). eScience in the Cloud: A MODIS Satellite Data Reprojection and Reduction Pipeline in the Windows Azure Platform. Berkeley, CA: UC-Berkeley.

Simmhan, Y., Van Ingen, C., Subramanian, G., Li, J. (2010) Bridging the Gap between the Cloud and an eScience Application Platform. Bloomington, IN: Indiana State University; Charlottesville, VA: University of Virginia

Ramakrishnan, L., Canon, S., Cholia, S., Jackson, K, Shalf, J. (2010) Defining Future Platform Requirements for e-Science Clouds. Indianapolis, Indiana: Association for Computing Machinery.

Microsoft (2010d) "Microsoft and the National Science Foundation Enable Research in the Cloud" retrieved on October 16th, 2010 from http://www.microsoft.com/presspass/press/2010/feb10/02-04nsfpr.mspx

Microsoft (2010e) "How to: Migrate a Database by Using the Generate Scripts Wizard (SQL Azure Database)" retrieved on October 16th, 2010 from http://msdn.microsoft.com/en-us/library/ee621790.aspx

Amazon (2010) "Amazon Elastic Compute Cloud (Amazon EC2)" retrieved on October 16th, 2010 from http://aws.amazon.com/ec2