THE UNIVERSITY OF
MELBOURNE

Department of
Computer Science and Software Engineering

Term Paper

for

COMP90024 (433678): Cluster and Grid Computing

Version: 1.0
May 30, 2012

Copyright notice

Department of Computer Science and Software Engineering
The University of Melbourne
Victoria
AUSTRALIA
3010

ICT Building
111 Barry Street
Carlton

Tel: +613 8344 1300
Fax: +613 9348 1184
`http://www.csse.unimelb.edu.au/`

Version: 1.0
May 30, 2012.

Credits

This document was written by Terence Siganakis (134860), Scott Ritchie (330975), Andrew Vadnal (326558), and Adam Whiteside (327705).

CONTENTS

ONE

Introduction

Over the course of the last decade outsourcing the delivery of computational resources to external companies has become more and more popular for personal, business and scientific uses. In a manner analogous to water, gas and electricity, computing power can be treated as a resource and charged based on individual usage [4]. This delivery of computing resources as a service over the Internet is known as Cloud computing.

Cloud computing has a number of benefits for end users wishing to access and use external resources:

- Resources can be accessed from anywhere; users are no longer tied down to their machine in a physical location.
- Users do not have to maintain their own hardware, other than a machine to act as a portal to these utilities.
- Users are able to access much greater computing resources than would otherwise be available to them.
- Users no longer have to worry about losing data if their machine fails as it is distributed across the Cloud.

1.1 Cloud Computing Architecture

Cloud computing can deliver services to both end users and developers. These services extend to a number of different levels, providing resources which are beneficial to all types of users. [4]:

- **Software as a Service (SaaS):** Individual applications which make use of the cloud for either storage of computing power.
- **Platform as a Service (PaaS):** A platform for development across a distributed network of machines is provided to the end user.
- **Infrastructure as a Service (IaaS):** Off-site hardware infrastructure is provided to a number of end users to securely access and use for their own purposes.

In this report we will be focusing on the PaaS and IaaS services available to developers, rather than looking at SaaS solutions.

TWO

Related Works

In this report four cloud computing services are examined; Google App Engine, Microsoft Azure, Amazon EC2, and Manjrasoft Aneka. Each of these offers a different type, and level of services to developers to run applications on the cloud.

Name	Description	URL
Google App Engine	Google App Engine is a PaaS/IaaS which allows developers to develop and deploy automatically scaling web applications on Google's Infrastructure	`https://developers.google.com/appengine/`
Microsoft Azure	Microsoft Azure is a PaaS/IaaS which facilitates both application hosting and storage on a globally spanning network of Microsoft hosted data centres	`http://www.windowsazure.com/en-us/`
Amazon EC2	Amazon EC2 is a IaaS which provides instances; customisable virtual machines with differing specifications for end users to use as cloud based computers. Each instance comes with a small amount of storage space.	`http://aws.amazon.com/ec2/`
Manjrasoft Aneka	Aneka is a PaaS which simplifies the development of distributed applications on private or public cloud infrastructures	`http://www.manjrasoft.com/products.html`

Table 2.1: Description of cloud computing services

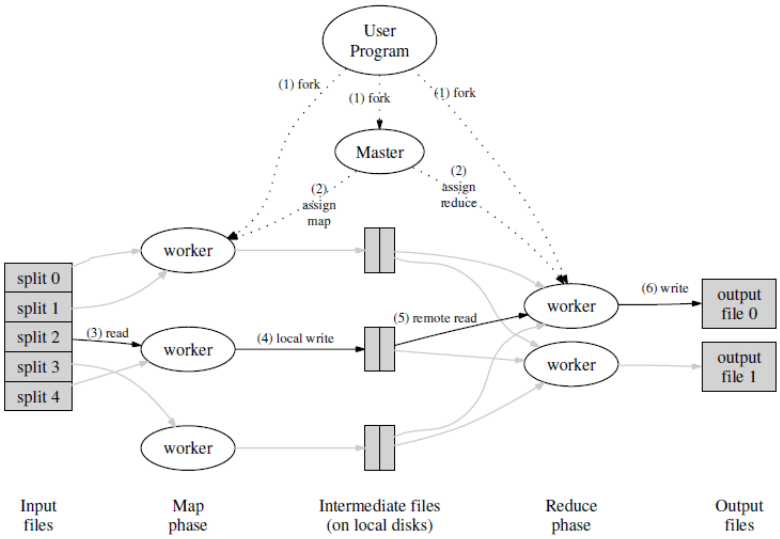
THREE

Google App Engine

3.1 Description

Google App Engine is a combined PaaS and IaaS cloud computing environment provided by Google. It provides a platform that allows developers to develop and deploy distributed web applications on top of Google's infrastructure, abstracting all notion of parallelism away from the developer. Google takes care of all the parallelism behind the scenes while guaranteeing a fast, scalable environment for developers to work in[1].

While Google is secretive about the underlying architecture of its servers, Google App Engine is most likely deployed across Google's servers using some variation of *MapReduce*. Developed in the late 1990s, *MapReduce* is a method for distributing programming tasks across multiple processes. *MapReduce* is based on functional programming paradigms, specifically the *Map* and *Reduce* functions typically found in such languages. Developers supply a list of data, and a function to apply to that data. This is the *Map* step in *MapReduce*. Another function is also provided by the developer, to be used in the *Reduce* step, which combines the resulting list of data to obtain the final result. All parallelism is handled by *MapReduce*, abstracting all notion of parallel programming away from application development. This is *MapReduce*'s strength and what allows Google to provide guaranteed automatic scaling to users of Google App Engine. [2]



`Overview of MapReduce execution.` [2]

[1]Google App Engine. *Why App Engine*, Retrieved May 14, 2012. https://developers.google.com/appengine/whyappengine

3.2 Application Runtime Environment

Google App Engine provides three runtime environments to develop and deploy applications in; Python (2.5.2 and 2.7.2), Java (SDK 5 or 6), and Google's own language, Go (experimental). Developers can also use other JVM based languages such as Ruby and JavaScript in the Java runtime environment. A local development environment which emulates Google App Engine is also provided.

Google does not provide the standard runtime environments for these languages. Instead restricted versions of the respective standard libraries are provided, giving Google the ability to efficently scale and sandbox the runtime environments. The restrictions to the standard libraries are as follows[2]:

- Computers can only connect to Google App Engine over HTTP(S) on standard ports.
- No filesystem access.
- Applications can only access other computers using Google's provided web services.
- Code can only run in response to a web request, queued task or scheduled task.
- Applications must take no longer than 60 seconds to respond to a request.

In addition to this restricted standard library, Google also provides a number of services:

- APIs for authenticating users through Google Accounts.
- An admin console for maintaining existing applications
- Image manipulation

3.3 Data Storage

As Google restricts the standard libraries of its runtime environments to prevent reading and writing files, they instead provide two other methods for persistent concurrent data storage: Memcache and the Datastore.

3.3.1 Memcached

Memcached is a service provided by Google which provides a high performance, in memory, key-value store that can be accessed across multiple instances of an application[2].

[2]Google App Engine. *What Is Google App Engine?*, Retrieved May 14, 2012. https://developers.google.com/appengine/docs/whatisgoogleappengine

3.3.2 Datastore

Datastore is a automatically scaling database-like service which allows for persistent storage within an application. The Datastore is a NoSQL schemaless object datastore, allowing for persistent storage of Objects.

Objects in the datastore are entities with a *kind* and a set of a properties. Using Java as an example, objects can be persistently stored in the datastore; the class is the entity's *kind* and its properties are the object's attributes. These can be stored and accessed using either JDO or JPA's persistence managers and query engines[2].

The Datastore is based on the *High Replication Datastore* design: a system based on the *Paxos algorithm* is used to replicated data across multiple servers. The datastore is strongly consistent and uses optimistic concurrency control[3]. This means that transactions can proceed without locking because it assumes that multiple transactions can be completed without affecting each other. When a commit is made to the datastore each transaction checks that no others have modified the relevant part of the datastore since the transaction started. If changes are found, the committing transaction is rolled back[4].

3.3.3 External Services

While the Datastore is automatically scalable and provides high performance, it does not guarantee atomic transactions where relations between objects exist. Some situations require relational algebra, so Google also provides access to its relational SQL database service known as *Cloud SQL* from within Google App Engine[2].

Google also provides access to its *Cloud Storage* system, which allows for storage of files up to a terabyte in size. This however cannot be accessed from the Google Go runtime environment[2].

3.4 Cost

Google App Engine can be used for free, giving any developer 1 GB of storage and up to 5 million page views a month. Once either of these limits are exceeded service to an application will be cut off for that month unless billing options are implemented. Pricing for Google App Engine's services is complex, containing varying costs for reads/writes to the datastore and hours per instance required. A minimum of $2.10 per week must be spent if the billable services are used, though developers are able to specify a maximum daily budget[5].

[3]Google App Engine. *Using the Master/Slave Datastore*, Retrieved May 14, 2012. https://developers.google.com/appengine/docs/java/datastore/usingmasterslave

[4]Google App Engine. *Datastore Overview*, Retrieved May 14, 2012. https://developers.google.com/appengine/docs/java/datastore/overview#Datastore_Writes_and_Data_Visibility

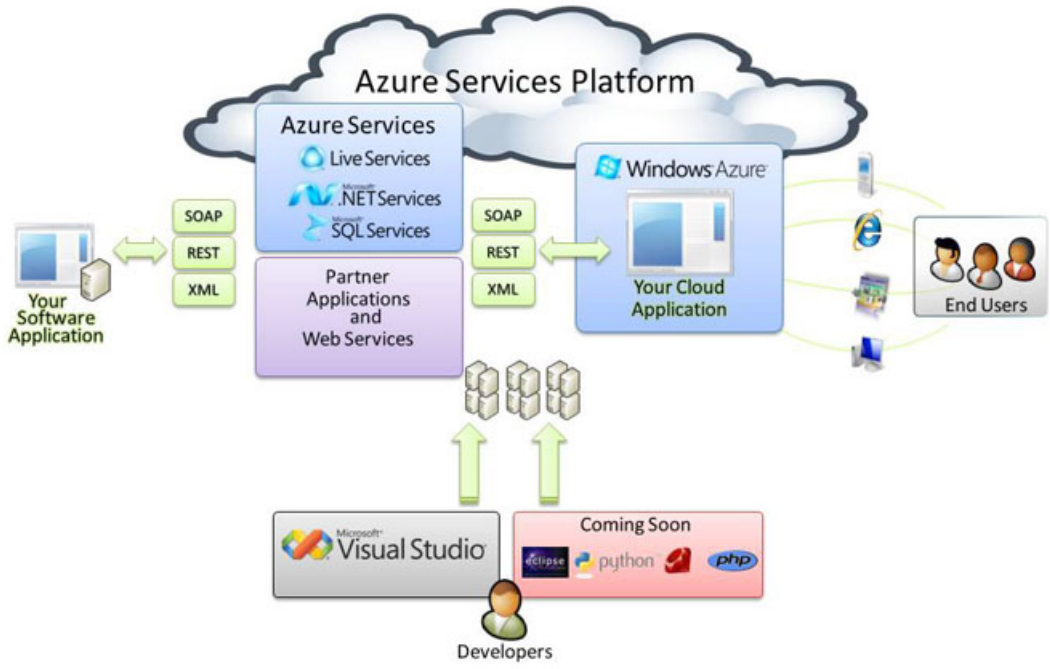
[5]Google App Engine. *Billing and Budgeting Resources*, Retrieved May 14, 2012. https://developers.google.com/appengine/docs/billing

FOUR

Microsoft Azure

4.1 Description

Microsoft Windows Azure can be perceived as an operating system provided to users by a service. This ensures that Azure can be used from anywhere, providing that the OS is hosted on a Microsoft-based cloud. Officially released on the 1st of February 2010, the Microsoft Windows Azure software is a cloud computing platform on which web applications can be readily built (using languages such as Java and Python, and frameworks such as .NET and Ruby on Rails), hosted and scaled across the globally spanning network of Microsoft hosted data centres. On-demand services are also hosted on these data centres, notably Windows Azure[6], SQL Azure and Windows Azure AppFabric. The following figure displays a high level architectural outline of the Azure cloud:



`High level architectural outline of the Azure cloud`[7]

Windows Azure is an OS that facilitates both application hosting and data storage. It is upon these two services that users will build and maintain their products once an Azure

[6]Windows Azure. *Home Page*, Retrieved May 14, 2012. http://www.windowsazure.com/en-us/

[7]Temi Odurinde. *Windows Azure – Microsoft Clouds Computing*, Retrieved May 14, 2012. http://www.temi.co.uk/windows-azure-microsoft-clouds-computing/

subscription has been purchased. SQL Azure is a scaled, cloud based implementation of Microsoft SQL Server – a relational database server and Windows Azure AppFabric represents a collection of cloud computing services at the middleware level.

4.2 Available Services

Notable services are included within AppFabric include, but are not limited to[8]:

- Access Control Service – Controls user authorisation on related services and applications.
- Service Bus – Ensuring that secure connections are in place on cloud-based applications.
- Caching – Allows applications to utilise the high-speed caching service provided by the Microsoft cloud; ensuring a high access time to application data.

4.3 Cost

Establishing a competitive price for a cloud-based service is of utmost importance, especially with other notable cloud providers such as Google App Engine and Amazon EC2 in the same consumer space. Microsoft Azure offers three pricing packages[9] that are in place for the Azure cloud service. These packages include:

- Consumption-based – Costs are incurred only when data is used.
- Subscription-based – Committing to a fixed price over a specified number of months.
- Volume-based – Tailored for large organisations that already possess at least one Microsoft license – allowing for these services to be integrated into the Microsoft cloud.

4.4 Security

The implementation of rigorous security measures are of critical importance, particularly within a cloud-based service which is available to consumers on a global scale. When users subscribe to the Azure cloud, the credit card used to transfer funds is associated to the user making the payment. In addition, access to the cloud service is performed via signing in through a Windows Live ID account.

Once a consumer has signed up to the Azure cloud, enforcing a secure means of user authentication is paramount to the security of any stored data. In order to differentiate

[8]Wikipedia. *AppFabric*, Retrieved May 14, 2012. http://en.wikipedia.org/wiki/AppFabric

[9]GigaOM. *Everything You Need to Know About Microsoft Azure*, Retrieved May 14, 2012. http://gigaom.com/2009/07/14/microsoft-azure/

between application hosting and data storage, Windows Azure uses separate methods used to authenticate users to each service. These methods are outlined in the following table [3]:

Subjects	Objects	Authentication Mechanism
Customers	Subscription (Compute & Storage)	Windows Live ID
Developers & Operators	Windows Azure Portal/API	Live ID (Windows Azure Portal) or Self-signed certificate (SMAPI)
Role Instances	Storage	Storage account key
External Applications	Storage	Storage account key
External Applications	Applications	Customer–defined

Accessing applications hosted on the Azure cloud can be performed by two methods: via the Windows Azure website itself or via the Service Management API (SMAPI). The latter requires the registration of a public/private key and a self-signed certificate associated to each user utilising the cloud service when uploading developed applications. SMAPI authentication references the aforementioned key pair and user certificate before each user session can commence.

When accessing stored content on the Azure cloud, users require an account-specific Storage Account Key (SAK). This key is able to be changed at any time.

The Windows Azure platform must provide fundamental security concepts on its cloud service, in order for consumers to acknowledge that their data is stored safely on the Microsoft cloud. Confidentiality, integrity, availability and accountability are the terms used to describe how security is approached on the Azure platform. Notable features pertaining to each of these terms are as follows [3]:

- **Confidentiality:** Encryption is used internally within Windows Azure for protecting control channels and is provided optionally for customers who need rigorous data protection capabilities.
- **Integrity:** Each Virtual Machine (VM) is connected to three local Virtual Hard Drives (VHD). The first VHD contains one of several versions of the Guest OS. The consumer selects the new patches they wish to apply, ensuring that the OS is always at the most recent version. The second VHD contains an image constructed by the Fabric Controller (FC). The last VHD contains configuration data, all paging files and storage.
- **Availability:** The Azure platform ensures that stored data will not be compromised by employing rigorous backup techniques. These techniques include Data replication and Geo-replication[10].

[10]Windows Azure. *Windows Azure Storage*, Retrieved May 14, 2012. http://www.windowsazure.com/en-us/home/features/storage

- Data replication involves replicating the data three times in the same data centre as a precaution against hardware failure in turn ensuring that data is always available.
- Geo-replication involves replicating data between data centres that are not situated within geographic vicinity. This approach safeguards against any natural disasters.

Adopting the Microsoft Windows Azure software is a viable solution for both a personal or business need to utilise a distributed cloud computing service to store and maintain developed applications. As with moving towards any new software, it is up to the consumer to adequately research many factors such as cost, security, data storage capabilities and data availability when making use of a cloud based hosting service. The consumption and subscription based pricing packages are ideal for consumers who are both unsure of how much data they will need and consumers who wish to move to the Microsoft cloud for a defined month period.

FIVE

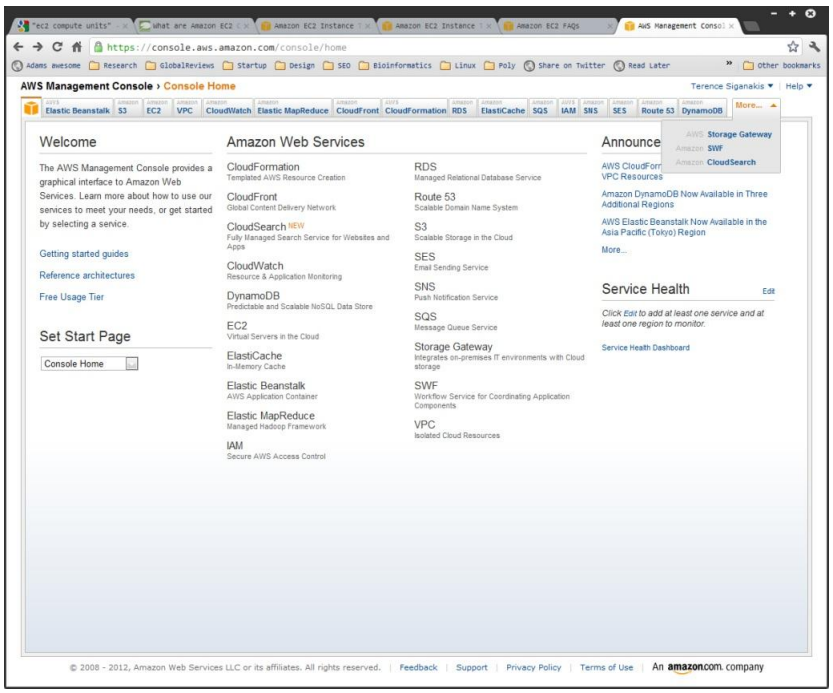
Amazon EC2

5.1 Description

Amazon Elastic Compute Cloud (EC2) has been in operation since 2006[11], when it commenced its first limited public beta of the service. Since then EC2 has proven extremely popular, and is currently the largest commercially available cloud service with some estimates putting the total number of servers at over 500,000+ servers[12].

The EC2 product offering changes nearly monthly, with an enormous variety of products as a part of its offering, including databases (DynamoDB), private clouds (VPC), auto scaling, load balancing (ELB), caching, etc. In the interest of brevity, this report will focus on the core product offering, the ability to create and utilise virtual servers.

The following image displays the login page to the EC2 console, with all the different services that can be administered through the console.



[11] Wikipedia. *Amazon Elastic Compute Cloud*, Retrieved May 14, 2012. http://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud

[12] Network World. *Amazon uses nearly half a million servers to power EC2, researcher estimates*, Retrieved May 14, 2012. http://www.networkworld.com/news/2012/031612-amazon-ec2-257384.html

5.2 Instances

The key abstraction used with Amazon EC2 is that of an "instance". An instance is essentially a single computer with a set of specifications. The specifications of instances have their own peculiarities. Each instance type has a set amount of memory, and a set number of "EC2 Compute Units". According to the documentation an EC2 Compute unit is described as:

> The amount of CPU that is allocated to a particular instance is expressed in terms of these EC2 Compute Units. We use several benchmarks and tests to manage the consistency and predictability of the performance of an EC2 Compute Unit. One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor[13].

In the same document, they describe another instance type as having 2 x Xeon X5570 quad core processors which equates to 33.5 compute nodes. So we can assume that a single Xeon X5570 quad core processor is about 16 Compute Units.

EC2 Instances support a variety of different operating systems, including Windows, Linux. In fact it is even possible to create your own instance types by uploading your own VMWare / VirtualBox image.

Once an instance has been created, it is a fully functional computer to which you have full access (e.g. root in Linux, Administrator in Windows).

There is no limit to the number of instances that you can spawn, aside from cost. In fact one enterprising company built a 30,000 core cluster (3,809 instances) using EC2 (which can be rented out for $1,279 per hour)[14].

5.3 Disk Access

Instances themselves have a small amount of storage associated with them which is essentially utilised for the OS installation. User data is stored using Amazons "Elastic Block Storage" (EBS) service, which is a critical aspect of EC2. EBS enables you to create virtual disks of any size and apply them to your instances. These virtual disks can be administered as logical entities separate from the computer that they are connected to. For instance, it is trivial to clone a virtual disk and connect it to a new server. Indeed, this service enables "Snapshots", in which all the EBS virtual disks associated with an instance can be duplicated, and a new instance launched from that snapshot.

EC2 Instances and their storage is administered through the EC2 Console, which allows you to launch new instances, clone instances, create virtual disks, assign IP addresses etc.

[13] Amazon Web Services. *Amazon EC2 Instance Types*, Retrieved May 14, 2012. http://aws.amazon.com/ec2/instance-types/

[14] Ars Technica. *$1,279-per-hour, 30,000-core cluster built on Amazon EC2 cloud*, Retrieved May 14, 2012. http://arstechnica.com/business/news/2011/09/30000-core-cluster-built-on-amazon-ec2-cloud.ars

More importantly, all these functions can also be completed using the EC2 API, enabling an application to automatically scale itself (add more servers depending on load), or for you to administer your cluster through your iPhone.

5.4 Availability & Reliability

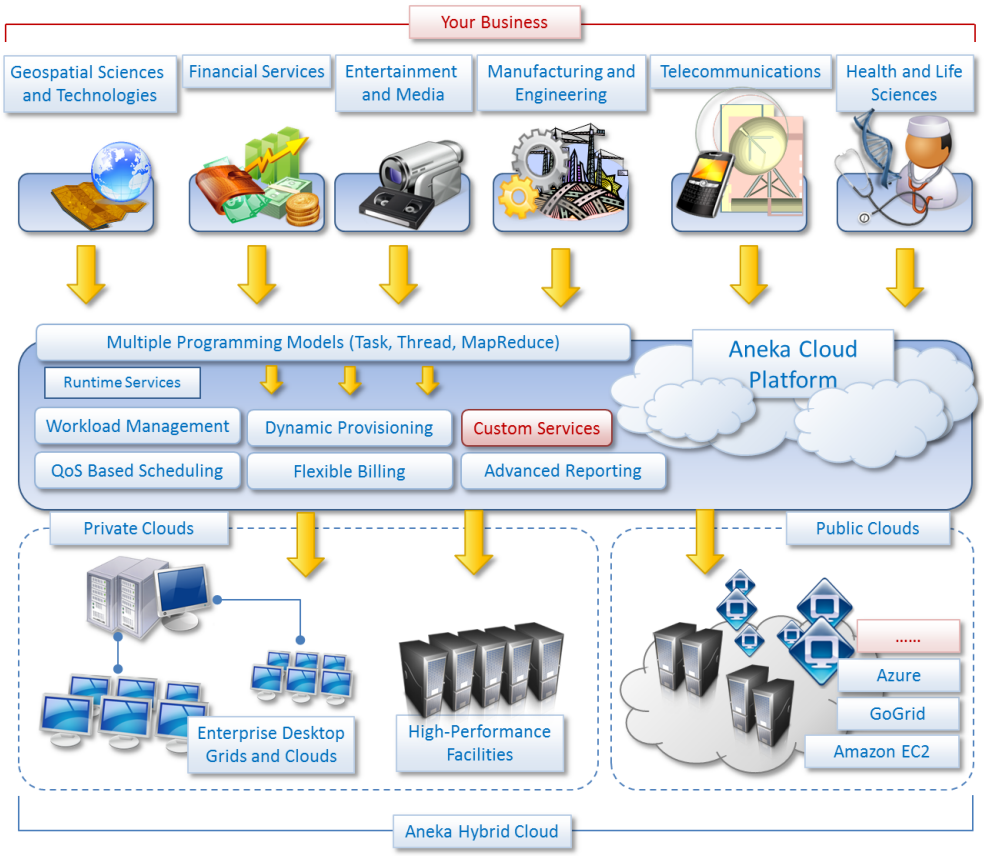
A key strength of EC2 is the end users ability to specify what data centre instances are launched in. This is done by specifying a region and availability zone. Each availability zone is guaranteed to be in a different datacentre, while each region is geographically separated. At present, EC2 is available in the following regions, US East Coast, US West Coast (Oregon), US West Coast (California), EU West (Ireland), Asia Pacific (Singapore), Asia Pacific (Tokyo), and South America (Sao Paulo). This enables applications to be developed with servers all around the world to reduce latency and improve user experience. It also means that a disaster in one region or with one data centre should not take down your entire cluster.

SIX

Aneka

6.1 Introduction

Aneka is a Platform-as-a-Service (PaaS) that enables the development of distributed applications for use across a broad range of cloud, grid and cluster platforms. Originally developed as part of the Gridbus Project, Aneka was later commercialised by Manjrasoft and is developed on top of .NET technology. It is available freely for a trial period[1]15, however additional arrangements must be made for commercial use. Aneka allows for applications to be developed in any language that is supported by the .NET runtime, such as VB.NET and C#, and through the use of Mono, allows applications to run on Linux and OSX as well as Windows infrastructures[4].



[15]Manjrasoft Pty Ltd. *Download Aneka Software and User Documents*, Retrieved May 16, 2012. http://www.manjrasoft.com/manjrasoft_downloads.html

6.2 Platform as a Service

PaaS solutions, such as Aneka, provide a platform from which distributed applications can be developed that scale on demand[1]. Aneka is a pure PaaS in that it does not provide an underlying hardware infrastructure. This differs from other PaaS solutions such as Google AppEngine and Microsoft Azure, which are both PaaS and Infrastructure-as-a-Service (IaaS) solutions. Instead, Aneka is designed to work as a platform layer on top of a variety of infrastructures, including combinations of multiple different options. This allows for the use of local resources when available, with the added ability for the provision of additional cloud resources from a service such as Amazon EC2 when the required job is unable to be completed on the local infrastructure within the required deadline. The other combined PaaS/IaaS solutions are also tied to their respective IaaS, giving Aneka the added benefit of being able to more freely swap between IaaS providers to meet changing resource demands.

6.3 Programming Models

One of the key strengths of Aneka is its flexibility, providing three different programming models (Task, Thread and MapReduce), which has enabled the use of Aneka in a broad range of applications in the fields of Engineering, Education and the Life Sciences.

- The Task Programming Model is best suited to programs that are "embarrassingly-parallel" as tasks are handled entire independently with no restriction on their execution order[3]17.
- The Thread Programming Model is best suited to porting existing threaded applications to a distributed computing environment, treating each thread as a task that is distributed across the available resources in the Aneka network[4]18.
- The MapReduce Programming Model is best suited to applications working on large datasets. Through the definition of *map* and *reduce* functions, Aneka will distribute the data across available resources, apply the map function and then reduce the output[5]19.

In addition to a broad range of existing functionality, Aneka has a layered and modular software architecture which enables additional programming models and services to be developed and plugged into Aneka easily.

[16]Manjrasoft Pty Ltd. *Aneka Cloud Computing Schema*, Retrieved May 16, 2012. http://www.manjrasoft.com/images/aneka_cloud_computing_schema.png

[17]Christian Vecchiola and Xingchen Chu. *Developing Task Model Applications*, Aneka Tutorial Series, Retrieved May 16, 2012. http://www.manjrasoft.com/download/TaskModel.pdf

[18]Christian Vecchiola and Xingchen Chu. *Developing Thread Model Applications*, Aneka Tutorial Series, Retrieved May 16, 2012. http://www.manjrasoft.com/download/ThreadModel.pdf

[19]Christian Vecchiola and Xingchen Chu. *Developing MapReduce.NET Applications*, Aneka Tutorial Series, Retrieved May 16, 2012. http://www.manjrasoft.com/download/MapReduceModel.pdf

Each programming model is based on the abstraction of job execution into 4 distinct parts: the Work Unit, the Manager, the Executor and the Scheduler. The Work Unit is dependent on the programming model being used and, for example, could be an entire Task or a single Thread. The Manager is the application that the user interacts with, to submit jobs to the Aneka system and to collect their results at the end. The Scheduler is what organises the communication inside the Aneka system. It schedules each work unit and then packages and dispatches the applications required along with any required input files. It is also responsible for collecting the results from each node and making it available to the Manager. The Executor is node specific and is tasked with running the specific application given to it by the Scheduler[4].

6.4 Deployment Model

Aneka was specially designed on top of the ECMA 335 specification, which is a Common Language Infrastructure (CLI), giving it great portability for use in heterogeneous network environments. This allows for the use of Aneka on existing physical hardware, even taking advantage of dead cycles on desktop machines, as well as allows for Aneka to be plugged transparently into external third party services such as Amazon EC2. Aneka provides a Platform Abstraction Layer (PAL) in order to help develop distributed programs that can take advantage of this CLI. The PAL provides a platform independent interface to a range of actions that would otherwise have to be platform specific, such as when it is required to interact with the hardware of the machine or accessing properties of the Operating System of the node being used.

Both programs that are specifically written to work with Aneka and existing applications are supported, though existing applications may be limited or work less effectively than if they had been modified. For existing applications, even provides a Parameter Sweeping services which will take a given application and a set of parameters and will run the application with all possible combinations of the given parameters with each combination handled by the system as an independent job and has resources allocated accordingly.

6.5 Quality of Service / Service Level Agreements

An important component of distributed computing is the ability to define requirements of Quality of Service (QoS) and Service Level Agreements (SLA)[1]. Aneka provides an easy to use management tool that can negotiate specified QoS and SLA requirements with the system, and can be used to bring in additional resources in order to meet these agreements. Aneka additionally has a counter-offer service, whereby if a job cannot be completed within the requested resource limitations, an offer will be made by the system stating the minimum resources it can complete the job it.

6.6 Additional Services

Aneka provides a number of additional features including handling of billing, dynamic scheduling, workload management and advanced reporting. Reporting allows users of

Aneka to closely monitor what is going on within the system, as well as providing billing functionality to charge each user of the system for their usage. The use of dynamic scheduling and workload management allows for the automatic scaling of the Aneka system as workload calls for it, providing flexible and automatic scaling to external cloud services such as Amazon EC2, Windows Azure and GoGrid Cloud Service.

SEVEN

System Comparisons

The following table compares each of the cloud computing services explored in the previous sections.

Properties	**Google App Engine**	**Microsoft Azure**	**Amazon EC2**	**Aneka**
Service Type	PaaS & IaaS	PaaS & IaaS	IaaS	PaaS
Supported Services	Deploy (Web Applications)	Deploy/Storage	Deploy/Storage	Deploy
Deployment	Web Applications	Azure Services	Customisable VM	Applications
Scaling	Automatic	Automatic	Manual	Manual
Abstraction of Parallelism	Full	Full	None	Some
Deploy on third party infrastructure?	No	No	No	Yes
Page Delivery Time[20] (seconds)	7.307	8.039	9.849	System Dependent

Table 7.1: Comparison of each of the cloud computing services

Each system covered offers different services to the user, and as such each has strengths and weaknesses based on what is required from the Cloud service.

Both Google App Engine and Windows Azure provide not only the infrastructure required to run scalable applications, but they additionally provide a platform for application development designed to best take advantage of the infrastructures they provide. This allows for very efficient use of resources, and for the easier development of applications, at the cost of inflexibility in the type of applications that can be deployed and tying applications to the infrastructure of the platform being used. Google App Engine and Microsoft Azure differ slightly in the type of applications they are best suited to. Google App Engine is directed toward the development and deployment of Web Applications, while Microsoft Azure is aimed to deploy Windows based applications into a scalable cloud environment.

Amazon EC2 differs from Google App Engine and Microsoft Azure in that it doesn't provide a Platform service on top of its infrastructure. Through the use of Virtual Machines (VMs), Amazon EC2 is remarkably flexible in what type of applications can be developed on it, at the cost of the inbuilt scalability and abstraction provided by the Platforms as part of Google App Engine and Microsoft Azure. It is therefore up to the

[20]CloudSleuth. *Cloud Provider Page Delivery Time Ranking – March*, Retrieved May 15, 2012. https://cloudsleuth.net/blog/cloud-provider-page-delivery-time-ranking-%E2%80%93-march

user to ensure that their applications can take best advantage of the infrastructure provided by Amazon EC2, making application development more difficult. A key difference in the infrastructure made available by Amazon EC2 is the ability to select the location of your instances, giving you flexibility to place your instances nearer to where your target audience is located. It is however, as a result of all this flexibility, more complex to configure and set up than other options, and poor inter-node performance limits can limit its usefulness in applications that are not simply parallelised.

Manjrasoft's Aneka aims to try and provide a compromise between the two by acting purely as a PaaS. It provides a level of abstraction which allows for easier development of scalable applications, while additionally being platform agnostic. This allows it to take advantage of wasted CPU cycles on a series of office desktops, or seamlessly scale to make use of infrastructure services such as Amazon EC2 when extra resources are required. It allows for the flexibility in both what infrastructure to use, as well as flexibility in the applications that can be built and developed.

The downside of Manjrasoft's Aneka is that your applications are still tied to the Platform as with the other PaaS solutions. As the platform service is not tied to a specific infrastructure, not as much can be done to get the most out of the infrastructure as can be done when the platform and infrastructure are tightly interwoven. Aneka is also less available than the other services covered. Amazon EC2, Google App Engine and Windows Azure can be signed up for through their respective websites, however Aneka requires (beyond its trial period) contacting Manjrasoft to determine future billing arrangements.

EIGHT

Success Stories

8.1 Google App Engine – GigaPan

GigaPan is a image exploration service which allows users to explore, share and comment on gigapixel panorama images. Formed in 2008 as a commercial spin-off of successful collaborative research between NASA and Carnegie Mellon University (CMU), GigaPan delivers a service which allows users to explore the extraordinary detail of over 50,000 high-resolution panoramas from around the world. These panoramas are created from thousands of images taken from a robotic camera mount, and stitched together using image stitching software[21].

In January 2009 GigaPan took a panorama shot of the Obama Inauguration speech. Shortly afterwards this huge panorama image went viral, resulting in 30 million views in just two days. This prompted the developers to try Google App Engine removing the 200 mb/s load from the CMU network where the site was originally hosted. This had fantastic results for the developers of GigaPan. The intense demand was removed from CMU's network, and they were still able to provide the gigapixel images latency free, which was one of their original concerns[22]. Since this trial GigaPan has moved its entire website to Google App Engine python platform and further developed tools which integrate with Google's other services such as Google Earth and Google Maps[23].

8.2 Microsoft Azure – Sharpcloud

Sharpcloud is a British company that was formed in 2009 on a simple but groundbreaking idea: to apply highly visual and commonly used social-networking tools to the crucial corporate project of developing long-term road maps and strategy.

The Sharpcloud service involves executives and other users working within a Web browser, creating a framework for their road map with a single click on the interface. They then define the various attributes and properties they want to track, such as benefit, cost,

[21] GigaPan. *About GigaPan*, Retrieved May 14, 2012. http://www.gigapan.com/cms/about-us

[22] Google Code Developer Conference. *Interview with GigaPan Developers Rich Gibson, Gabrielle O'Donnel*, May 2009. http://www.google.com/events/io/2009/sandbox/gigapan.html

[23] Google App Engine. *App Engine Developer Profiles*, Retrieved May 14, 2012. https://developers.google.com/appengine/casestudies#gigapan

and risk. The idea behind Sharpcloud is that "We only remember 10% of what we read versus 50% of what we see."[24] - driving their passion for creating visual software.

The developers envisioned the audience for their product being customers on a global scale, thus needing to ensure that their software was scalable and able to handle a high volume of throughput. Instead of using the limited funds that were available to purchase and maintain a server farm, a cloud computing solution was chosen instead. By utilising the Windows Azure platform, Sharpcloud was able to leverage the scalability and security measures that were available in order to market their product quickly and in a cost effective manner. Choosing to migrate to the cloud rather than maintaining own servers is now saving Sharpcloud U.S.$400,000 to $500,000 a year.

Sharpcloud developers were able to make use of the Windows Azure development fabric service which was installed on user machines. This allowed all development to be performed locally, using the cloud only for undertaking product testing. This saved the startup company a significant sum of money, as it was not paying usage fees for development time to be performed on the cloud.

Sarim Khan, the Chief Executive Officer and Co-Founder of Sharpcloud explains how the Windows Azure platform has aided his company. "We wouldn't exist if we had to build out of this level of server capability ourselves. Windows Azure makes it possible for us to scale the service as needed, using - and paying - only for what we need."[25]

8.3 Amazon EC2 – FourSquare

Foursquare is a location based service that enables people to share their current location with their friends. Founded in 2009, Foursquare was a pioneer in location based services and has grown to more than 20 million users, with over 5 million "check-ins" per day (when a user opens the Foursquare app on their iPhone / Android and clicks "check in", sharing their current location with their friends), and with more than 750,000 partner businesses[26].

Foursquare uses AWS to perform data analysis on hundreds of millions of events generated by its application servers. Foursquare utilises AWSs implementation of Hadoop Map-Reduce, "Amazon Elastic Map Reduce", which enables dynamic cluster resizing enabling foursquare to match the size of their cluster with the workload size. Flexibility is a key benefit of this arrangement, as it means that data scientists, analysts and engineers can spin up their own clusters to perform on-demand analysis without having to worry about capacity being available. This flexibility translates into massive cost savings for foursquare over hosting their own servers capable of meeting peak demand.

Since foursquare has a "base load" of activity that they require, they purchased $1 million worth of "reserved instances", which provides them with guaranteed access to a number of instances at a reduced cost which reduced their AWS bill by 35%.

[24]sharpcloud. *What it can do*, Retrieved May 15, 2012. http://www.sharpcloud.com/what-it-can-do.html

[25]Microsoft Case Studies. *sharpcloud*, Retrieved May 15, 2012. http://www.microsoft.com/casestudies/Windows-Azure/sharpcloud/Software-Startup-Triples-Productivity-Saves-500-000-with-Cloud-Computing-Solution/4000006685

[26]foursquare. *About foursquare*, Retrieved May 14, 2012. https://foursquare.com/about/

Mattew Rathbone, a software engineer explains the benefits of AWS over hosting servers internally: "By expanding our clusters with Reserved Instances and On-Demand Instances, plus the Amazon EC2 price reductions, we have reduced our analytics costs by over 50% when compared to hosting it ourselves. Additionally, we have decreased the processing time for urgent data-analysis, all without requiring additional application development or adding risk to our analytics."[27]

8.4 Aneka – GoFront

The GoFront Group was established in 1936 and is one of the 512 state run enterprises in China. GoFront is China's largest manufacturer of rail electric traction equipment. In addition to the production of locomotives, motor train sets and other urban transportation vehicles; GoFront also is the largest researcher into the development of rail related technologies. It has over 3000 science and technology staff of an overall employee base of over 10,000[28][29][4].

The development of new high speed electric locomotives and other transportaion vehicles is a large part of the research that occurs at GoFront. The prototypes for new designs are developed in AutoDesk, however part of the design process requires the rendering of these protoypes into high quality 3D images using the AutoDesk rendering software, Maya.

These images are used in the identification of potential design flaws in the prototypes. As such, images are required from many different camera angles and over 2000 frames from more than 5 different angles is required to generate the final 3D rendered image. Each frame is of such high quality, that it can take up to 2 minutes to render. This can lead to the complete set of images taking over 3 days to finish, a process which must be completed after every modification to the prototype is made.

Through the combined effort of Manjrasoft and the GoFront IT department, Aneka was deployed to take advantage of 20 desktop machines that were being under utilised. These machines were used for nothing other than the processing of word documents, and as such were turned into a private cluster with the use of the Aneka platform. The system was setup in a simple Master-Slave configuration, with 1 master computer distributing the frames to the 20 slave computers.

A custom GUI was developed in C# using the Aneka Task Model API, that enabled the disribution of the Maya tasks to the Aneka Grid and then collecting the rendered images. Through its use, the runtime of 3 days has been reduced to 3 hours, over 20 times faster. This was with the use of only 20 of 50 computers that were identified as suitable to being added to the Aneka Grid. Through the use of otherwise wasted cycles on existing hardware, the Aneka Platform was able to fully utilise the resources

[27] Amazon Web Services. *AWS Case Study: foursquare*, Retrieved May 14, 2012. http://aws.amazon.com/solutions/case-studies/foursquare/

[28] Manjrasoft. *GoFront Group (China)*, Retrieved May 16, 2012. http://www.manjrasoft.com/GoFront-Aneka-Use_Case.pdf

[29] Manjrasoft. *SOUTHERN CHINA RAILWAYS GOFRONT GROUP: Distributed 3D Rendering To Accelerate Prototyping*, Retrieved May 16, 2012. http://www.manjrasoft.com/flyers/Manjrasoft%20Solutions%20-%20Rendering%20-%20Case%20Study.pdf

available to GoFront and drastically improved the rending time of an important part of their product development process.

NINE

Summary and Conclusions

Existing services for development on the cloud provide a wide range of solutions for different needs. These allow developers to effectively and efficiently develop distributed applications on the cloud using mature development platforms to deploy on their own systems, globally networked high-performance infrastructure, or a combination of both. The integration and usage of these cloud based services has been successful for a number of commercial applications. These successes were documented for each of the examined systems in this report including Google App Engine, Microsoft Azure, Amazon EC2 and Manjrasoft Aneka.

None of the services covered are ideal in all circumstances, and the best system will differ based on the application that needs to be developed and deployed on the cloud. Google App Engine and Microsoft Azure offer specific Platforms for the quick development of applications best suited to their respective environments. Amazon EC2 provides the most flexibility to developers and organisations in terms of being cross-platform compatible, while Manjrasoft's Aneka aims to allow for smaller scale projects that involve a more straightforward development of an application on a cloud-based service.

BIBLIOGRAPHY

[1] Rajkumar Buyya, Suraj Pandey, and Christian Vecchiola. Cloudbus toolkit for market-oriented cloud computing. *CoRR*, abs/0910.1974, 2009.

[2] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 1, January 2008.

[3] Charlie Kaufman and Ramanathan Venkatapathy. Windows azure™ security overview. Windows Azure Whitepapers, August 2010.

[4] Christian Vecchiola, Xingchen Chu, and Rajkumar Buyya. Aneka: A software platform for .net-based cloud computing. 2009.