Cloud Computing

Barry Denby

Griffith College Dublin

February 9, 2022

Cloud Computing: What is this module about?

- Will cover the theory about how clouds are constructed and function
- Will also cover the practical details of how you go about using one
- And the various options available to you from a development perspective
- You'll get practical experience of coding for a cloud platform

How will the module be assessed

- 50% continuous assessment consisting of three progressively harder coding assignments
  - Note we will use Google App Engine for this
  - We will use the free limits available to everyone
  - i.e. if you sign up for a trial or provide a credit card to Google you are doing it wrong
- 50% exam consisting of 6 questions and you choose 5

Speaking of Assessment

- I will be experimenting with the following this time.
- You will get all of your CA in Week 2 but it is your responsibility to learn the necessary toolkits and produce the relevant cloud apps on time
- You will also need to show me live in labs what code brackets you have completed. I will only correct those code brackets that you've shown me. This is to encourage live feedback and prevent misinterpretation

What is Cloud Computing?

- Very difficult to determine as there are far to many definitions
- No single definition encompasses all aspects of the cloud
- Very difficult to pin down
- We will use the NIST definition as this tends to be the most accepted definition

NIST definition of Cloud Computing

- Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and releases with minimal management effort or service provider interaction.

What is Cloud Computing?

- Taken on its own the definition doesn't mean a lot until it is broken down
- Convenient: easy access to resources and very little setup cost
- On-demand: available and ready to use as required
- Shared pool: you do not have exclusive access to the resources and you will share with other applications and users

What is Cloud Computing?

- Configurable computing resources: you can determine and choose what resources you will use in your apps on the cloud
- Rapidly provisioned and released: if your application needs to scale up or down then your app has speedy access to extra resources and can release resources just as quickly
- Taking all of the above into consideration this is a version of utility computing
- Definition: Utility computing is the packaging of computing resources, such as computation, storage, and services, as a metered service

What is Cloud Computing?

- Thus cloud computing acts in a similar way to water/electricity/gas
  - there resources are there on demand
  - you can use as much or as little as you like
  - but you only pay for the resources that you use
- in cloud computing
  - you pay for CPU usage
  - you pay for I/O usage
  - you pay for storage
- The more you use the more you pay

What is Cloud Computing?

- In this course you will cover
  - the different models of cloud computing
  - what model is appropriate for your task
  - what options are available to you
  - concepts of data protection and user privacy in the public cloud
  - how the cloud model affects the way you write apps
  - how to write cloud apps

Cloud Computing

- This brings us up to the 2000's onwards. It is in the middle of this decade when Cloud Computing gains traction as a concept
- However, it took many advances to get this to work as cloud computing is not a single idea but an amalgamation of different fields of Computer Science
- Virtualisation is key to cloud computing
  - A machine with associated settings that can be migrated from one physical machine to the next
  - Should and virtual machine fail a new one can be started in its place
  - If more power is needed then the virtual machine can be duplicated and started elsewhere

Cloud Computing

- Distributed Systems
  - Multiple machines need to be able to coordinate
  - failures need to be detected and handled
  - Distributed System should look like a single unit to the user
  - i.e. distribution is hidden
  - Efficient workload and communication division
- Distributed algorithm design
  - Distribution of computation
  - Distribution of data
  - Synchronisation
  - Failure handling

Cloud Computing

- Taking these small advances together has led to cloud computing becoming popular in the mid 2000's
- Amazon was amongst the first to popularise it
  - They had a rather large problem with their web infrastructure
  - As the Amazon web store only hits peak demand twice a year (black friday/cyber monday and christmas)
  - Thus the rest of the year they have computing power lying idle, consuming power, and generating heat

Cloud Computing

- Thus they came up with EC2
  - This was run on top of all their web infrastructure
  - Amazon rewrote their stores to run on EC2
  - Which let them do two things
  - 1) they could scale up and down their store resources as necessary
  - 2) they could rent out spare capacity to others in non peak times

Cloud Computing

- Since this time many other companies have their own cloud platforms
- Google/Microsoft/IBM and others have their own offerings
- Even enterprises are converting their own IT infrastructure to private cloud environments for better use of resources
- Some companies even run their entire business on the cloud

Cloud Computation Model

- Every action your application takes will have a monetary cost
- The more you do/use = the more you pay
- Some cloud services will offer you a free limit, but beyond that you will have to pay
- If you hit your free limits then your application will be shut off until limits are reset

Cloud Computation Model

- Your applications must be parallelisable
- The trend in processing power has been to distribute work amongst multiple CPU's and cores
- Processor speeds have not increased as rapidly over the last 5 to 10 years
- The trend has been to pack in more cores to a CPU

Cloud Computation Model

- Your applications must be efficient
- Every unit of CPU, I/O, storage will cost you money
- The more efficient you are, the less resources you will use
- Thus you should try to use the most efficient algorithms and storage mechanisms if possible

Cloud Computation Model

- Your apps must be secure, as in a public cloud you share space with apps from other companies/vendors
- You will need to encrypt your data in cases where other apps spill over into your data areas
- You will also need to ensure that user's cannot see each other's data
- Data protection is extremely important in such a public environment

Cloud Computing Models

- In this lecture we will discuss the different models that are available for using the cloud
  - One of the first issues that you will have to decide on is which of these models you wish to use
  - That fundamentally determines how much control you will have over the service you wish to use/provide
  - Some of these models require very little work
  - While others give you a great degree of customisation

Cloud Computing Models

- The three main models that cloud computing exposes are the following:
  - Software as a Service (SaaS)
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)
- Everything else is a variant of these

Software as a Service (SaaS)

- You've used plenty of SaaS applications of this already
  - Gmail, Google Search, Google Calender
  - Dropbox, OneDrive
  - iCloud
  - etc

Software as a Service (SaaS)

- Definition of SaaS
  - SaaS is a model whereby the customer licenses applications and provisions them to users on demand. The services run on the provider's infrastructure and are accessed through a public network connection. Applications may be made available through the Internet as browser applications, or they may be downloaded and synchronized with user devices
- Above taken from: Cloud Computing Explained: Implementation Handbook for Enterprises

Software as a Service (SaaS)

- What this means?
- If you build a cloud application, someone who licences that off you and uses your infrastructure is using SaaS
- The application and data are stored in the cloud. However, the licensee does not own the application

Software as a Service (SaaS)

- What does it mean for the person or company that licences the application
  - They have little to no control over the application look and feel
  - They cannot fix any security bugs. The company that licences the application is entirely responsible for security
  - High possibility of vendor lock-in
    - May be difficult to extract your data from the service
    - and convert it into a form that can be used by another service
  - Very little work involved
    - one time setup effort required
    - vendor takes care of updating software, data backup, ensuring security

Software as a Service (SaaS)

- What does it mean for the person or company that licences the software
  - if you find the software that fits your needs perfectly
    - no work is required
    - however this model only really covers the common things that every person on the net will use
  - No coding or cloud knowledge is required

Platform as a Service (PaaS)

- Definition of platform as a service
  - The capability to deploy onto the cloud infrastructure, consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, or storage, but has control over the deployed applications and, possibly, application hosting environment configurations.
- Taken from Cloud Computing: Theory and Practice

Platform as a Service (PaaS)

- Examples of PaaS include
- Google App Engine (which we will be using in the practical section)
- Windows Azure: Microsoft's version of PaaS

Platform as a Service (PaaS)

- In this model the cloud vendor will provide infrastructure along with an automatic provisioning system for resources
- Will provide you a set of APIs to write your cloud applications with
- Restricts the languages you can use to develop your applications
- And in certain cases will restrict what you can use in each language

Platform as a Service (Paas)

- You will write an application to these APIs and test it
- If you app works locally you can then upload it to the provider's cloud where it becomes publicly accessible
- As a cloud developer I imagine this and the next model is where you will spend most of your development time

Google App Engine

- This is the PaaS model that will be used during the practical sessions of the course
- Applications can be coded in a number of languages
  - Java: which is what we used to use
  - Python: which is what we will use now
  - Go, PHP: other alternatives that we will not use

Google App Engine

- The PaaS model works by monitoring: CPU, Storage, I/O, Bandwidth, and API calls
- App Engine will give you free daily and hourly limits for these
- Should you hit any of these limits you app will be shut down until one of two things occur
  - you wait for the limits to be reset
  - you pay for additional resources
- Other PaaS providers work in a similar way

Platform as a Service

- With Platform as a Service you will be given a series of tools (list taken from Cloud Computing Explained)
- IDE: for development of applications. Can be a dedicated IDE or a plugin to an already existing one
- Integration Services: data marshalling, DB integration into apps, security, storage persistence, and state management
- Scalability Services: for concurrency management and failover
- Instrumentation: Track activity of app and users. Also gives idea where bottlenecks are in application (for optimisation purposes)

Platform as a Service

- Workflow facilities: Application design, development, testing, deployment, and hosting.
- User Interface support: building UI's for your applications so they are easy to use.
- Visualisation tools: visual indication as to how apps are performing and what way users are interacting with them
- Collaboration services: to support distributed development
- Source Code Services: version control, testing, auditing, etc

Why use PaaS?

- You get a full set of integrated tools that fit in with the software development lifecycle
- You focus on the application and not the infrastructure that it will run on
- There are similar downsides to SaaS but some are more serious
- Currently there is no standardisation on APIs. The functionality in one API may not be available in another API

Why use PaaS?

- Due to the lack of standardisation vendor lock in is likely despite which vendor you choose
- each vendor has their own combination of of API/languages/setup for their PaaS offering
- Data encryption and user privacy are now your problems
  - While a PaaS vendor will give you some assurance on data security between your app and other apps
  - it will be up to you to ensure that data trusted to you by users is properly stored and encrypted
  - also your responsibility that user's cannot see each other's data

Platform as a Service

- As you can see data security and storage are issues in cloud computing
- We will explore these in detail in later lectures
- While you are restricted to a single vendor you do get a lot of power, and the infrastructure is handled for you
- If you require yet more control and customisation then you will need Infrastructure as a service

Infrastructure as a Service

- The model that Amazon AWS is based on
- Only provides you with virtual machines and virtual resources
- It is up to you to setup these machines with OSes, servers, storage, and your application code
- The upside is that you get full customisation
- However, this requires a lot of effort to construct and function correctly

Infrastructure as a Service

- Definition of IaaS
  - Infrastructure as a Service (IaaS) is the simplest of the cloud offerings. It is an evolution of virtual private server offerings and merely provides a mechanism to take advantage of hardware and other physical resources without any capital investment of physical administrative requirements. The benefit of services at this level is that there are very few limitations on the consumer. There may be challenges including (or interfacing with) dedicated hardware
- Above taken from: Cloud Computing Explained: Implementation Handbook for Enterprises

Infrastructure as a Service

- While IaaS provides very little to you as a developer it gives you the most flexibility and power
- Also provides the best view for how a cloud functions
- You have full control over the software stack that is contained in your virtual machines
- If you need the most control you should use IaaS however, it requires the most effort to get working

Infrastructure as a Service

- Because IaaS deals with virtual instances, storage, I/O, bandwidth etc pricing models are much simpler
- You pay for virtual machines with a set configuration, higher spec means higher price
- In the case of EC2 they are rented by the hour
- Thus if you leave an instance sitting idle doing nothing you will run up a bill for this
- The only extras you need to pay for is for any additional bandwidth, storage, and services you require

Infrastructure as a Service

- Taking EC2 as an example we will show how an app is constructed using IaaS
- First a virtual machine is rented (either preconfigured or completely empty)
- All necessary software for the stack is installed (including OS if necessary)
- At this point it looks exactly like renting a normal web server but here is where differences start

Infrastructure as a Service

- If you start to experience more demand that exceeds your capacity you will need to find extra bandwidth and processing power
- difficult to do but requires a lot of work
- However, in the cloud extra resources can be elastically added on demand
  - if you get another virtual machine you can start up a copy of the one you already have
  - if it is bandwidth or storage it will be made available to all instances immediately

Infrastructure as a Service

- This is the core idea of the cloud
- Resources are available on demand as and when they are needed
- So if your app has peak demand a few times during the year you need only pay for those resources around those peak times
- Outside of peak demand you can give those resources back to the cloud

What does IaaS provide?

- Storage virtualisation: redirects all of your I/O requests from VMs to storage. you have no idea how storage is arranged
- Memory virtualisation: the machine your VM runs on may be only using a fraction of memory available on a server
- Server virtualisation
  - Each server is an abstraction of cloud hardware
  - You have zero idea as to what hardware it runs on
  - You rent these servers on an hourly basis

Infrastructure as a Service

- Virtual Machines are free to move around the cloud, the cloud provider can pause and relate your instance
- Generally done if a node needs to be updated, needs maintenance, or has a failure
- Should a Virtual Machine fail the provider will simply kill it and start a new one in its place.
- Makes the cloud good at failure handling

For next week

- There will be no labs this week but for next week you will need to get your environment set up
- You will need an editor that can handle Python and Flask syntax (I would recommend Atom)
- You will need an installation of Python 3.x
- you will also need the app engine SDK for Python
- refer to the Google App Engine by Example notes for how to set up the SDK