Enhancing the ‘Virtual Computer Laboratory’ (VCL) Concept through the Use of External Cloud Resources to Offer Distance Learning Laboratory Environments in an ‘Anywhere and Anytime’ Capacity

Abstract

This project aims to further develop a proof of concept solution for a cloud-based distance-learning capability, offering virtual laboratory environments (also called Laboratory as a Service (LaaS)) rather than relying on traditional computing resources owned by the host organisation (i.e. Anglia Ruskin University).

The service is intended to be ‘on demand’ irrespective of where the physical resources for the virtual laboratory are located or where the end user is situated (which are key characteristics of services commissioned within the cloud).

A primary objective is to demonstrate that an existing open source solution can be used to enhance software access (offered online as a service rather than installed on individual computers) through the use of typical cloud provision such as Amazon Web Service’s Elastic Cloud (EC2) which will extend both the functionality and scalability of the virtual laboratory. Key factors for the user would include ensuring that users can ‘save’ their work and return to it (i.e. ‘persistence’), and also reload previously saved versions (i.e. ‘revert to snapshot’).

Keywords

cloud, virtual laboratory, EC2, VCL, computer

Adrian Winckles (adrian.winckles@anglia.ac.uk) and Sebastian Brabiner (sebastian.brabiner@student.anglia.ac.uk)
Faculty of Science and Technology
Background
In many modern educational environments, it is common practice to ‘virtualise’ the student computer desktop to deliver

- A consistent experience which is easy to maintain and support
- The ability to offer a remote desktop experience supporting a 24/7 availability and distance learning capability

As Winkles and Jeries have previously suggested,

The growth within organisations of new computing paradigms such as ‘Virtual Desktops’ has allowed the concentration of IT resources centrally and the distribution of mobile computing platforms (e.g., smartphones, tablets, etc.) where users are able to access their work desktop from anywhere, at any time wherever an Internet connection exists.

The application of virtualisation for delivering quality distance teaching allows academic staff to offer customised desktops to suit the learner’s needs based on course requirements. There are differing definitions of virtualisation which include:

- to create abstract computer resources which are only virtual software versions of something rather than really existent (Michocka and Shwartsman, 2008)

and

virtualization enables one server or computer to act as many (Robb, 2008). (2013, p. 10-11)

Even though this approach has many advantages, it historically still follows the traditional ‘mentality’ of owning the physical computing hardware and storage to run all the virtual desktops within a centralised computing function. In theory, this could mean that, depending on demand for service, two extremes of feast or famine could occur. In other words, demand for a virtual computing desktop could outstrip supply meaning users are left without service for lengthy periods of time, or the capacity of central computing hardware might sit idle while not being properly utilised.

The modern paradigm that could provide the solution to these scenarios is Cloud Computing.

The Cloud and Amazon Elastic Cloud (EC2)
According to the National Institute of Standards and Technology (NIST) (Mell and Grance, 2011), Cloud Computing is

a model for enabling ubiquitous, 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 released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models (Mell and Grance, 2011, p. 2)

There are a number of ‘cloud’ characteristics according to NIST, which could help improve the resource issue:

On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling. The provider’s computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.
Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time (Mell and Grance, 2011, p. 2)

By utilising a cloud model of delivery, computing resources can be deployed from spare capacity available potentially anywhere on the Internet, transparent to the user. It has the potential advantage to the organisations IT service provider that resources can be supplied on demand (or Just-in-Time (JIT)) without the having to purchase and own the necessary computing hardware which could be idle or the organisation might not have enough capacity to meet user demand.

The Virtual Laboratory

As Winckles, Spasova and Rowsell (2011) have previously explained,

According to Wiseman, Wong, Wolf and Gorinsky (2008), a virtual laboratory is a ‘facility where students can access real laboratory equipment remotely’ (2008, p. 427). However, new technologies have furthered the above statement and have enabled the manipulation not only of real-world equipment but of its virtualised analogies as well, which gives a more specialised meaning to the term ‘virtual lab’. Leitner and Cane (2005) support it by saying that ‘any local computer hosting a simulation’ is considered a virtual lab. They elaborate even further and use the term to describe a computational grid, used for solving computational problems with geographically distant resources. (2011, p. 4)

The conventional model of a virtual laboratory is shown in Figure 1 where the resources needed for the virtual laboratory are ‘local’ and are finite (usually based on estimated usage) which does not solve the feast or famine scenarios.

![Figure 1 – Traditional virtual laboratory approach](image)

This project intends to take the Virtual Computing Lab concept and make the ‘cloud’ service central to the delivery of distance-learning laboratory resources as demonstrated in Figure 2. The initial research approaches in the short to medium term are:

- Explore the Lab as a Service (LaaS) concept to assess the impact and practicalities of traditionally resource intensive IT learning utilising a complex IT model, delivered entirely by distance learning using a virtual lab system
- Provide a basis for determining the requirements and suitable development platform so that a cloud based paradigm for virtual labs (LaaS) might be developed

Within a combination of a virtual laboratory and a cloud environment, the two terms become synonymous with each other, the virtual machine (vm) and the instance effectively are the same thing.
Design and Implementation of a ‘Lab as a Service’ (LaaS)

There are a number of key milestones that need to be considered in the design and implementation of a cloud-based virtual computer lab or ‘LaaS’:

- Use an existing virtual laboratory system to provide the mechanism for users to schedule lab resources and to provide an interface to commission / decommission cloud resources
- Research and select an appropriate cloud provider when resources can be commissioned or decommissioned easily based on a suitable cloud image being available
- Select a suitable course and set of modules where the virtual laboratory system can be evaluated
- Develop appropriate code with the virtual laboratory and cloud provider’s API (Application Programming Interface) to allow the commissioning and decommissioning of an appropriate ‘instance’ to take place once a lab session has been scheduled
The open source virtual laboratory platform which has been successfully utilised for past experimentation with projects is the Virtual Computer Lab (VCL), supported by the Apache Foundation and industrial sponsors such as IBM, was chosen as a platform for further development.

VCL has another advantage in that a certain amount of initial development work has been carried out on developing an ‘interface’ module for use with Amazon EC2 (Elastic Cloud) service offering allow the automatic creation and deletion of virtual machine instances within the cloud.

Computer Science modules such as Network Computer Systems, Operating Systems, and Internet and Network Security Forensics were identified as initial modules which could benefit from the use of on-demand computing resources commissioned within the cloud.

The initial idea for developing an EC2 provisioning module for VCL was proposed by Mann and Toews (2011), and a proof of concept Perl-based code project was launched with Google Code (in conjunction with the Apache Software Foundation), EC2-PM (2012), and earlier forum based developer proposals for EC2, incubator-vcl-dev (2010). The problem, however, with these additional modules is that, whilst a worthy effort in both writing and testing the code, as with any successful open source project volunteers are needed to maintain the source code as the main VCL project continues to evolve.

The key to the success of this project was ensuring access to Amazon EC2 could be arranged successfully and that the initial work on the creation of sample Linux-based instances was successful. According to Mann and Toews (2011), the use of Windows-based instances with EC2 is problematic, not because of the VCL itself but because the Windows licensing model causes problems when EC2 tries to create a new instance. The choice of Linux-based instances has the advantage that there are no licensing issues as Linux is an open source operating system and the majority of underpinning software used within the selected modules is also open source.

Further proof of concept prototype work is ongoing, which includes:

- Automated commissioning / decommissioning of instances using cloud technologies such as Amazon EC2 without the need for backend internal resource.
- Closed secure grouping of virtual machines into port groups networked together to irrespective of local or cloud residency to offer a sandboxed ‘complex’ laboratory.

Conclusion

The use of cloud provision for virtual lab resources is not an issue that can be ignored and traditional IT departments within educational establishments will increasingly be limited in the amount of budget that they can afford to have sitting idle or to save the state (persistence) of student work.

VCL would be a perfect solution for cloud brokering virtual machine instances for LaaS, from the point of view of its interface and function. It provides a clean, easily used interface, and a relatively resource-light core. It is multi-platform, modular and extensible.

It was disappointing that whilst the setting up of the cloud based environment was relatively simple, the actual adaption and testing of the EC2 commissioning module was problematic even though two different developers were engaged on the project the implementation of the EC2 module is still under development.

Both application developers experienced issues in terms of:

- Expecting the EC2 provisioning module to have been fully written, tested and operational (although the scope of the project expected some development work to get some parts of the code operational and then to test them appropriately).
- Issues involving the lack of up-to-date documentation for VCL and its quick update cycle which has left most (also similarly undocumented) modules no longer compatible (although this is a disadvantage of using open source code).
- Major issues with the underlying architecture of VCL mean it is not suitable for ‘hypervisors’ (Virtual Machine managers) which are not stored on the local network and the way addressing is allocated to the instances created. Cloud instances have neither of these things and therefore VCL cannot interface with them in the way it is designed to.
• Making VCL suitable for cloud brokering is a very large task, requiring restructuring of a large portion of its basic architecture. The existent prototypes show that the user interface can be restructured quite easily to allow for cloud brokering, and that an EC2 provisioning module is more than possible. It simply does not fit into the VCL as it stands.

The limitations that make VCL unsuitable for cloud brokering are due to be removed and, in the same version, initial support for cloud brokering will be introduced. Institutes desiring to use VCL as a cloud brokering tool would be best served directly contributing to the trunk open source development of VCL.

Upon successful development of the provisioning module, further work could be undertaken to develop an even more feature-rich VCL platform which could include,

• Integrating the IEEE P1876™ Standard for Networked Smart Learning Objects for Online Laboratories concept into LaaS.

• Developing the seamless connectivity and security between the cloud instances forming the complex laboratory with VXLAN/VPN technology to extend the LaaS concept truly between data centres for a true distributed complex laboratory in the Cloud.

References


