Enhancing the Delivery of Virtualised Labs in Academic Curricular

By Jonathan Ward

Birmingham City University – TEE/CTN

BSc (Hons) Computer Networks

Supervisor: Mr Rehan Bhana

May 2011

Abstract

Birmingham City University's (BCU) School of Computing, Telecommunications and Networks (CTN) teach a Microsoft Academy based theme in all ICT degrees based around Infrastructure Technologies. Currently lab sessions for these modules are delivered using desktop virtualisation on individual desktop PCs, typically one per student in a session. This is deemed an unsuccessful solution by students and staff, and the theme leader has asked for a more centralised system that is easier to manage, providing virtual machine deployment from one location rather than students having to suffer long start up times and heavy resource usage. Waterfall methodology is used to provide structure to this research and development of a solution. The labs provided use Microsoft Windows platforms, including Windows Server 2008, however an open-source cloud offering has been considered to keep costs down at the server end.

Acknowledgments

I would like to thank Mr Rehan Bhana for all his support throughout my time at university, especially within my final year. Thanks also to Chas Marwaha and Clive Davis for allowing the relevant changes to this project so late in the day.

Glossary

CC	Cluster Controller
CLC	Cloud Controller
CPU	Central Processing Unit
IaaS	Infrastructure as a Service
ICT	Information & Communication Technology
KVM	Kernel Virtual Machine
NC	Node Controller
OS	Operating System
RAM	Random Access Memory
PaaS	Platform as a Service
SaaS	Software as a Service
UEC	Ubuntu Enterprise Cloud
VM	Virtual Machine

Table of Figures

List of Tables

Contents

Enhancing the Delivery of Virtualised Labs in Academic Curricular

Aim

To propose a new or improved platform for teaching virtualised based curriculum at BCU.

Objectives

1. Identify a method of establishing change in an ICT Infrastructure
2. Model and capture existing platform of delivery
3. Research and analyse alternative platform and technologies
4. Propose and design a secure solution for the 'Infrastructure Technologies' theme at CTN/ BCU
5. Develop and test a prototype solution for delivery of virtualised labs
6. Evaluate the sustainability of the solution, and make recommendations

Problem

The existing method uses desktop virtualisation on individual "fat clients" to provide lab sessions to 20-30 students at a time, with the increase in demands with new infrastructure platforms such as Windows Server 2008 these virtual machines get larger over time and drain the individual host machine resources.

The course director wants a more centralised, manageable solution where virtual machines can be launched on the fly and even customised on a session by session basis.

Rationale

To produce a new solution from the existing virtualisation used for Infrastructure Technology lab sessions and exams within the university.

Virtualisation can be an inexpensive way of running multiple environments on an "as and when" basis. The primary benefits of this technology are server consolidation and higher utilisation, essentially creating savings for an organisation. This is achieved by running up to 15 virtual machines on one physical machine.

This solution will enable "on the fly" virtual machine deployment across the LAN to any client machines that require the images. This means pre-configured template images can be launched into fully-fledged machines and duplicated for that session only. This adds an ease to management from a lecturer's point of view.

Virtualisation typically means computation is done at the back-end i.e. where the VM is stored, this means cheaper front-end machines or "thin-clients" can be utilised; there is an increase in server demands due to this centralisation.

The private cloud model is being used for the management & deployment control, adding ease to the user performing deployment.

Security is one of the major concerns in business when cloud is discussed and this project looks at some of those concerns and how they can be combatted, some are simple techniques such as encryption.

Review of existing knowledge

Introduction

This section will identify the existing implementation for the virtual labs and will investigate alternate technologies that can be implemented to improve the current model. This investigation will form the basis of the new proposal.

Current situation

Currently each lab has 20 physical client machines with access to the core university infrastructure via a 100mb LAN link with CAT5 cabling. The current specifications are outlined in Table 1.

Table 1 Hardware and Software setup for current labs

Component		Model	Manufacturer	Description
Lab Machines				
	Operating System	Windows XP 86x	Microsoft	32bit edition
	Processor	Intel Core 2 Duo E8400 2.0GHZ	Intel	Dual Core CPU
	Motherboard	unknown	unknown	With on-board sound
	Graphics Card	NVIDIA Quadro FX580 PCI-EXPRESS 512 MB	NVIDIA Leadtek PCIe	512 MB dedicated memory
	Random Access Memory (RAM)	4GB RAM	HP	Two paired 2GB dims
	DVDRW drive	32x	Stone Computers	
	Network interface card	Broadcom Netlink ™ Gigabit Ethernet	unknown	1000mbps (100 used)
	Hard drive	320 GB HDD	Fujitsu	SATA II
Network Components				
	Switch	100MB	Cisco	Switch connecting machines in lab to university core infrastructure
	Network Cable	CAT5	unknown	
Software (Used on lab Machines)				
	Virtual PC 2007	32bit edition	Microsoft	Installed on 20 lab machines for student labs sessions

Virtual machines				
	• Windows 2003 server professional (86) • Windows 2008 server professional (86) • Windows XP professional (86) • Ubuntu Linux	All 32-bit	Microsoft, Canonical	Installed locally on the 20 machines in the lab; used for student lab sessions

Currently the clients store their own instances of the virtual machines required for all labs, these VMs are impersonal to the user and revert to a default start when shutdown. This is an issue in exam situations where VMs must be left on after the session, which takes up valuable lab time for others, in order to be assessed and marked; this also makes validation of marking near impossible.

The existing infrastructure is outlined in the Figure 1.

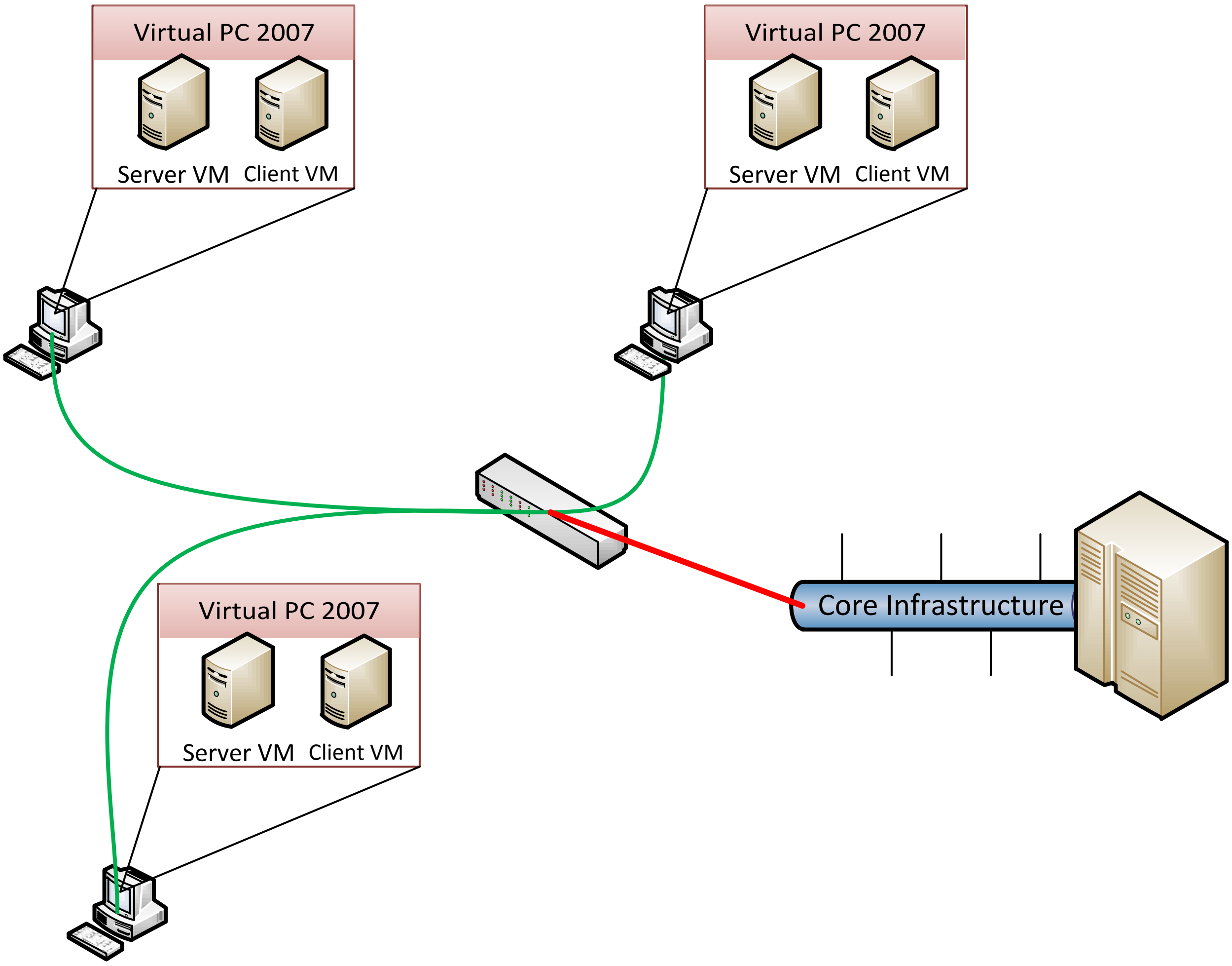


Figure 1 Existing Virtual Infrastructure

Cloud Computing

“Cloud computing is managed, shared applications, development platforms or computing infrastructure via a network” (McDonald, K (2010))

“Cloud” is a term coined in around 2004; however its principles have been around for many years, web based email services such as Hotmail are technically cloud computing as they are a resource found through a network, i.e. the Internet, other business applications such as salesforce.com have been around for a long time too (since 1994), in fact this is one of the first Software as a Service (SaaS) platforms, which is discussed later.

Microsoft’s recent “To the Cloud” advertising campaign shows how far cloud computing as a term has come, it is appealing to the home consumer and advertising features of the Windows 7 operating system that utilise elements of the Internet in a cloud like manner

Cloud Service Models

There are 3 core service models of cloud computing, and there are many examples of each. Many providers of cloud infrastructures coin their own models too, and Cisco’s 4th model as presented on 28th February 2011, in Amsterdam is Data Centre as a Service (DCaaS), this is where the whole data centre is the service, including all the cooling, power and floor space. The 3 primary models are:

- Software as a Service (SaaS) - Software that is utilised and accessed over the cloud, this can be a simple webmail system such as Hotmail or Gmail or any front end application that is part of your business processes such as a CRM or ERP application, that is hosted by a 3rd party organisation.
- Platform as a Service (PaaS) - A development environment for SaaS services for testing, development or use, examples include IBM Test and Development cloud and Amazon Web Services (AWS).
- Infrastructure as a Service (IaaS) - Provides the provisioned physical infrastructure for an organisation, this is where you rent physical servers and utilise them however you want, from wherever you want. This is a service typically offered by larger organisations and examples include Microsoft Azure Cloud and Amazon EC2/S3 infrastructures.

(Adapted from McDonald, K (2010) - *Above the Clouds - Managing Risk in the World of Cloud Computing*)

Cloud Delivery Models

There are 3 models of delivery for cloud services, these are:

Private - Private clouds (aka, on-premises cloud) are cloud deployments inside the organization's premises, managed internally without the benefits of the economy of scale but with advantages in terms of security**.**

Public - Public Clouds are the original concept of cloud computing based on the ubiquity of the internet. This type of cloud provides all the benefits of the economy of scale, ease of management, and ever growing elasticity. The major concern about this style of deployment is security.

Hybrid - Hybrid Clouds are a deployment type that sits between the private and the public clouds. Hybrid Clouds are usually a combination of private clouds and public clouds, usually, managed using the same administration and monitoring consoles.

(Adapted from "Cloud Computing, the new IT Paradigm" http://itechthoughts.wordpress.com/2010/02/23/cloud-computing-the-new-it-paradigm/)

Cloud Decision

Cloud computing is the "buzz word" of the entire IT industry and is something that should not be ignored, for the purpose of this project, and bearing in mind the customer is the university, I think a private cloud model should be used to deliver the virtual machines. Further research into this area has highlighted the open-source Ubuntu Enterprise Cloud (UEC) as a possible provisioning mechanism, and being open-source means we can keep costs down to a minimum.

UEC uses a platform known as Eucalyptus, which is what Amazon's EC2 service is built upon. The structure for this requires two server machines across two different networks; with client machines (the lab PCs) having a Linux based operating system that has virtualisation capabilities provided by the KVM module.

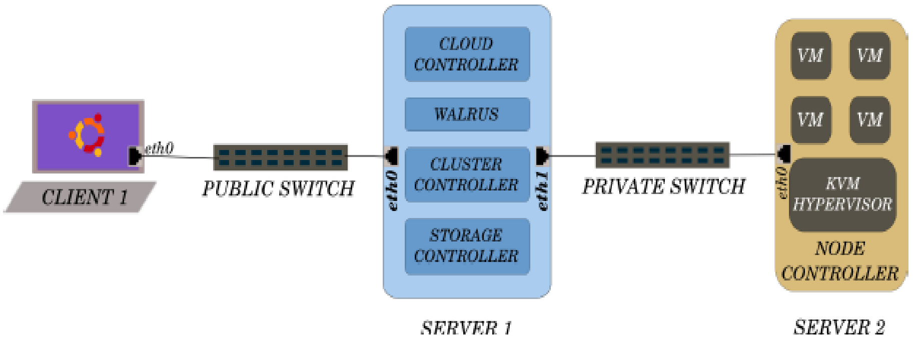


Figure 2 UEC Physical Topology

Virtualisation

Virtualisation lets you run multiple virtual machines on a single physical machine, with each virtual machine sharing the resources of that one physical computer across multiple environments. (Source: VMWare.com)

There are many levels of virtualisation, Bare-Metal and OS-Level are two examples; both have their advantages and disadvantages.

Bare-Metal virtualisation (also known as full virtualisation) is where a special kind of operating system (a hypervisor) is used instead of the traditional Windows or Linux platforms that we are virtualising, this hypervisor allows access to levels of the hardware that other methods do not allow, and as such this means that the virtual machines can be manipulated and adjusted as if they are on real hardware. This method is commonly employed at an enterprise level; my own experience comes with running VMware ESX 4 on 3 identical machines to allow for failover clustering and vMotion, which allows you to migrate live machines between physical machines, maintaining your uptime. VMware estimates a consolidation ration of 15:1 when deploying a virtual solution to replace existing physical servers; this has economic and green benefits.

OS-level virtualisation is where an application installed within a Windows or Linux operating system is used to build, manage and run virtual machines; examples of this include Virtual PC (which is used in the current implementation), VMware Workstation, and Oracle's VirtualBox. One other OS-level product that exists is KVM, the Kernel Virtual Machine, this is a Linux only platform and "Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc." (www.linux-kvm.org.) This is to be the chosen virtualisation platform for the solution, as it is what the Ubuntu Enterprise Cloud uses as its platform.

Recommendation

Based on the research above a decision has been made to produce an infrastructure with UEC as a private cloud which will be used to deploy virtual machines using KVM to the desktops, this means processing is done at the server (cloud) end and for the users there should be little impact on the local resources. The implementation must investigate how to save changes to virtual machines as they will be created on the fly as and when they are required.

There is a level of education required for the staff to manage the system once it is operational and discussions will be held throughout as to how this project can progress.

Storage is an issue, as virtual machine images can be very large, and with up to 20 instances of each at one time a lot of storage will be required. UEC can use iSCSI which is a storage controller mechanism that is cheaper to run than an enterprise level storage area network (SAN) and this will be further investigated.

I have produced the topology below to show the initial idea for the solution (Figure 3).

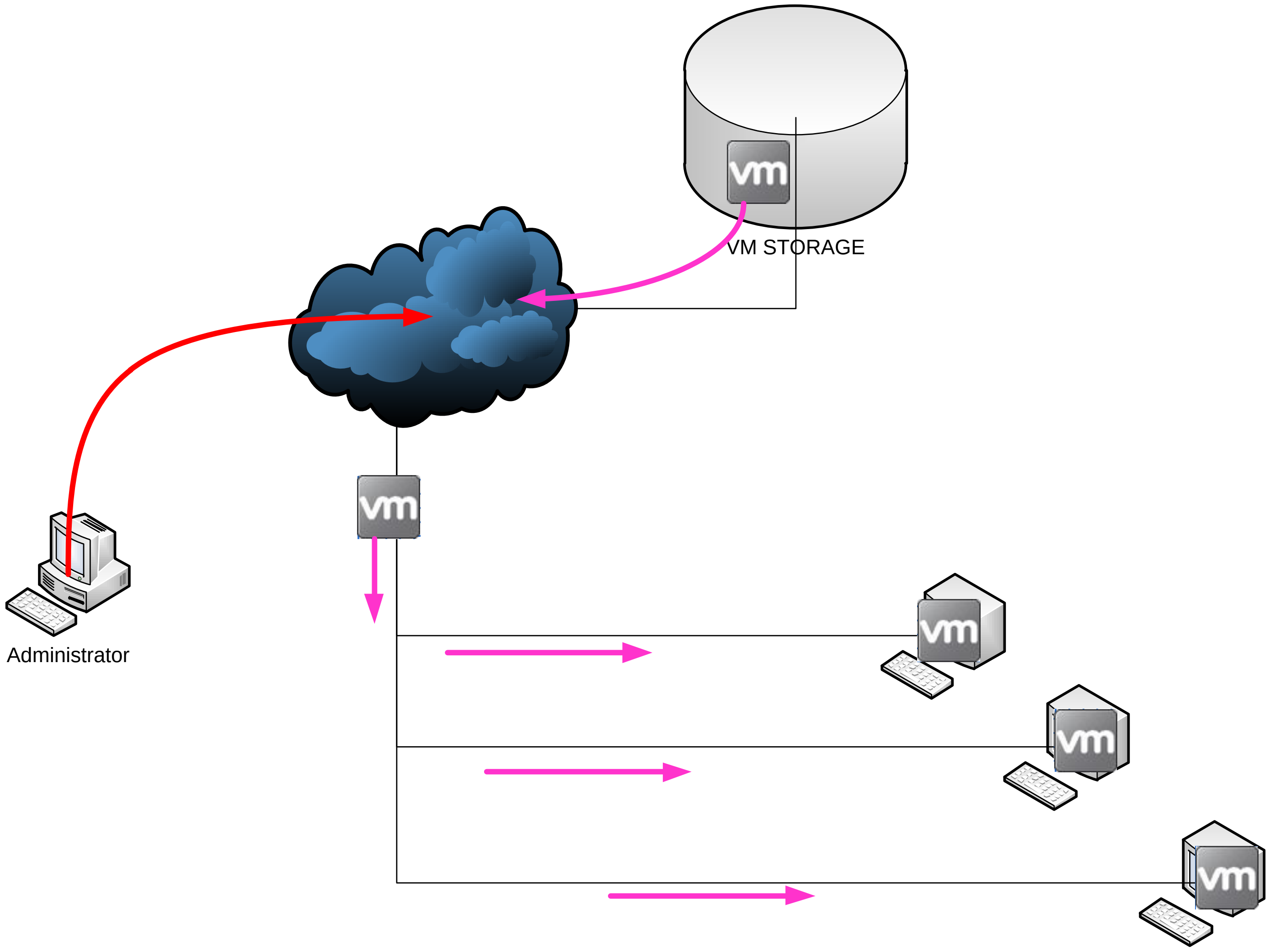


Figure 3 the Proposed Solution

Primary Research

I have conducted a questionnaire to gather the thoughts of the students who currently use the virtual labs. From this questionnaire my aim was to produce a requirements list for the final solution and to use this as a measure in my evaluation.

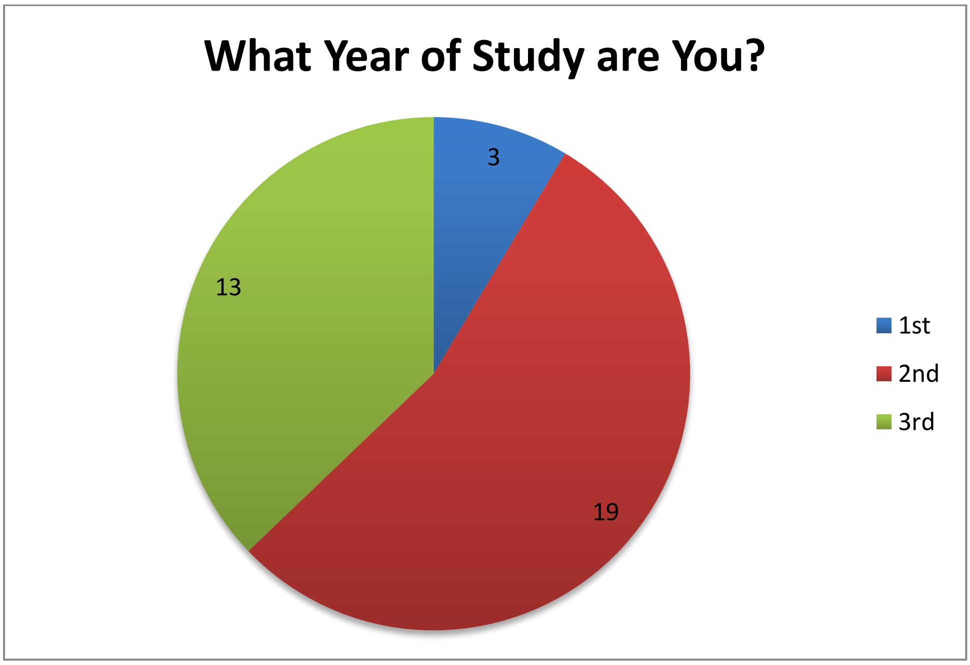
The questions asked were:

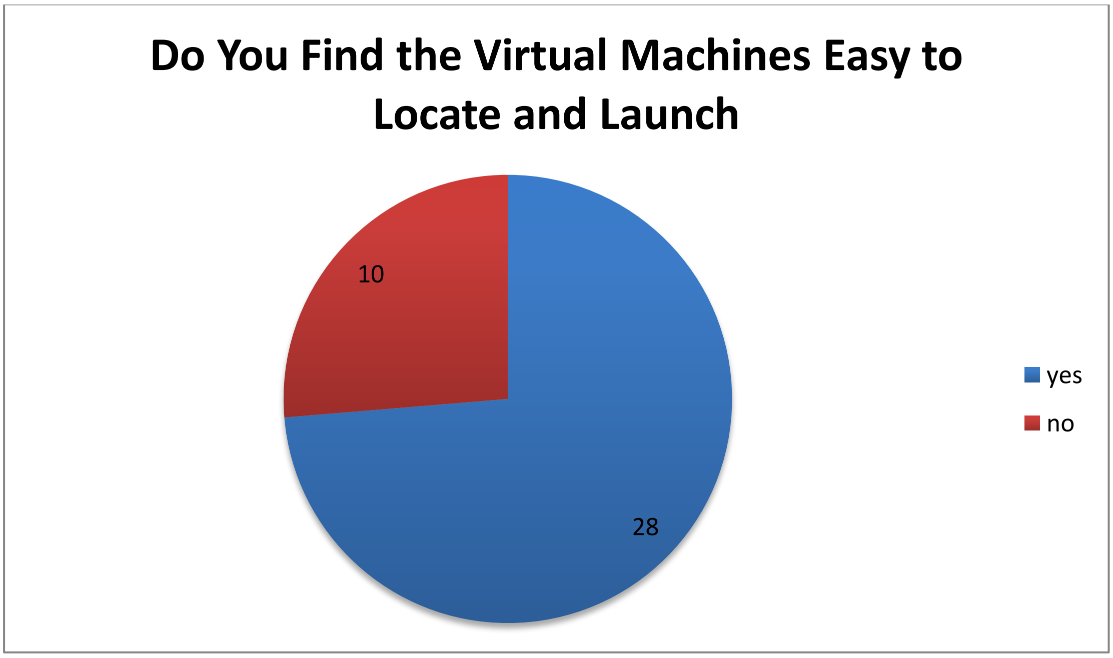
1. What year of study are you?
2. Do you like the labs being in a virtual environment?

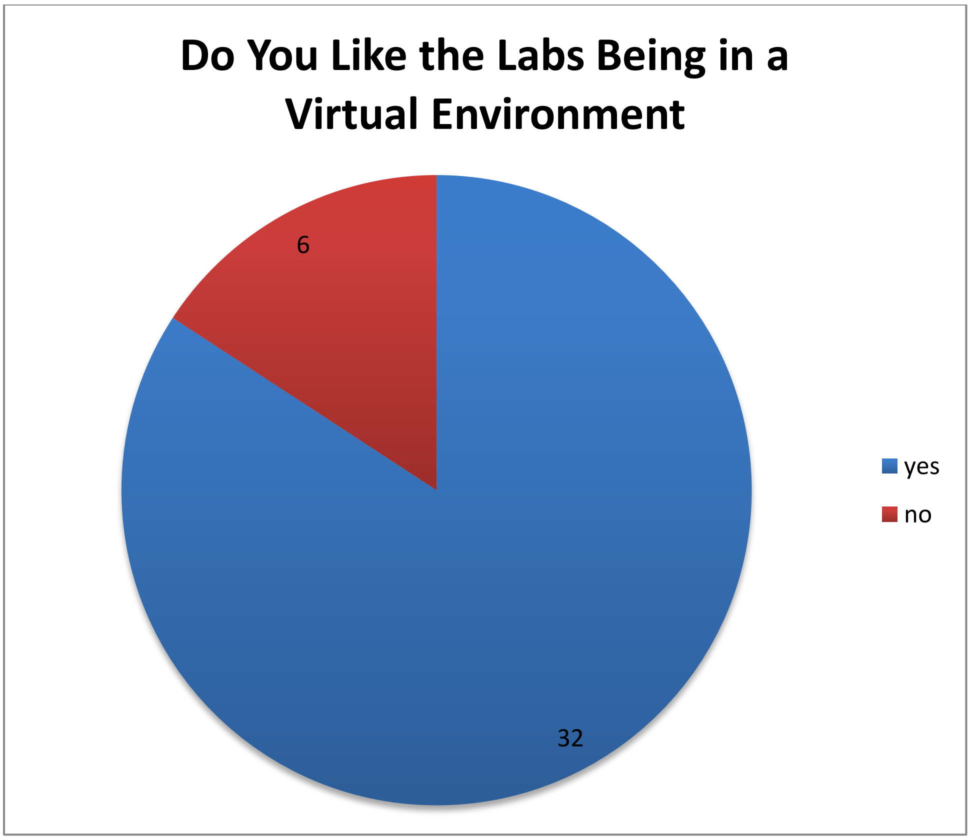
3. Do you find the virtual machines easy to locate and launch?
4. What concerns do you have with these virtual machines?
5. What would you consider an acceptable start-up time for each virtual machine?
6. Do you have any ideas on how the virtual labs can be improved from an infrastructure perspective (i.e. not the lab content but how the virtual machines operate)

I had 38 responses, with 20 completing the optional open ended question 6.

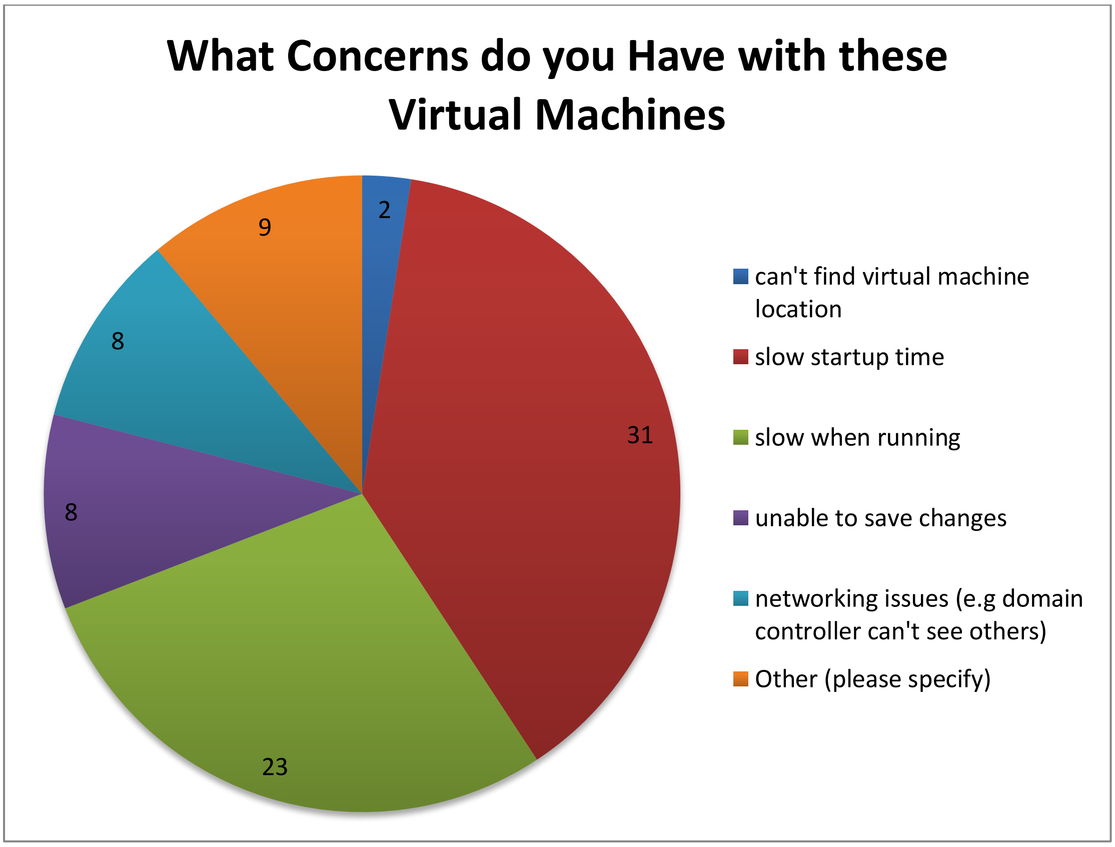
The results are shown below.

1. 

2. 



3.

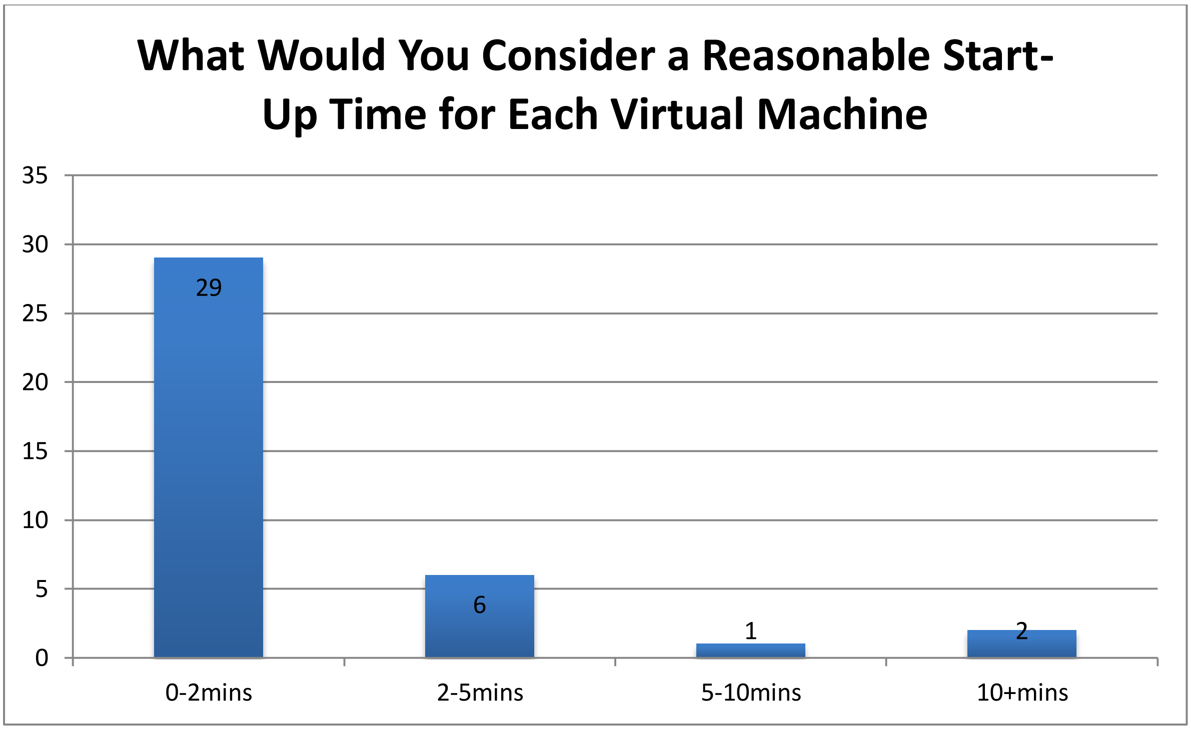


4.

The “other” responses were (9 responses):

- “Taking an exam, in 40 minutes on virtual machines that take at least 10 minutes to restart. You do the math”

- “Changes are still there from previous person. Shouldn't need to enter a new password every lab”
- “When starting step by step configuring the lab then suddenly stops and I don't know the issue and I have to restart the whole process again and waste my time and achieving nothing”
- “Can’t locate them in all of the rooms when you need to practise for exams”
- “Whilst I don’t find it hard to find the VMs, they could be more clearly signposted for beginners”
- “No comment”
- “Computers restarting”
- “No concerns”
- “No direction from Tutor, should at least be actively teaching/explaining the lab to the class. Not waiting until we don't understand”

5. 

6. Written Responses shown below, 20/38 participants answered, 2 answers omitted.
   - “If the machines were linked to the student disk it would be easy to save work, and maybe more powerful machines or better memory would help with start-up times”
   - “Placing it on a better infrastructure would solve most issues”
   - “Smoother operation when multitasking in the virtual machine environment”
   - “GET BETTER COMPUTERS! 2. You need at least 8 gig of RAM per each machine to run 2+ VM on it. 3. Just get a server rack, fill it up with server blades and use the computers as dumb terminals to connect to the server that will run all the virtual machine”
   - “Remove all passwords. Quicker PC's”
   - “They need to be faster”
   - “Perhaps finding a method to shrink the VHD's to reduce start-up time”

- “Reduce the start-up time”
- “Use VMware instead of virtual pc”
- “I don’t know how this setup. But should be able to reset once the work has done on it, and should be load balanced in speed, and performance that not one virtual work faster and others slower”
- “More instructions could be given and important topics covered in the labs”
- “Just generally faster, as when re booting during a test takes a lot of time, and you aren’t able to do anything until they reboot themselves”
- “More options to play with”
- “Please let the lecturers explain the lab aim and objectives first, and then do the lab”
- “The start-up should be faster”
- “The start-up time should be less than a minute because this allows the student to have more time to do the work”
- “Having separate machines to store the server OS's”
- “Allow us to save current progress on a working VM”

Primary Research Analysis

It is important to analyse the results gained above, in order to better understand what the end-users (students) want out of a system. Whilst the “client” for this project is the university staff, their clients are the students, and it’s their experience that is going to be enhanced. 56 responses suggest that the speed of the virtual machines is a primary concern to most users. This is one of the problems that the cloud computing model will attempt to address, as the computing is done within the cloud not on the client machine. Analysis of the written responses suggest that the students have a decent grasp of the technologies involved with provisioning the current virtual environment, and they are aware of the power required to correctly implement the labs in this way. Some responses have concerns regarding the setup if the actual virtual machines, including a couple of references to passwords used to log in to them. This highlights the fact that even if the provisioning is improved, there is a human element involved with ensuring the setup is satisfactory, and perhaps the university staff could use this questionnaire as a basis for them improving their “service”.

Methodology of change management

The need for change in the organisation has forced a more strategic approach, and below are the considerations for which approach to use.

Within an ICT infrastructure there are many policies and frameworks that can be followed to successfully design, manage, deploy and maintain that infrastructure. The way of carrying out these is defined as a methodology

ITIL (IT Infrastructure Library) is an ICT management framework made of many parts and is a project maintained by the OGC (office of government compliance). One of the major parts of ITIL is change management as it is a crucial part of IT Service Management (ITSM).

ITIL can be modelled as shown in Table 2.

Phase	Discipline	Description
Setting Objectives	Service level management	Identify, negotiate, and agree to services to be provided, quality measurement and IT performance targets to be provided to users.
Planning	Application & System Design	Plan and design IT infrastructure to meet Service levels committed to user.
	Capacity Planning	Plan for systems growth requirements
	Configuration Management	Create and maintain systems configuration information
	Asset management	Create and maintain asset inventory; track and monitor such use of assets
Execution	Incident Management	Detect, record, resolve problems
	Backup and recovery	Design alternative systems and resources to immediately restore IT services when problems occur.
Measurement	Performance Management	Monitor system performance data; tune system for optimal achievement of service levels committed to users.
Control	Change Management	Control all changes to the system to ensure that change does not degrade system performance
	Security Management	Control and administer access to the system to minimize threats to system integrity
	Availability Management	Monitor and control system resources and IT operation to maintain system availability
	Problem Management	Monitor and control system Known Errors and proactively remove them from the environment
	Financial Management	Monitor and control system IT expenditures

Table 2 ITIL Version 2

The purpose of change management is for:

- Minimal disruption of services
- Reduction in back-out activities
- Economic utilisation of resources involved in the change

This means that any changes made must have little impact on the business processes but benefit the organisation in the long run.

When developing a system such as the one for this project, there are 2 methodologies commonly used, these are:

- Waterfall Methodology
- Spiral Methodology

Waterfall Methodology

Waterfall Methodology was developed by Winston Royce in his 1970 paper the term waterfall was adopted later.

The waterfall name refers to its sequential flow in a top-down fashion.

The stages of waterfall are illustrated in Figure 4, below.

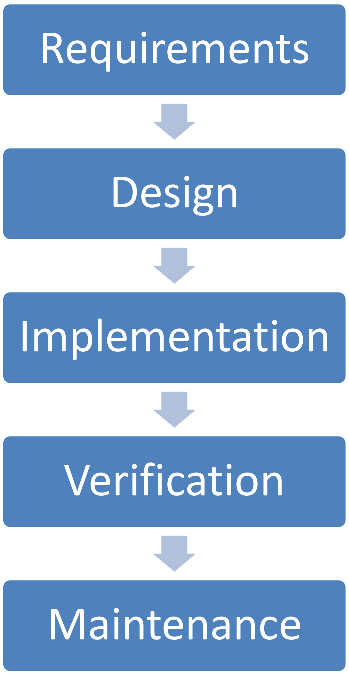


Figure 4 Waterfall Methodology

The stages are explained in Table 3, below.

Table 3 Waterfall Stage Explanation

Stage	Description
Requirements	This stage explains why the project was created and the reason for requirement of the project by identifying the problem and a possible solution.
Design	In this stage the requirements identified in the previous stage are used to create a design plan that meets the identified requirements.
Implementation (prototype)	At this stage the created design from the previous stage is created (implemented). The implementation stage is usually the most time consuming.
Verification	In this stage the implemented project is verified by identifying whether the implementation was successful usually by testing if it was unsuccessful the project goes back to the design or implementation stage.
Maintenance	At this stage the system has been successful verified and is maintained and monitored to ensure that it works correctly without issues.

Spiral Methodology

Spiral methodology is a slightly later development at 1986, and was developed by Barry Boehm in his paper "A Spiral Model of Software Development and Enhancement". This method is an extension of waterfall as it introduces a prototyping stage, allowing the stages to spiral. This means the waterfall stages can be repeated until the desired result is achieved. Spiral methodology is used for more complex enterprise level projects and is traditionally more time consuming than the waterfall model.

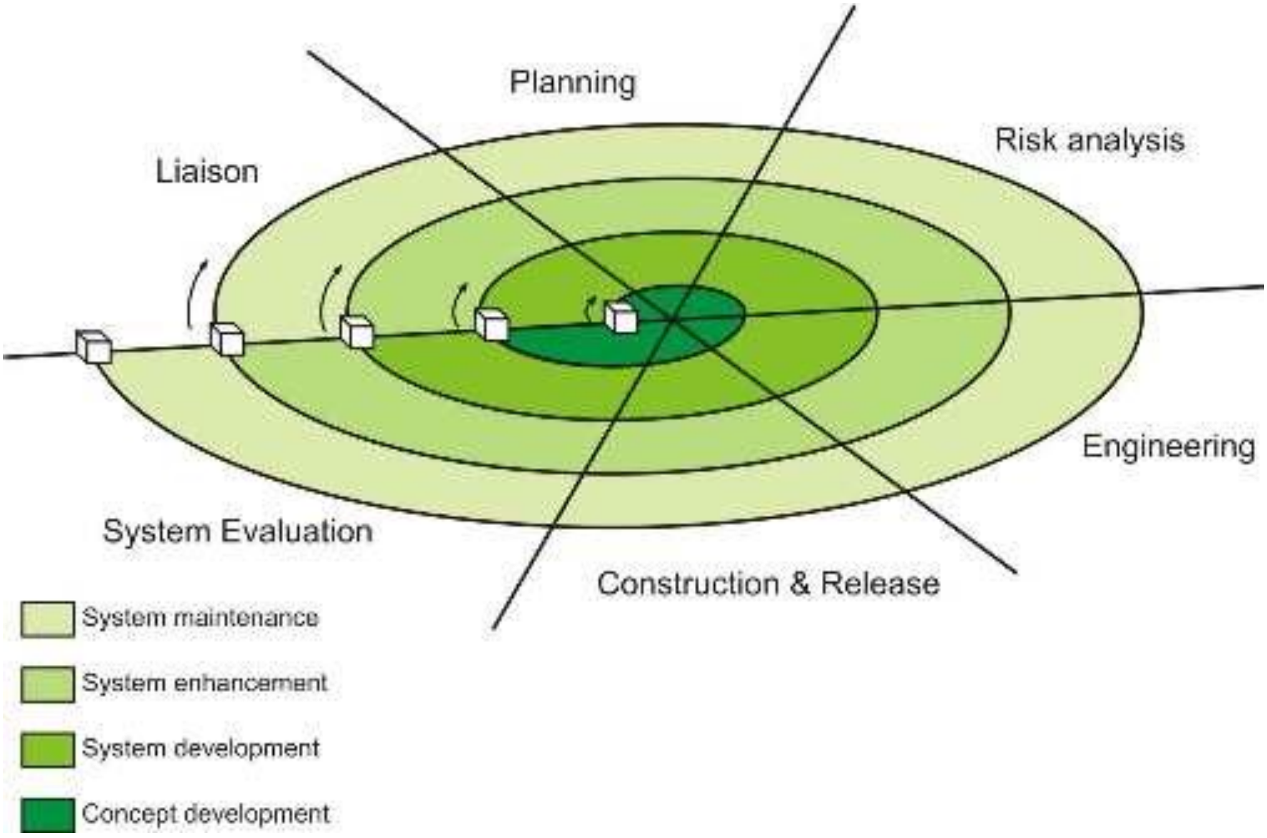


Figure 5 Spiral Methodology

Other Methodologies

Software Development Lifecycle

The software development lifecycle (SDLC) is a circular methodology commonly used by software developers in large and small development projects. It starts with the initial idea, but before the design is done, as in waterfall, a feasibility study is performed which tests how possible the solution is before any design time is wasted. Analysis of the requirements and systems is essential for a specification to be written before any design is performed. The SDLC includes a test and review phase which can in itself be a smaller circle within the main loop.

This methodology is relevant as although the name suggests a specific link to software development, there is a license to adapt this to a system development lifecycle. System could be defined as any technical based business process, in this case the development of a virtualised lab solution, utilising hardware and software.

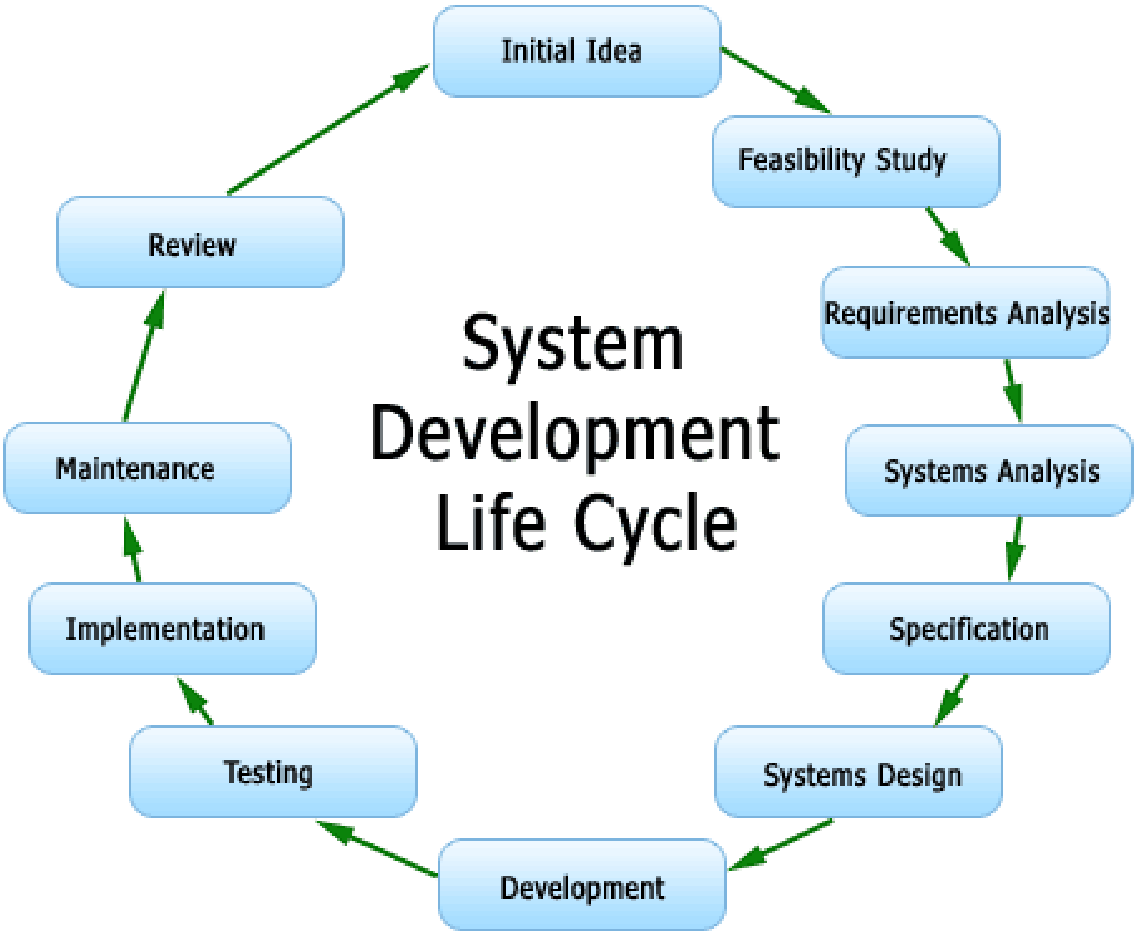


Figure 6 Software Development Life Cycle

Rapid Application Development (RAD)

This is a form of spiral methodology which includes "rapid prototyping" and is focussed on the development of applications rather than systems. Rapid prototyping is where at the inner spiral (concept development stage) lots of small working models are used to demonstrate features of the end product, but they do not necessarily work with each other, it's within the System Development stage that these independent modules begin to form the overall solution.

Decision

This project will utilise the waterfall model, but there will be an inclusion of a prototype, due to time constraints this prototype will more than likely act as the final solution with perhaps a few minor tweaks after the evaluation phase. In an ideal situation the spiral method would better suit this project if it was on a larger scale with more scope; this is because of its nature where you keep advancing on each version.

Private Cloud Implementation

For monetary reasons I have opted to build the UEC cloud within a virtual environment using Oracle's VirtualBox, open-source solution, this is also due to a lack of physical networking equipment. The primary concern with this method of implementation is that VirtualBox on my host PC cannot do virtual acceleration from within a virtual machine as it can't provide Intel-VT or AMD-V technology within virtual hardware. This just means that the virtual machines that are being deployed will be running slightly slower than if the cloud and node controllers were on physical hardware. Table 4 shows the host machine specification.

Table 4 Host PC Specification

Component	Specification/ Description
Processor (CPU)	AMD Athlon 64 X2 5200+ (2.9Ghz Dual-Core)
Memory (RAM)	4Gb DDR2-5300
Disk (HDD)	500Gb SATA II
Operating System	Windows 7 Ultimate 64-bit SP1
Virtualisation Platform	Oracle VirtualBox 4.0.4

Table 5 outlines the setup for the 2 virtual machines I have built for the cloud controller and node controller. It is possible to use multiple node controllers, however for this example there is only one.

Table 5 UEC Server Details

	Cloud Controller	Node Controller	Client Machine
	Username: uecadmin Password: password		
Functionality	CLC, CC, Walrus, SC	NC	Web Client
NICs	Eth0 (core network) Eth1 (cloud network)	Eth0(Br0) (cloud network)	Eth0 (core network) Eth1 (cloud network)
IP Addresses	Eth0 - 192.168.0.121 Eth1 – 192.168.20.1	Eth0 – 192.168.20.2	Eth0 – DHCP Eth1 – DHCP
Hostname	Server1.jw.com	Server2.jw.com	Client1.jw.com
Gateway	192.168.0.1	192.168.20.1	192.168.0.1

Cloud Controller (CLC)

The CLC was built using Ubuntu 10.10 (Maverick Meerkat) Server edition (64-bit), the CD (ISO Image) includes the cloud installation from the boot menu, and is extremely self-explanatory. The step-by-step process allows for network configurations as per the table above, as well as the usual system setup parameters such as time zone and keyboard language.

The cloud controller acts as the gateway for all nodes and images, and contains the web-based configuration services. Typically the web-based platform is accessed by just typing in the ip address in a browser. An enhancement of this is the HybridFox extension for the Firefox web browser. This allows you to manipulate images and instances once they exist on the system. This extension is built on the ElasticFox add-on used for the Amazon EC2 cloud, which UEC is based on.

Node Controller (NC)

This was built with the same ISO image as the cloud controller, and its primary purpose is to manage the lifecycle of the images from creation, through running until destruction. The node controller acts as a data source, reporting information on images to the cloud controller, and therefore the web based interface and 3rd party tools such as HybridFox. One cloud can run many node controllers, in this project there will be only one for simplicity.

Other Controllers

The Cluster Controller (CC) manages nodes and deploys images to them, this is effectively a background service installed on the cloud controller, and managed through the cloud controller service. The Walrus service is a storage service utilising open API standards such as SOAP and REST that manages the Storage Controller. In a large scale implementation the storage controller would be managing multiple iSCSI interfaces. ISCSI is a technology/ interface commonly used in storage solutions such as Storage Area Networks (SAN) within a company.

UEC Image Management

Managing images is the key element of this whole project; the aim is to deploy Windows images via the KVM virtualisation technology through a UEC cloud to the end user.

Documentation for developing Windows images is incomplete and appears to come with flaws such as needing extra drivers/ system files for the operating system, meaning they could have a performance hit when running, especially with Windows 2008. Windows 2008 doesn't use the traditional NTFS disk file system, used by previous versions of Windows. The new file system uses the GPT (GUID Partition Table) system, which is a different partitioning mechanism from the previous MBR (Master Boot Record). This is a problem with KVM deployment, which will support MBR out of the box, but there are hacks required for the GPT. 64-bit installs of Windows 2008 (i.e. the R2 version cannot boot from MBR disks (but they can read them), whereas 32-bit installs can.

I am following the windows deployment guide in the Eucalyptus Beginners Guide eBook, this provides a number of hacks, and with extra research I aim to get a working image running.

Server Installation

During the installation of the cloud and node controllers I have produced some screenshots to show various stages for explanation.

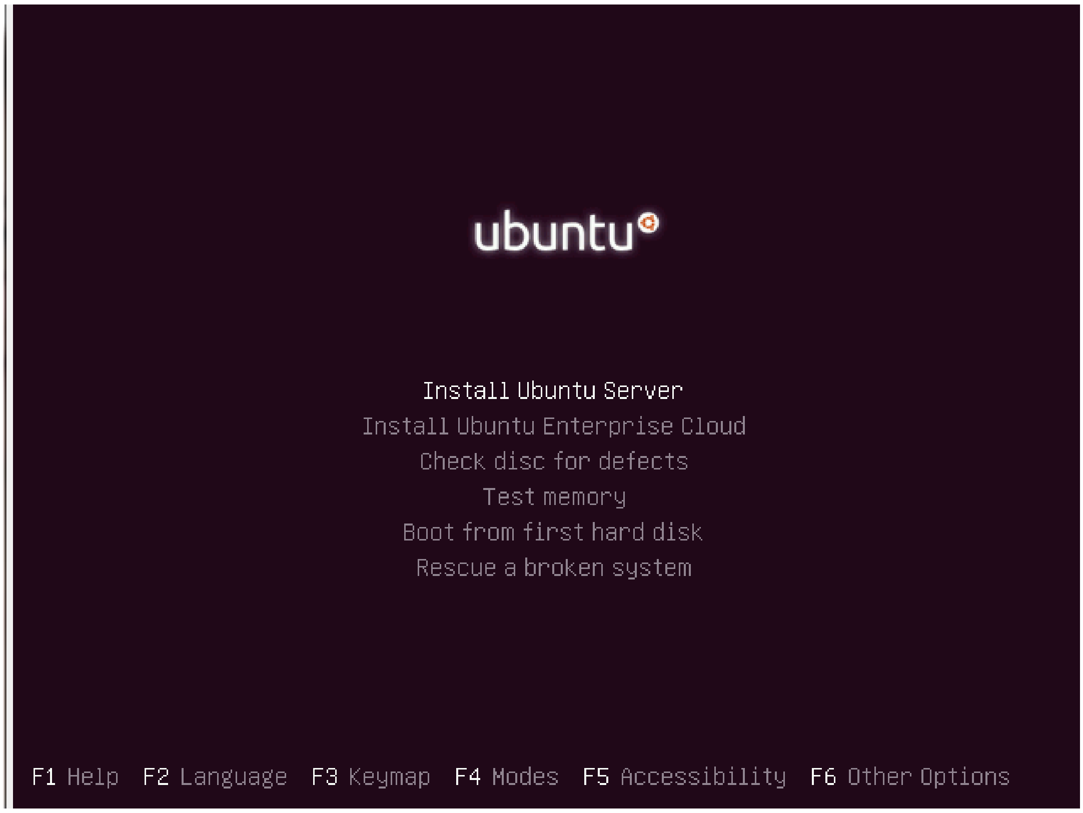
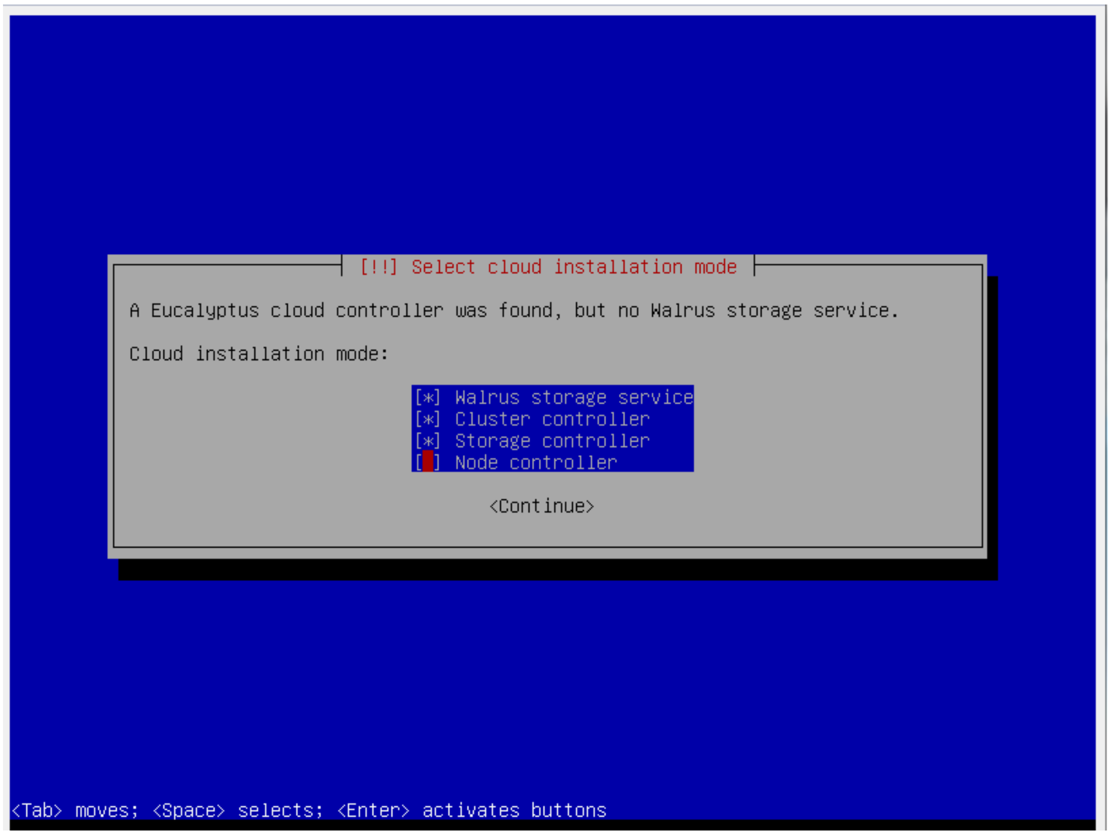

ubuntu
Install Ubuntu Server
Install Ubuntu Enterprise Cloud
Check disc for defects
Test memory
Boot from first hard disk
Rescue a broken system
F1 Help F2 Language F3 Keymap F4 Modes F5 Accessibility F6 Other Options

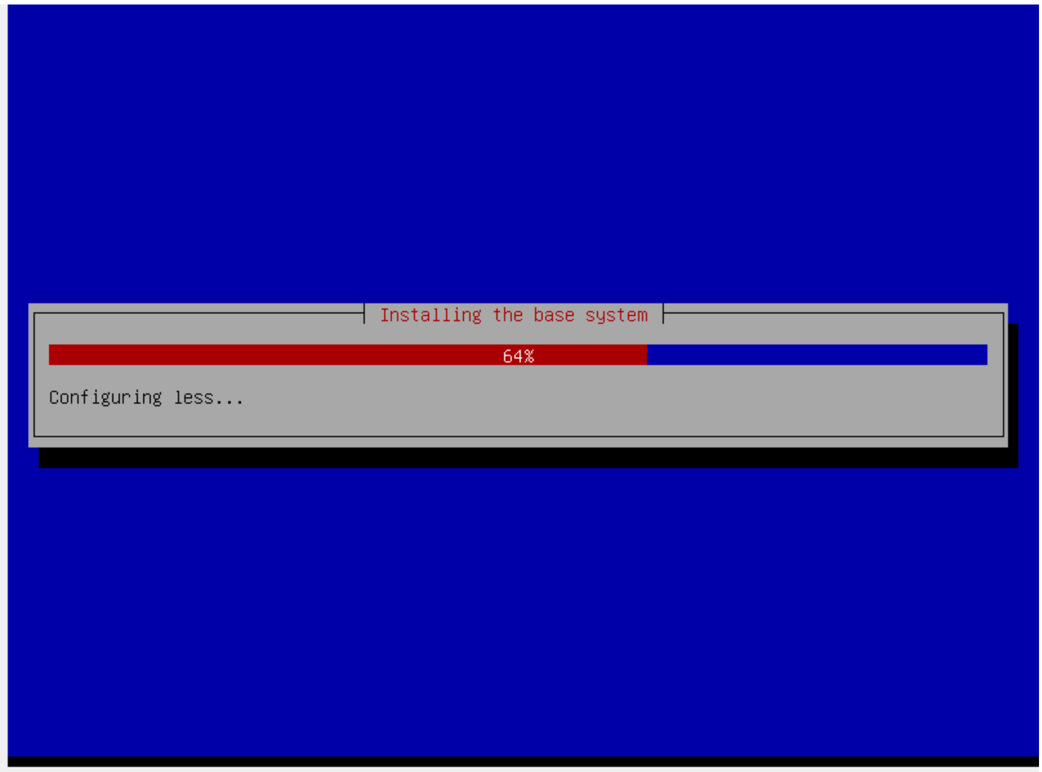
Figure 7 Ubuntu 10.10 Boot Screen

From the Ubuntu boot screen, as shown in Figure 7, you can select to "Install Ubuntu Enterprise Cloud" this allows the installation of the different servers such as the CLC and Node Controllers. Figure 8 shows the options for installing the different services, in this case the CLC VM is being configured with Walrus, CC and SC.



```
[!!] Select cloud installation mode

A Eucalyptus cloud controller was found, but no Walrus storage service.

Cloud installation mode:

                    [*] Walrus storage service
                    [*] Cluster controller
                    [*] Storage controller
                    [ ] Node controller

                          <Continue>

<Tab> moves; <Space> selects; <Enter> activates buttons
```

Figure 8 Configuring CLC options

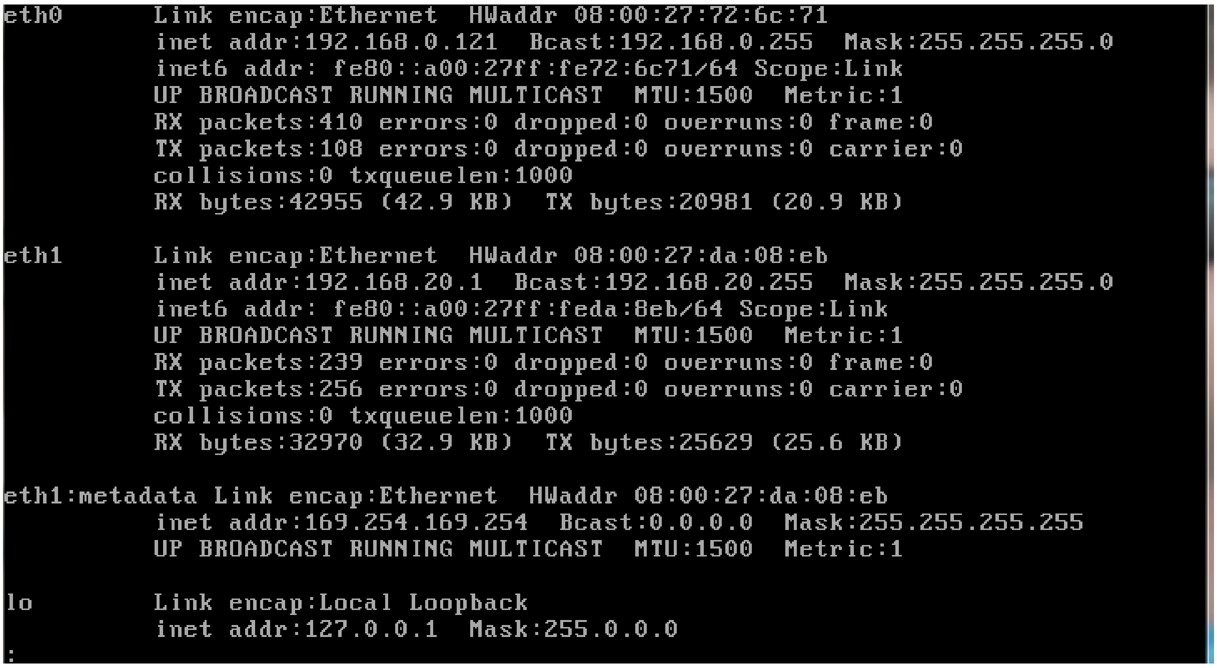


```
Installing the base system

64%

Configuring less...
```

Figure 9 Installation of the CLC

Figure 9 shows the installation of the operating system, after this stage some automatic configuration is done based on the options chosen for the controller.

The network for both virtual machines was configured as per Table 5, and the ifconfig network information output from the CLC is shown in Figure 10. This shows the ip addresses that represent the two networks in the system.



```
eth0      Link encap:Ethernet  HWaddr 08:00:27:72:6c:71
          inet addr:192.168.0.121  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe72:6c71/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:410 errors:0 dropped:0 overruns:0 frame:0
          TX packets:108 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:42955 (42.9 KB)  TX bytes:20981 (20.9 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:da:08:eb
          inet addr:192.168.20.1  Bcast:192.168.20.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feda:8eb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:239 errors:0 dropped:0 overruns:0 frame:0
          TX packets:256 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:32970 (32.9 KB)  TX bytes:25629 (25.6 KB)

eth1:metadata Link encap:Ethernet  HWaddr 08:00:27:da:08:eb
          inet addr:169.254.169.254  Bcast:0.0.0.0  Mask:255.255.255.255
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
:
```

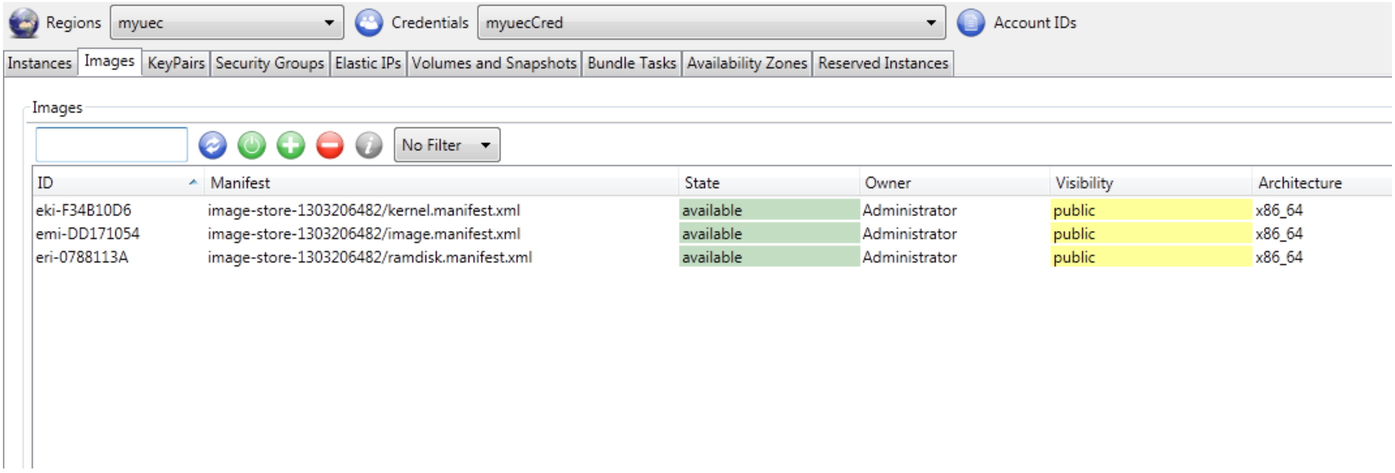
Figure 10 ifconfig output from CLC

The options for installing a node controller are virtually the same, except for when you have to input the address of the cluster controller, which in this case is the eth1 address of the CLC (192.168.20.1) which is the internal cloud network. The node controller is only configured with one physical network interface as it's used purely to manage instances and is accessed via the cloud controller.

HybridFox Plugin & Web Interface

As already explained the HybridFox web browser plugin gives a GUI management console that performs the command line tasks associated with manipulation of images in UEC. The command line toolkit that comes with UEC is called euca2ools and HybridFox emulates its capabilities.

Figure 11 illustrates what the plugin window looks like when there are instances available. In this case the image downloaded in Figure 14 is shown as being available, i.e. it is not running yet but can be.



Regions: myuec

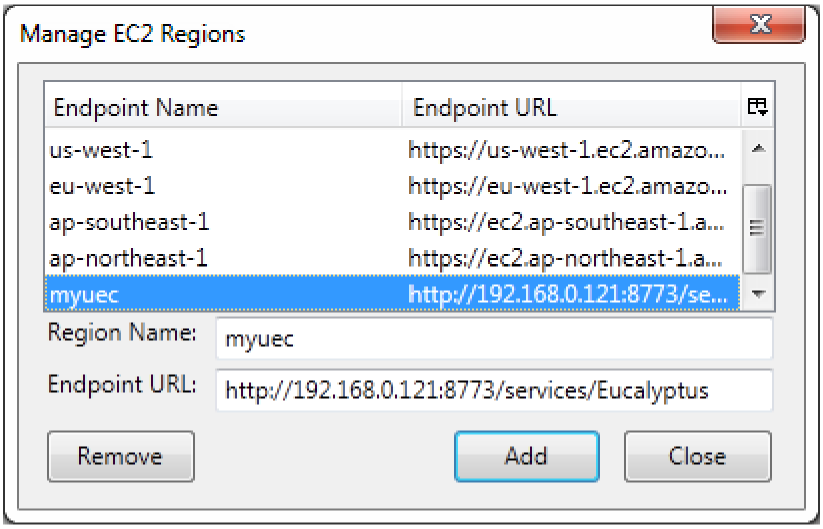
Credentials: myuecCred

Images

ID	Manifest	State	Owner	Visibility	Architecture
eki-F34B10D6	image-store-1303206482/kernel.manifest.xml	available	Administrator	public	x86_64
emi-DD171054	image-store-1303206482/image.manifest.xml	available	Administrator	public	x86_64
eri-0788113A	image-store-1303206482/ramdisk.manifest.xml	available	Administrator	public	x86_64

Figure 11 HybridFox example of Instances available

Figure 12 shows the "region" configuration for my UEC installation. Regions are the individual cloud clusters that are configured for your environment. HybridFox includes the amazon EC2 clouds as default, as per its fork from the ElasticFox add-on for EC2.



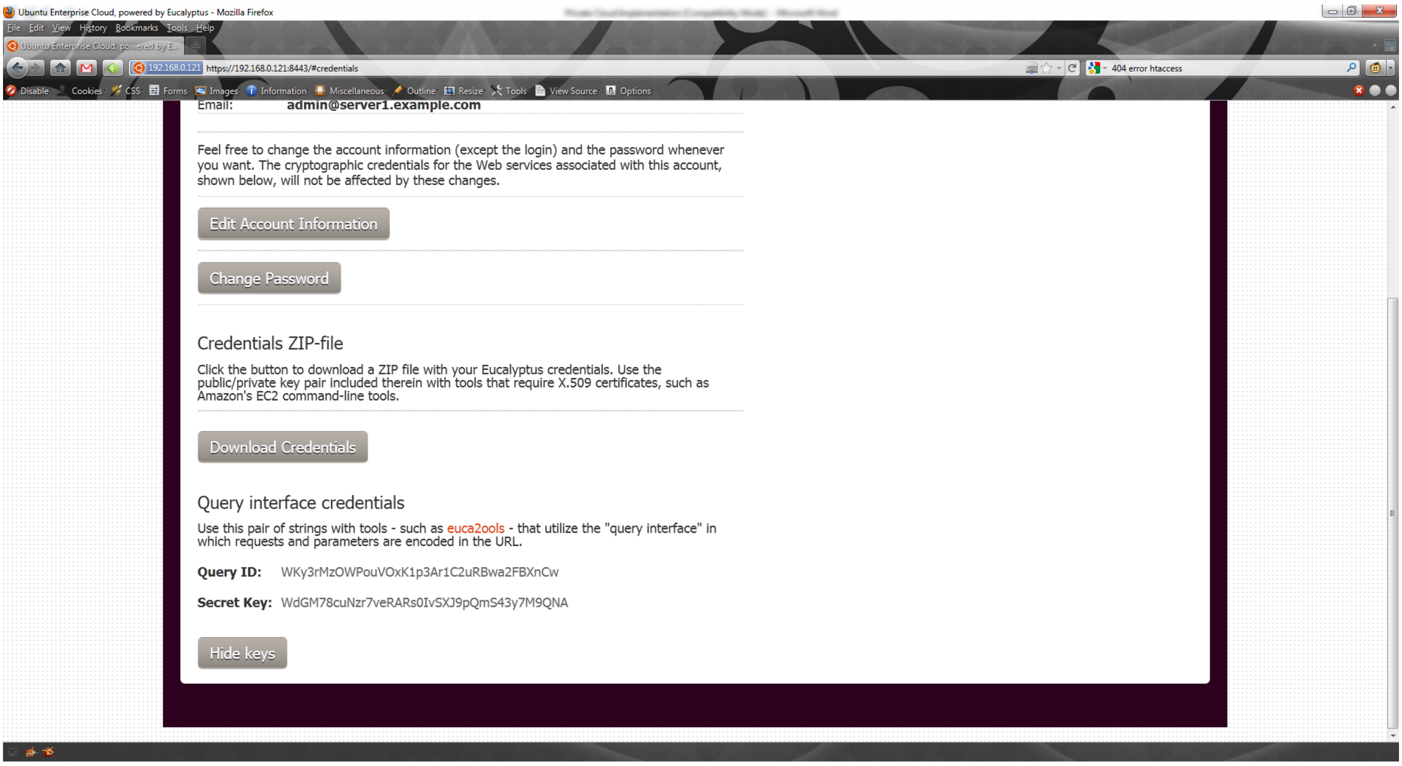
Endpoint Name	Endpoint URL
us-west-1	https://us-west-1.ec2.amazo...
eu-west-1	https://eu-west-1.ec2.amazo...
ap-southeast-1	https://ec2.ap-southeast-1.a...
ap-northeast-1	https://ec2.ap-northeast-1.a...
myuec	http://192.168.0.121:8773/se...

Region Name: myuec

Endpoint URL: http://192.168.0.121:8773/services/Eucalyptus

Figure 12 the configured region for HybridFox plugin

The /services/Eucalyptus string refers to the http server files in the cloud controller installation. The Region is secured by credentials which are key-value pairs. For my installation the values are accessible via the web interface made available on https port 8443, as shown in Figure 13.



Ubuntu Enterprise Cloud, powered by Eucalyptus - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Ubuntu Enterprise Cloud, powered by E...
192.168.0.121
https://192.168.0.121:8443/#credentials
404 error htaccess
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Email: **admin@server1.example.com**

Feel free to change the account information (except the login) and the password whenever you want. The cryptographic credentials for the Web services associated with this account, shown below, will not be affected by these changes.

Edit Account Information

Change Password

Credentials ZIP-file

Click the button to download a ZIP file with your Eucalyptus credentials. Use the public/private key pair included therein with tools that require X.509 certificates, such as Amazon's EC2 command-line tools.

Download Credentials

Query interface credentials

Use this pair of strings with tools - such as euca2ools - that utilize the "query interface" in which requests and parameters are encoded in the URL.

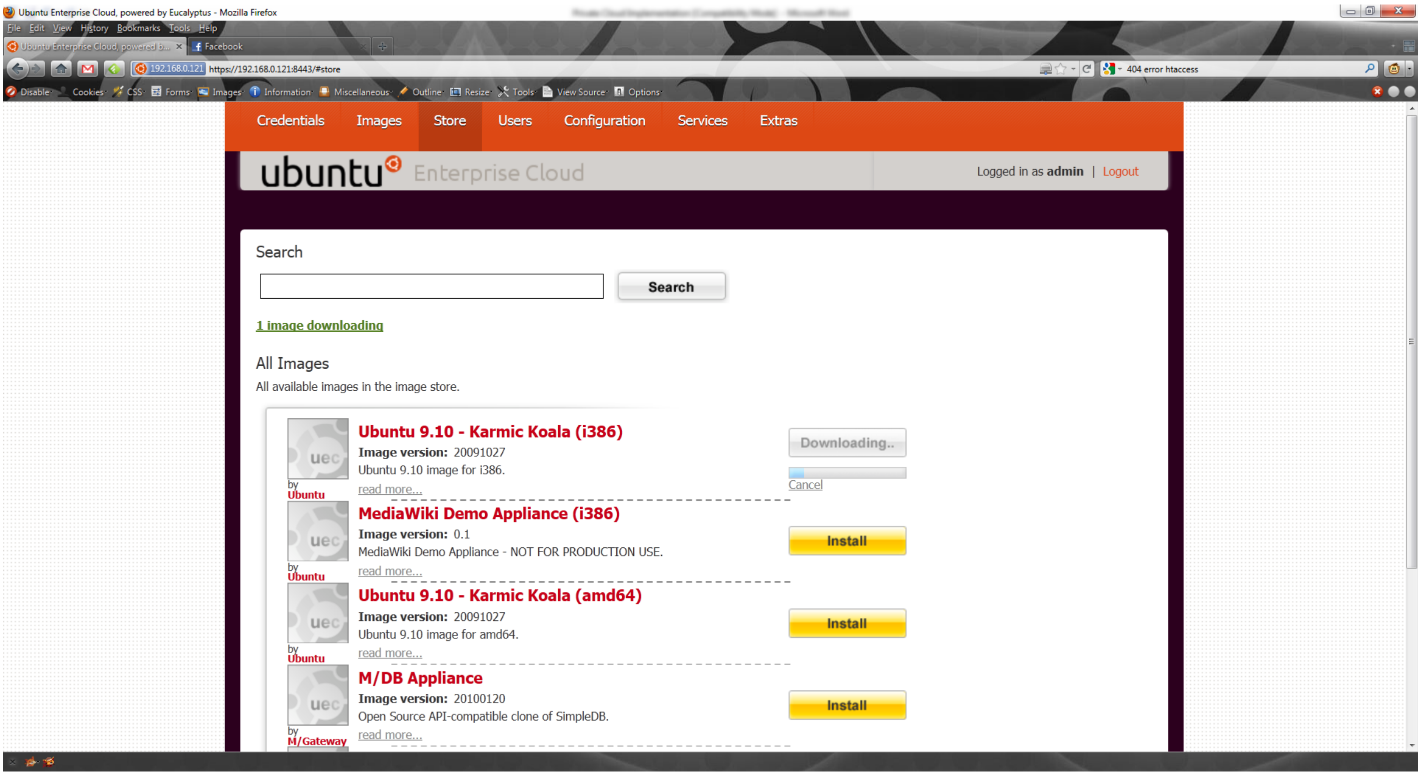
Query ID: WKy3rMzOWPouVOxK1p3Ar1C2uRBwa2FBXnCw

Secret Key: WdGM78cuNzr7veRARs0IvSXJ9pQmS43y7M9QNA

Hide keys

Figure 13 UEC Web Interface- Credentials page

Images within UEC can be downloaded through the Ubuntu store via the web interface; this is shown in Figure 14. I downloaded this image in order to test the workings of the “cloud” before putting emphasis on creating Windows images for deployment.


Ubuntu Enterprise Cloud, powered by Eucalyptus - Mozilla Firefox
https://192.168.0.121:8443/#store
Credentials
Images
Store
Users
Configuration
Services
Extras
ubuntu Enterprise Cloud
Logged in as admin | Logout
Search
Downloading..
Cancel
Install
Install
Install

Search

1 image downloading

All Images

All available images in the image store.

Ubuntu 9.10 - Karmic Koala (i386)
Image version: 20091027
Ubuntu 9.10 image for i386.
read more...
by Ubuntu

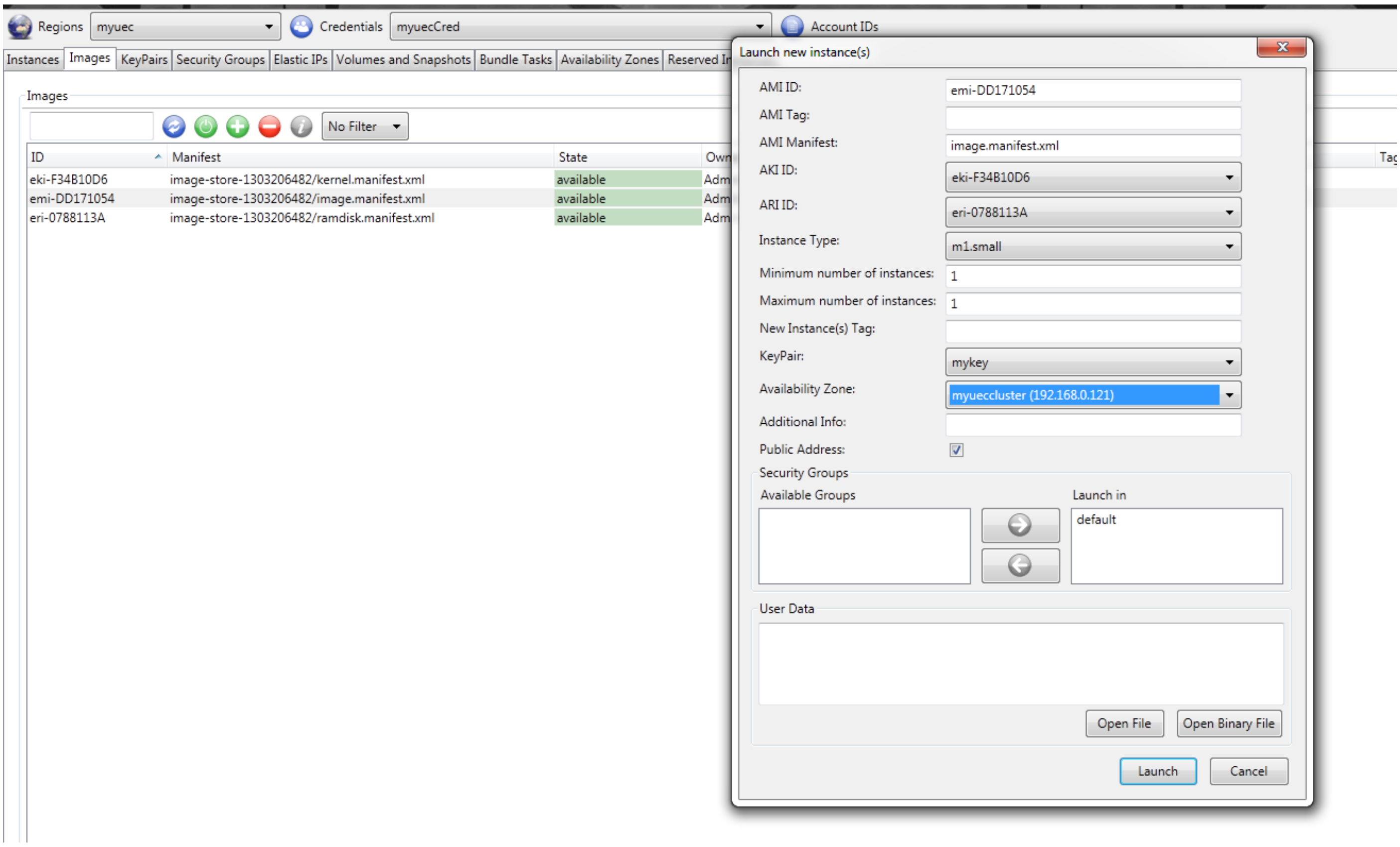
MediaWiki Demo Appliance (i386)
Image version: 0.1
MediaWiki Demo Appliance - NOT FOR PRODUCTION USE.
read more...
by Ubuntu

Ubuntu 9.10 - Karmic Koala (amd64)
Image version: 20091027
Ubuntu 9.10 image for amd64.
read more...
by Ubuntu

M/DB Appliance
Image version: 20100120
Open Source API-compatible clone of SimpleDB.
read more...
by M/Gateway

Figure 14 Downloading Ubuntu provided images

In order to run the image that is downloaded you have to configure instances, as shown in Figure 15.



ID	Manifest	State	Own
eki-F34B10D6	image-store-1303206482/kernel.manifest.xml	available	Adm
emi-DD171054	image-store-1303206482/image.manifest.xml	available	Adm
eri-0788113A	image-store-1303206482/ramdisk.manifest.xml	available	Adm

AMI ID:	emi-DD171054
AMI Tag:	
AMI Manifest:	image.manifest.xml
AKI ID:	eki-F34B10D6
ARI ID:	eri-0788113A
Instance Type:	m1.small
Minimum number of instances:	1
Maximum number of instances:	1
New Instance(s) Tag:	
KeyPair:	mykey
Availability Zone:	myueccluster (192.168.0.121)
Additional Info:	
Public Address:	☑

Security Groups
Available Groups
Launch in
default

User Data

Figure 15 Configuring instances

Figure 15 shows default options for launching an instance of the Ubuntu Karmic Koala image that was downloaded earlier. The ".emi" file is the image itself, ".eki" is the kernel image, which acts as a boot loader and the ".eri" is the storage that is provided for the image (virtual hard disk).

Working Ubuntu Image

I found it extremely difficult to initiate a working Ubuntu image. Using the image downloaded from the Ubuntu store shown in Figure 14 I tried to launch one instance, in order to save on the VM resources. However, although I left this to initialise for some time, it never entered an active state. After some debugging, and checking of the network i.e. firewalls, port blocks, and routing information I could not find a reason for this. My conclusion is that there is an issue with communication between the CLC and Node Controller, which is installation related rather than purely a networking issue. Due to this unforeseen issue and with time constraints I have decided to proceed with a guide on how to run a virtual lab session as per the requirements of the project, however I cannot provide evidence of a session being run.

Windows Image Creation

As mentioned previously, creating windows images is not a default application using the kvm architecture, however it is possible. Looking at the three file types associated with UEC images, we can see what needs to be created. The architecture states that we need an image file with a kernel image and ramdisk assigned to it. Windows is not a Linux/Unix based operating system and therefore the concept of a kernel image is unknown, however Windows does effectively have a kernel it's just not accessible in the same way.

An EMI file is a UEC specific bundle. The image itself can be an .img file. Creating an .img file for windows requires a ramdisk first, in this case 16 GB. This can be created using the following command:

```
dd bs=1M count=16000 if=/dev/zero of=w2k8.img
```

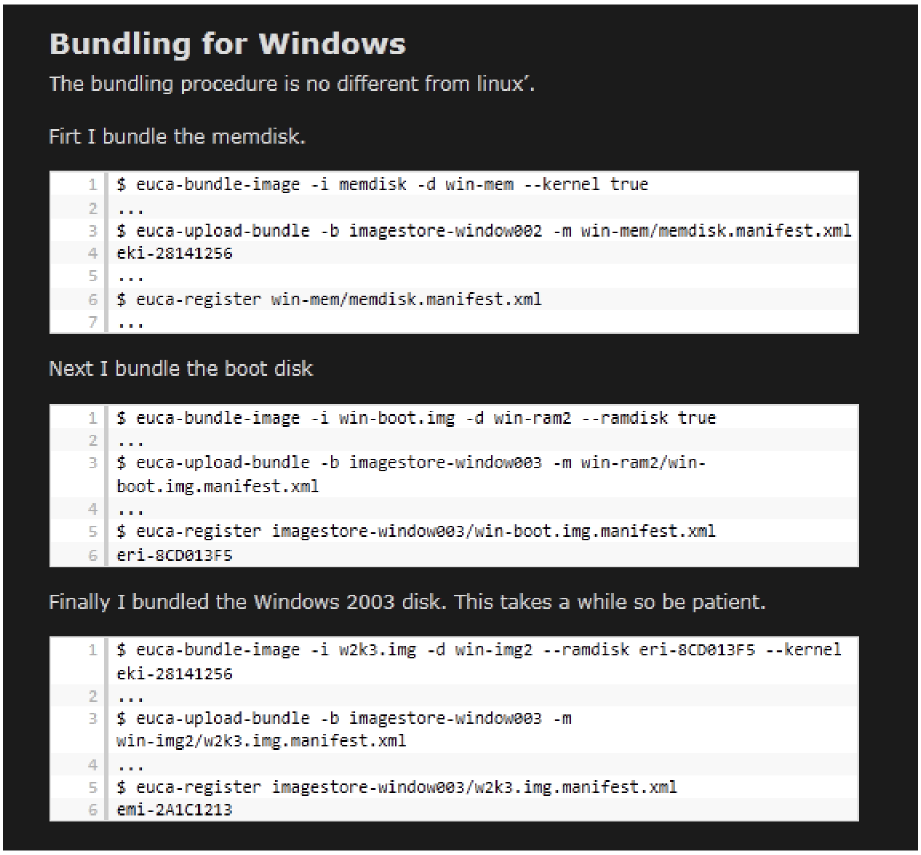
Then we use the kvm application to create the image:

```
kvm -m 1024 -cdrom win2008.iso -boot d -drive file=w2k8.img,if=scsi,boot=on -nographic
-vnc :0
```

This creates a kvm image (.img) from a disk image (.iso) of windows 2008.

Now we have our image we need to be able to run it, this requires a few extra system files in the windows VM in order for it to work properly within the kvm environment. These can be copied into the VM once network connectivity is established. It is important to ensure you bundle the correct network drivers (Intel e1000) with the original iso image to ensure networking capabilities.

Now that we have configured an image we can bundle it as an emi file, which allows launching from tools such as HybridFox. For this procedure I followed a guide from megam.info which did a test with Windows 2003, the same principal exists for 2008 , this is shown in Figure 16, below.



Bundling for Windows

The bundling procedure is no different from linux'.

Firt I bundle the memdisk.

```
$ euca-bundle-image -i memdisk -d win-mem --kernel true
...
$ euca-upload-bundle -b imagestore-window002 -m win-mem/memdisk.manifest.xml
eki-28141256
...
$ euca-register win-mem/memdisk.manifest.xml
...
```

Next I bundle the boot disk

```
$ euca-bundle-image -i win-boot.img -d win-ram2 --ramdisk true
...
$ euca-upload-bundle -b imagestore-window003 -m win-ram2/win-boot.img.manifest.xml
...
$ euca-register imagestore-window003/win-boot.img.manifest.xml
eri-8CD013F5
```

Finally I bundled the Windows 2003 disk. This takes a while so be patient.

```
$ euca-bundle-image -i w2k3.img -d win-img2 --ramdisk eri-8CD013F5 --kernel eki-28141256
...
$ euca-upload-bundle -b imagestore-window003 -m win-img2/w2k3.img.manifest.xml
...
$ euca-register imagestore-window003/w2k3.img.manifest.xml
emi-2A1C1213
```

Figure 16 Bundling UEC Windows images

Figure 16 shows the 3 files being created. The memdisk (kernel) .eki file is copied from a typical Linux memdisk image found in the syslinux package, used for network based Linux installations (PXE). The ramdisk image is created by building a virtual floppy disk similar to how we created the w2k8.img file and has a boot.ini file which specifies boot parameters and some system files used with the NT file system.

Running Windows Image

As previously stated, running an image will not work in my current installation, and it is not possible in the time available to get a running image working. Therefore the following section, which acts as my final "product" is a guide to running a virtual la, assuming these previous steps have resulted in a working image.

Solution

Following the failure to initialise a working Linux or Windows image on the virtualised cloud that has been built on the UEC platform, I have decided to still provide the solution as per the context of the project, however there is no evidence to show it active and running as this solution says. Therefore, this solution is a best-case scenario for how to deploy a UEC solution for the virtualised labs at BCU.

Infrastructure

Creating a private cloud and configuring images is the core aspect of this project, however without some decent hardware and infrastructure to run it off, there can be major downfalls such as performance hits and overloading of system resources. Therefore, it's important to note at this stage that the implementation produced here is not appropriate for a full installation and deployment. Firstly the two primary servers should be independent physical machines, with plenty of resources, the following are crucial for a smooth running system:

- 64-bit AMD-V or Intel-VT capable processor
- More than 4gb RAM (even more for Node Controller)
- Network Infrastructure without bottlenecks and with good bandwidth available (100mbps min, 1gbps preferred)
- Two network cards per machine for dual networking capabilities

From a networking point of view, if it's possible to run the cloud network on a VLAN (Virtual Local Area Network), this would be recommended. This keeps the bandwidth on the links fully available for the cloud infrastructure, by putting it in its own broadcast domain meaning it cannot see other network traffic. It is possible to route VLANs to the internet, meaning lab sessions that require this, can be run.

Windows Image Configuration

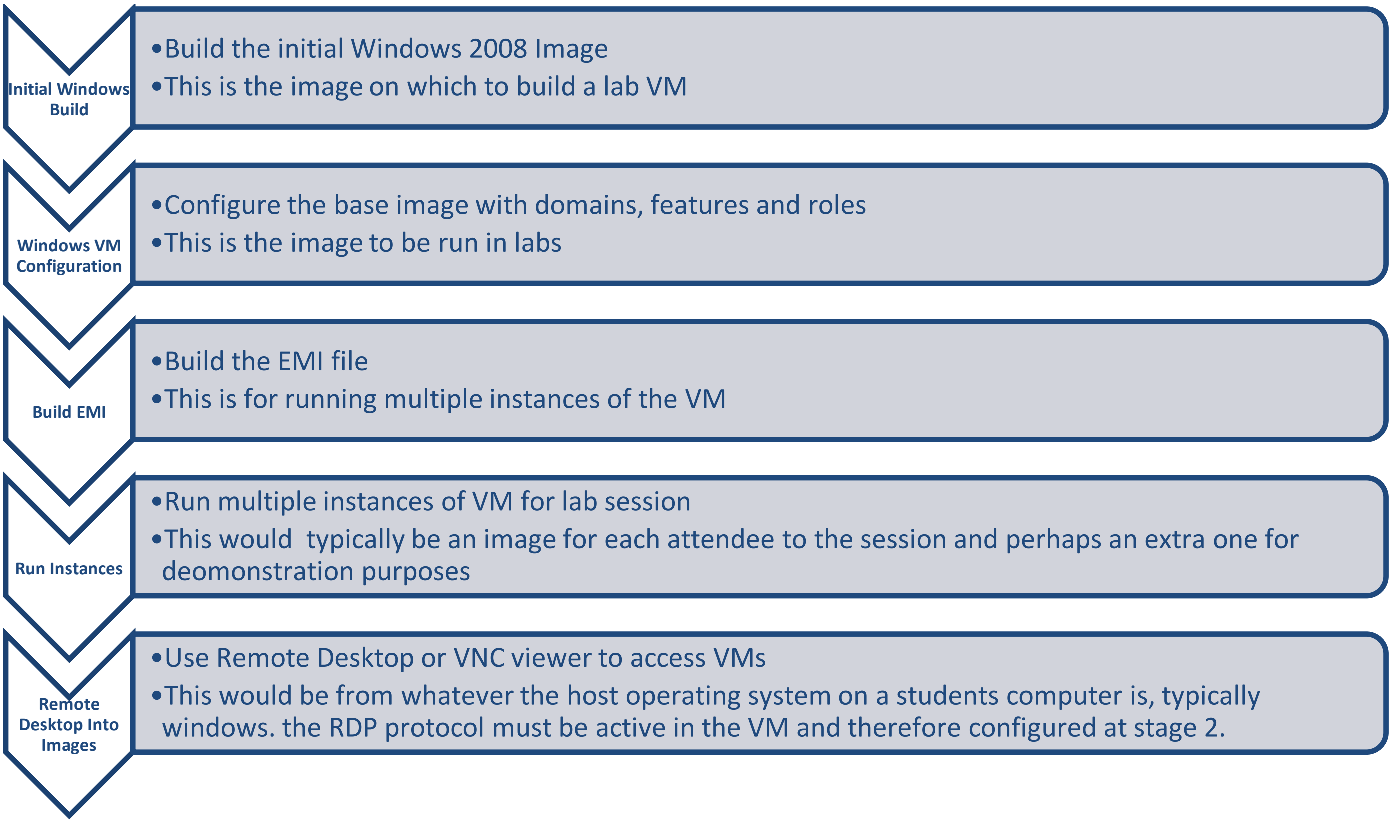
The implementation section talks about building Windows and Linux images to be deployed through the private cloud implemented. Once a base image for KVM is built it needs to be configured for student use, i.e. not just a basic installation. This includes the following:

- Username/ Password as per lab documentation
- Active Directory etc. installed on necessary images
- Removal of unnecessary services
- Updates installed to latest possible
- DNS/ DHCP, Domain name configured as per recommended by Microsoft Academy
- 3rd party tools installed where necessary e.g. putty

Once the base image has been configured with the above it can be converted into an EMI file as per the documentation specified in the previous sections. Converting into an EMI file means that multiple instances of the same image can be run; this is a necessity for running in lab sessions.

Lab Scenario

In Figure 17 I have illustrated the appropriate stages for the configuration and running of a typical lab session. The building and configuration of images has already been discussed, and once an EMI file is constructed this is a stage that can be omitted (until changes to the VMs need to be made). Running of instances has been covered also, and it's up to the session admin to decide how they do this and how many instances they require. The recommendation would be to access the Node controller through the HybridFox Firefox extension and just choose how many instances are required for the session. By doing this before a session starts, the start-up time is avoided by the students. This solves the major criticism shown in the primary research. The hardware at the back end will determine how long it takes the images to initialise and again this is a consideration that needs to be made at the building stage of the cloud.



Initial Windows Build

- Build the initial Windows 2008 Image
- This is the image on which to build a lab VM

Windows VM Configuration

- Configure the base image with domains, features and roles
- This is the image to be run in labs

Build EMI

- Build the EMI file
- This is for running multiple instances of the VM

Run Instances

- Run multiple instances of VM for lab session
- This would typically be an image for each attendee to the session and perhaps an extra one for deomonstration purposes

Remote Desktop Into Images

- Use Remote Desktop or VNC viewer to access VMs
- This would be from whatever the host operating system on a students computer is, typically windows. the RDP protocol must be active in the VM and therefore configured at stage 2.

Figure 17 Stages of configuring and running a lab session

The purpose of a cloud is to provide remote computing resources and this private cloud is no different. The virtual machines that are being deployed are running on the cloud infrastructure and in order for a user to gain access they need a client. In internet based (public) clouds the providers typically have their own clients that access the machines you run. Citrix is a common platform provider used for this. In this situation there is no need for complex applications, as the VMs are inside the local network all that is required is a method of accessing the VM operating system. Therefore if Windows is the remote OS the remote desktop application is sufficient. Mstsc.exe is the Microsoft Terminal Services Client, typically referred to as remote desktop; this allows the RDP protocol to access the remote OS and display its contents on the local machine. The /console parameter in mstsc allows direct access to the console, in other words the screen you would see if physically on that machine, not a separate session generated for the RDP user, this is an advantage as it uses no extra resources on the VM.

Evaluation

The purpose of this section is to evaluate the solution provided within context of the original aims and objectives, these were:

Aim:

To propose a new or improved platform for teaching virtualised based curriculum at BCU.

Objectives:

7. Identify a method of establishing change in an ICT Infrastructure
8. Model and capture existing platform of delivery
9. Research and analyse alternative platform and technologies
10. Propose and design a secure solution for the 'Infrastructure Technologies' theme at CTN/ BCU
11. Develop and test a prototype solution for delivery of virtualised labs
12. Evaluate the sustainability of the solution, and make recommendations

The new solution provided is a proposal on how the virtualised labs could be provided in an ideal world situation, as already established this would require some capital investment on hardware to provide the cloud platform. However the software cost is free, as Ubuntu is an open-source operating system.

Objective Analysis

Objective 1 stated that a method of establishing change management within ICT infrastructure must be identified. In this project the implementation stage, where building the cluster and basic images was discussed, is the prototyping stage of the chosen waterfall methodology. The Requirements and Design levels are the existing knowledge that is discussed and the proposed solution illustrated.

Objective 2 of modelling and capturing the existing platform was again explored within the existing knowledge section of this report. As discussed the existing model was very slow and unmanageable from an administrator point of view. Primary research conducted also highlighted many issues from a student perspective.

The 3rd Objective was to research and analyse the technologies available. The existing knowledge discussed many of the technologies that have been exploited to develop this design and solution, with a concentration on private cloud and virtualisation I have ensured that this proposal is technically sound and utilises existing technologies and ideas in a new way. This innovation is one of the core themes within this assignment.

Objective 4 was to propose and design a secure solution to build on the problems found by executing objective 2. Security is a common issue within cloud computing, the purpose of a private cloud is to keep the infrastructure "on-premises", this means that there is no access via a WAN, everything is controlled, administered and accessed via LAN. This is only the start of a secure solution. The design illustrated in the implementation section shows an internal cloud network as well as the local LAN. This internal network allows communication between only the two primary servers (and additional Node Controllers if necessary), this internal network utilises separate hardware (Network

Interface Cards) which could be controlled via a separate switch to the rest of the network, or segregated virtually using a VLAN. This offers security at the first three layers of the OSI network topology model. UEC also utilises encryption for communications between servers, this is through the use of the SSH protocol which is a common Unix/Linux communications tool that enables shell access to a remote server. SSH encryption comes in the form of key value pairs which are generated using the Diffie-Hellman Algorithm.

Objective 5 was partially met, its purpose was to develop and test a working solution. As the previous sections explain the solution has been designed as a best-case scenario. However, the objective is not fully met as a fully working image was not able to be run due to restrictions on the host machine and network running the virtual machines. The solution and guide provided should allow someone with Linux, networking and virtualisation skills to develop a working solution given more time.

Recommendations

The 6th objective is to provide recommendations on the solution, these are:

- The propose solution requires advanced hardware capable of virtualisation at the CPU level. Also a large amount of disk space is required for storing different images, and running multiple instances. RAM is also an essential commodity as this will affect performance with multiple instances running.
- Training of all appropriate staff on how to build and initialise images is essential to ensure a smooth overall system.
- More time needs to be spent on developing a prototype and there needs to be an extended testing phase, perhaps with 1st year students.
- Modification of the HybridFox extension could be made to make it simpler to administer the images, some controls could be phased out that are unnecessary for this implementation and some friendlier interface modifications could be made to allow for ease of use. This would require someone with good coding skills, familiar with Firefox extensions and the UEC commands.
- Recompiling the UEC servers and building images of them would mean simpler, pre-configured setups for future expansions, or extra implementations. Ubuntu is open-source and can be stripped of unnecessary packages and required packages cam be added at will, this means the whole OS is fully customisable.
- Exploration of other cloud/ virtualisation methods and weighing up of cost benefits and other advantages would mean a complete view of how the labs can be improved; Microsoft’s Hyper-V is another area that should be explored to offer an alternative to compare against.

Conclusion

The purpose of this project was to explore a new way of providing virtualised labs for the BCU/ CTN ICT themed modules as part of degree courses. The Primary research that was conducted suggested that many students were not happy with the current platform and many of the respondents provided good feedback on how they thought the labs should be improved. From this primary research and some existing knowledge in the areas of cloud computing and virtualisation I have developed a prototype solution that offers a best-case scenario view of how a new system could operate. The propose solution is a technically sound private cloud infrastructure which offers security, flexibility and scalability. The solution and evaluation provide an insight into how the system would operate in the university environment, and the methodology, as specified, is adhered to throughout. The implementation of the chosen solution may not have been entirely successful, however with a little more knowledge and more time available this is a very feasible solution to the problem.

Bibliography

Books

- Exam 70-643, Windows Server 2008 Applications Platform Configuration. (Microsoft Official Academic Course - John Wiley & Sons (2010) -ISBN 0470875046)
- Exam 70-646, Windows Server 2008 Administrator. (Microsoft Official Academic Course - John Wiley & Sons (2010) -ISBN 0470875070)
- Above the Clouds: Managing Risk in the World of Cloud Computing (It Governance Publishing, Kevin T. McDonald - It Governance Ltd (2010) -ISBN 1849280312)
- High Availability, Design, Techniques and Processes, (Floyd Piedad, Michael Hawkins - Prentice Hall PTR (2001) - Hardback - 266 pages - ISBN 0130962880)

Online Resources

- Eucalyptus Beginner's Guide: UEC Edition (PDF) (http://cssoss.files.wordpress.com/2010/12/eucabookv2-0.pdf)
- Microsoft.com
- Vmware.com
- http://itechthoughts.wordpress.com
- http://www.businesscomputingworld.co.uk
- http://www.techotopia.com
- http://hci-itil.com/options_frameworks.html
- Megam.info

Appendix I

Table/ Figure Sources

Table 2: High Availability, Design, Techniques and Processes, Floyd Pidad, Michael Hawkins, Enterprise Computing Series, Prentice-Hall, 2001 (PDF)

Figure 2: Eucalyptus Beginner's Guide: UEC Edition (PDF)

Figure 5: http://usep-ic.forumsmotions.com/t210-assignment-4-due-may-10-2010

Figure 6: http://www.sameeradilhan.com/what-is-system-development-life-cycle-sldc

Figure 16: http://megam.info/2009/12/24/windows-on-eucalyptus/