



Integrated (Private) Cloud Computing Environment for Access to Specialized Campus-bound Software Applications - Pilot Study

Dr. Emil H Salib, James Madison University

Professor in the Integrated Science & Technology Department at James Madison University. Current Teaching - Networking & Security and Cyber Intelligence Security. Current Research - Private Cloud Computing, Internet of Things (IoT), Mobile IPv6 and Design for Motivation Curriculum

Ian Healey, ASIS International

Mr. Alexander Ryan Chamberlain

Mrs. Livia S Griffith

Integrated (*Private*) Cloud Computing Environment for Access to Specialized Campus-bound Software Applications - Pilot Study

Dr. Emil.H.Salib, Alexander R. Chamberlain, Ian Healey, and Livia S Griffith
salibeh@jmu.edu, aleccham@cisco.com, Ian.Healey@asonline.org, and griffils@jmu.edu

Dept of Integrated Science and Technology (ISAT),
James Madison University (JMU), Harrisonburg, VA 22807

1 Introduction

A challenge in today's academic institution is to provide twenty four by seven (24/7) inexpensive access to computing resources independent of the physical hardware and software platform. Solutions being sought need to be simple to use and inexpensive to implement. Most universities utilize multiple computing labs running specific software. Licensing issues limit the ability to install software in multiple locations. Space limitations and class scheduling complicate the ability to provide the requested resources to a specific user population when needed. This can be frustrating, inconvenient and unproductive for students working on complex lab assignments with stringent deadlines. Faculty are searching for solutions that allow students on and off-campus access to expensive lab-bound software. Productivity merits the ability for students to have the flexibility to work on assignments as their schedules permit. In addition, universities bear overwhelming IT costs setting up and supporting labs spread across campus. Not only are expenses incurred in hardware costs but also in manpower hours setting up and tearing down computing labs, installing software and maintaining images.

In [1], the authors provide excellent survey of the opportunities of using Cloud Computing in educational environment. We also believe that the problems identified above can be solved by integrating a private cloud computing environment into James Madison University's educational resources. This challenge became the topic of a senior capstone project at James Madison University. Two students and their advisor proposed to address these issues by utilizing VMware vSphere [2] and Horizon View software [3] suites. Horizon View is a cloud computing solution that provides access to a virtual desktops pool managed by VMware vSphere suite and can be accessed on a wide range of inexpensive lightweight single-board computers, thin client devices or personal electronic devices already owned by students. In addition, the solution provides a secure off-campus connection for students to university lab-bound software. The solution allows faculty to create a master image specific to their teaching objectives. Administrators can clone, manage, monitor, and maintain the virtual desktops pools in a more efficient, timely and secure manner compared to today's decentralized environment. Some educational institutions have already taken steps towards implementing a Virtual Desktop Infrastructure (VDI) [4-8] as a replacement of gen-

eral purpose computer labs. In this paper we focus on replacing labs with a specialized licensed campus-bound software application.

An integrated private cloud computing environment has been created and implemented within our college's educational resources. In this paper, we present the solution's platform design, the three stages of solution implementation and the results from three successful pilot studies using the VMware vSphere and Horizon View software suites. Our ability to integrate VMware Horizon View with the *university's* Active Directory infrastructure has proven crucial to the success of this project. In the pilot studies, students had access to cloud-based virtual desktops using thin client devices (such as a Cubietruck [9] single-board computer) from on and off-campus. The clients ran the Horizon View desktop display protocols PC-over-IP (PCoIP) [10] and/or Remote Desktop Protocol (RDP) [11] over James Madison University's Virtual Private Network (VPN) services. The three pilot studies were conducted with a focus on usability, user access privileges, and preliminary performance assessment. The survey results collected supported many of our original objectives. We also identified a number of license issues and developed a set of recommendations for the university IT and network organizations if and when they are ready for campus-limited deployment.

2 Project Purpose, Definition and Objectives

2.1 Project Purpose

The main purpose of this project is twofold. One is to identify the opportunity of applying cloud computing in educational environment both for general and domain-specific computing needs as extensively described in [1]. The other is to develop the skills and expertise necessary to introduce Cloud Computing technology at James Madison University (JMU) Networking and Security curriculum. As a by product, we will package our private cloud computing solution (including hands-on labs) and make them available to other educators. One unique aspect of this project is its focus on the accessibility of virtual desktops (VDI) through the use of inexpensive and readily available thin client devices, such as, Raspberry Pi and Cubietruck; this has not been adequately addressed in the computer in education literature. The fact that we described in detail our solution implementation using a specific vendor's products should not be construed as an endorsement to that vendor and/or that vendor's products. As you will see in the section titled "Next Steps", we layout a plan of take advantage of emerging cloud computing Operating Systems such as Open-Stack [12].

2.2 Opportunity Statement

Labs across campus often have very specific and expensive software that can only be accessed from one physical location. The current approach of JMU IT organization is expensive and time-consuming. Computing lab hardware is replaced every three to four (3 to 4) years which costs the JMU a significant amount of money.

The following is a summary of the driving factors for undertaking this project:

- Lack of flexibility in accessing JMU licensed software;
- Lack of a single unified operating system containing all of the software needed for students to accomplish their educational tasks;

- The need for students to use multiple labs dependent upon course objectives;
- The need of faculty to have access to a flexible computing environment for the purpose of developing Cloud Computing curriculum;
- High cost of maintenance and upgrades of hardware and software in the labs across campus;
- Space limitations in a growing university environment;
- Flexibility in providing students access to frequently changing technology.

2.3 Proposed Solution

To address the efficiency and access issues of JMU's computing lab environments, we have proposed to configure a small-scale test bed consisting of VMware vSphere and Horizon View software suites. vSphere is a cloud computing and virtualization product available from VMware which allows instructors and administrators to create virtual desktops. Students can access the virtual desktops from different types of physical hosts, through web client software (HTML5 based console or VM Remote Console (VMRC) [13]) on the local network. Horizon View is a cloud computing environment that provides seamless and secure access to a virtual desktops pools managed by vSphere suite using PC over IP (PCoIP) [10] and Remote Desktop Protocol (RDP) [11].

Horizon View allows rapid deployment of specified lab environments. Simplified maintenance and timely updates is another advantage over traditional approaches such as using VMware Workstation [14]. Horizon View simplifies administration through centralized but shared resources such as CPU, memory, etc.. It allows for shared storage, fail-over and dynamic resource allocation among the created virtual desktops groups, thus enabling the full use of the processing power on the server.

2.4 Project Goal & Objectives

The project goal is to test the viability of Virtual Desktop technology utilizing an integrated secure cloud computing environment where students have access to JMU lab-provided operating systems and applications independent of location and access device.

To realize the above goal, we set out to accomplish the following project objectives:

- Implement a private cloud computing lab that provides access to operating systems containing cutting edge industry standard software;
- Enable remote access to labs, through JMU's Virtual Private Network (VPN) services;
- Provide easily configurable operating systems and virtual desktops for a wide range of courses and classes;
- Integrate the private cloud computing lab environment with JMU's Active Directory (AD) and Domain Name System (DNS) for seamless user provisioning, license verification and user access authentication;
- Enable students to Bring Your Own Device (BYOD) [4] that can securely connect to the Horizon View/vSphere environment to pull their respective virtual desktops and specific software applications; and

- Research the use of low-cost, low-power devices that could be configured to replace desktop and laptop computers in many of the labs across campus.

3 Solution Architecture & Components

Our solution is based on a standard client server model using Horizon View Connection appliance as our server and VMware Horizon View Clients [15] installed on one of many thin client devices available. Figure 1 shows the VMware Virtual Desktop Infrastructure (VDI) Computing Architecture based on Horizon View and vSphere software suites.

Below, we provide brief functional descriptions of the major components of the VDI computing architecture with a special emphasis on the Horizon View software suite.

3.1 ESXi Operating System & Server

ESXi is an operating system and a bare bones hypervisor for the VMware vSphere and Horizon View software suites. The ESXi hypervisor requires few resources. The ESXi server and vSphere components (such as vCenter) can be accessed via the vSphere client (Windows-based) or vSphere web client (involving vCenter).

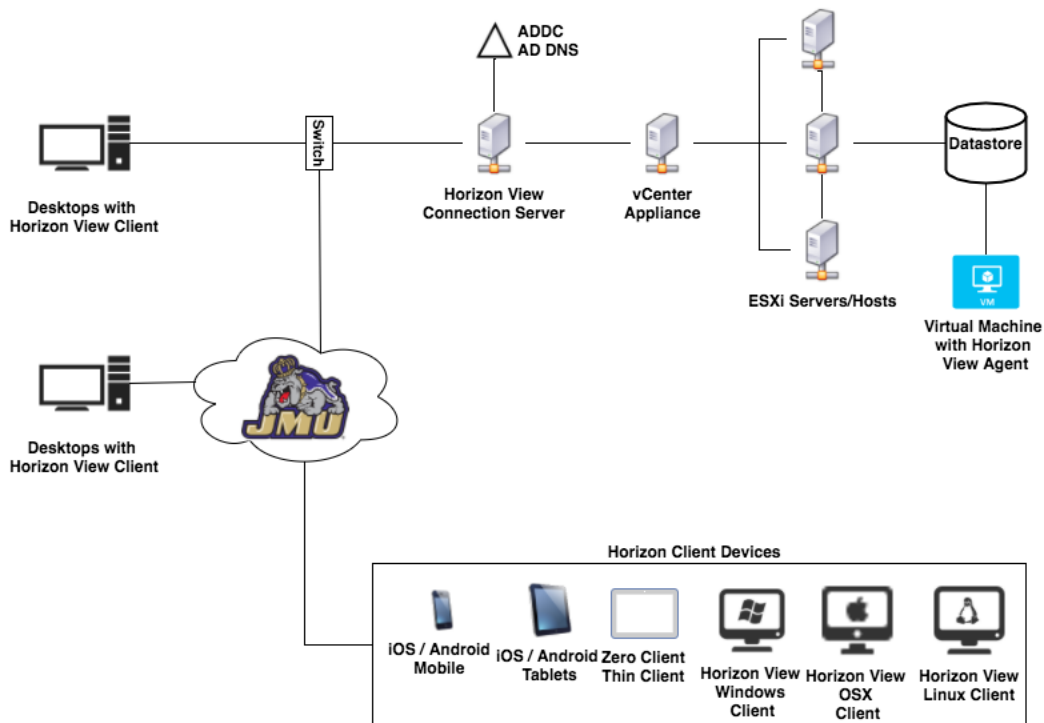


Figure 1: VMware Virtual Desktop Infrastructure (VDI) Computing Architecture

3.2 vCenter Appliance

vCenter is a management component of the vSphere software suite and is used in conjunction with ESXi servers. From the perspective of this project, the vCenter Server is included as an appliance and acts as a facilitator between ESXi (including VMs) and the View Connection server.

It manages pools of virtual desktops created under the Horizon View Connection server control. vCenter provides access to the web client interface (known also as Web Integration Client Plugin [16]). vCenter allows for resource management, managing privileges and identity services, and provides users and administrators web access to the virtual desktops running on the ESXi servers.

3.3 Horizon View Connection Server

The Horizon View Connection Server is virtual machine and internal server that manages, provides and brokers connections to pools of virtual desktops. It includes a View Administrator component which provides an interface to create, deploy and manage the virtual desktops pools. These pools must be entitled to specific users. The Horizon View Connection server is responsible for authenticating users against the Active Directory (AD) user data and access policies. Like all of the components of the Horizon View suite, the View Connection server typically runs on a Windows Server joined to an Active Directory domain.

3.4 Active Directory (AD)

To authenticate users, the vSphere web client provides administrators with additional authentication services against Active Directory (AD). We began our project using an internal private AD as a "proof of concept" before integrating our solution to JMU AD.

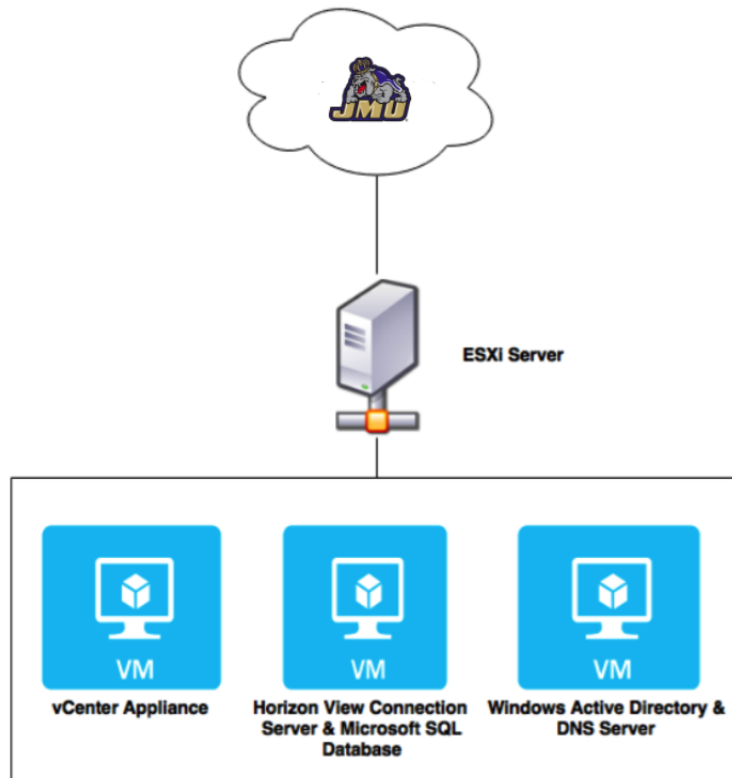


Figure 2: Horizon View Private Cloud Computing Environment (PCCE)

3.4.1 Internal Private AD

The Private Cloud Computing Environment (PCCE) configuration, as shown in Figure 2, was adopted in stages 1 and 2 of the project implementation (see Section 4.2). In that private configuration, the Active Directory Domain Controller was configured to host the private Active Directory instance. We configured vCenter to join and authenticate against this private Active Directory instance. This allowed us to create and assign privileges to users. We also had a Domain Name System (DNS) Server [17] co-located on the private AD Windows server. The DNS is a crucial element in the authentication process of the various components of the architecture requiring Fully Qualified Domain Name (FQDN).

3.4.2 Integration with the University AD & License Server

The Integrated Cloud Computing Environment (ICCE) configuration was adopted for stage 3 of the project implementation (see Section 4.3). In this configuration, we integrated the vSphere and Horizon View suites with JMU's AD and license servers as shown in Figure 3. At this point of the configuration, students would have the ability to use their JMU provided credentials (username and password) to access their virtual desktops. Software applications installed on the virtual desktops could receive license information directly from JMU's license server, eliminating further manual configuration by the system administrators. Integration with JMU's AD would enable college administrators access to the DNS server for provisioning of FQDNs.

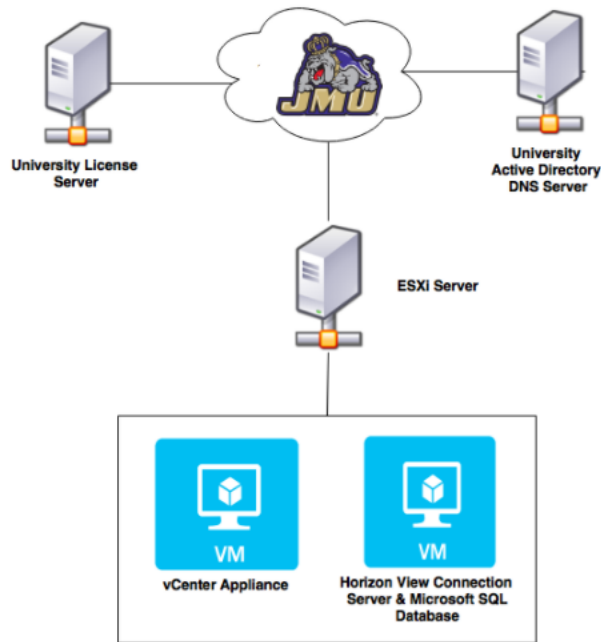


Figure 3: Horizon View Integrated Cloud Computing Environment (ICCE)

3.5 Virtual Desktops Pools, Horizon View Agent & Client

There are two types of virtual desktops pools: linked (persistent) pools and floating (non-persistent) pools. A linked pool contains virtual desktops that are entitled to specific users where each user

can only access one specific virtual desktop. Floating pools are pools of virtual desktops that do not follow a user. A floating pool is best suited for public settings such as a university's library where a user's profile and data are not saved. Unlike linked pools, the floating pools take up the least space on the vSphere/vCenter datastore. Managing floating pools requires the installation of View Composer. In this project we focused our effort on the implementation of linked virtual desktops pools and therefore View Composer will not be discussed in this paper. Horizon View Suite 5.5 (used in this project) supports only one OS virtual desktop type, that is, a Windows 7/8 image. Horizon View 6.0 introduced the support of Linux OS based virtual desktops [18].

The Horizon View Agent must be installed on all virtual desktops managed by the vCenter Server allowing communication with the View Connection Server. This occurs when the virtual desktop template is first created. Horizon View suite offers numerous Horizon View clients that enables users access to virtual desktops on thin client devices such as Cubietruck [9] and Dell FX100 [19]. In addition, many operating systems can have the Horizon View client installed [15]. Examples of operating systems supported are OSX, iOS, Android, Linux and Windows. Horizon View Client software strongly supports the Bring-Your-Own-Device (BYOD) approach.

3.6 On and Off-campus (Remote Access) to Virtual Desktops

In stage 3 the Horizon View environment was integrated into JMU's network (see Section 4.3). At this point students could gain access to the virtual desktops via JMU's VPN services. We found out that the VPN policy for the students dictates that the PCoIP port be blocked but not that of the RDP. As a result we had to configure the Horizon View Connection Server to accept either protocols.

4 Solution Implementation Stages

4.1 Overview

In this section, we describe the stages of our successful implementation of both a private and integrated cloud computing environments using the VMware vSphere and Horizon View suites. The initial implementation stages 1 & 2 focused on the private configuration. This approach is referred to as the Private Cloud Computing Environment (PCCE) (see Figure 2). In stage 3 of the project implementation, we integrated our solution with the JMU's AD and license servers. This approach is referred to as the Integrated Cloud Computing Environment (ICCE) (see Figure 3). Complete installation & configuration guide is available from the authors upon request [20].

4.2 Private Cloud Computing Environment (PCCE) - Stages 1 and 2

4.2.1 Installation and Configuration of vSphere Software suite

Figure 4 represents the starting point of our Private Cloud Computing Environment (PCCE) (to be referred to as stage 1). This beginning environment was the result of a previous capstone project [21] where the goal was to centralize virtual desktops resources on a single server using the vSphere suite over the stand-alone desktop VMware Workstation [14].

VMware vSphere ESXi 5.5 was installed on the host machine (Dell Precision T7600 tower server¹). We partitioned the hard disk such that the boot image resided on one partition and the system

¹It's fitted with dual Xeon E5-2630 CPU's, 256 gigabytes of RAM, 2x10G NIC and 3 TB's of storage.

software was stored on the other. We chose to install the vCenter as an appliance running on ESXi and it was joined to the internal private Active Directory (AD).

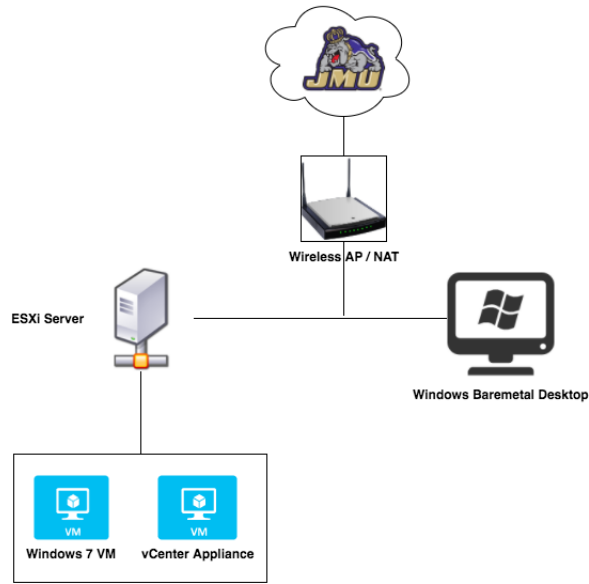


Figure 4: Networking Architecture of Stage 1.

Figure 5 represents stage 2 of the PCCE configuration where we deployed the virtual desktops required by the Horizon View Suite. We then configured the necessary Active Directory component. As this was a "test" environment we made sure to remove any policy and security settings that would hinder the deployment of Horizon View and its required software. We decided on modeling the structure of our Active Directory Tree as shown in Figure 6. This was designed to simplify implementation of stage 3.

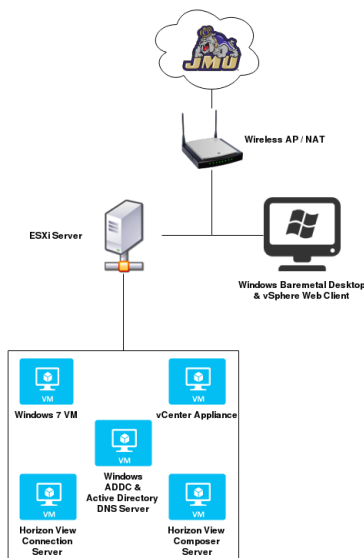


Figure 5: Networking Architecture of Stage 2.

4.2.2 Installation and Configuration of Windows AD

Authentication for the Horizon View (both the View Client and View Agent) is provided by Active Directory. The configuration of AD was a daunting task because of the many options and settings to consider (see Section 4.3.1).

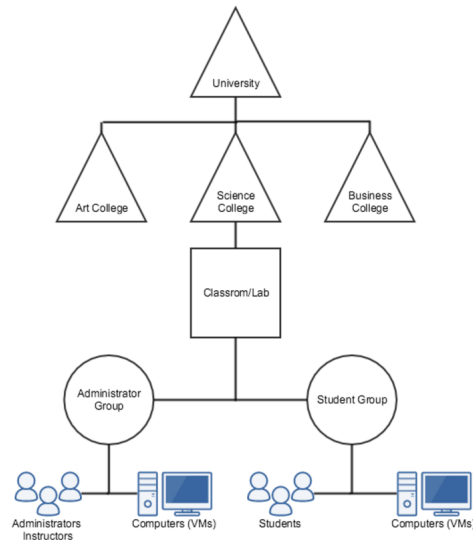


Figure 6: AD Organization Unit (OU) Structure/Tree used for PCCE.

4.2.3 Installation and Configuration of the Horizon View Suite

On a Windows Server 2008 R2 virtual desktop appliance (that is a virtual desktop running on ESXi and managed by vCenter), we installed the Horizon View View Connection Server. Next, we added vCenter to its server list using Horizon View Administrator web interface. The last step in the set up was to configure the Events database. We installed Microsoft SQL (MSSQL) server onto the Horizon View Connection VM; the most important variable to configure for MSSQL is switching the SQL Server from Windows Authentication to Windows and SQL Authentication.

4.2.4 Master Image with View Agent, Horizon View Desktop Pool and View Client

Creation of Master Image & Virtual Desktops Pool

Creating a “master Image” for View Connection Server virtual desktop pool was the same as creating a typical virtual desktop. We started by installing Windows 8 from an ISO (stored in the vSphere datastore), stepping through the standard Windows 8 installer, we were then presented with our virtual desktop. Through Windows Server network setting we joined the Master Image (virtual desktop) to the AD. We could now install specific course required applications. For Pilot Study 1 (see Section 5.2.1), we installed a pdf reader, Microsoft Office, and two web browsers. For Pilot Study 2 (see Section 5.2.2), we installed Multisim [22]. We installed Solidworks and ArcGIS for Pilot Study 3 (see Section 5.2.3). For efficiency of deployment, we have opted to use the Horizon View service, known as ThinApp [23]. It virtualizes applications by encapsulating application files and registry settings into a single ThinApp package. A part of this package, the

Horizon View Agent was the final application installed on the Master Image. Finally, we connected to vCenter via the vSphere web client to clone the Master Image virtual desktop to a snapshot, and convert that snapshot into a virtual desktop template.

In the View Connection server administrator panel, we created a virtual desktops pool for users by selecting the desired Master Image (template), entering the number of virtual desktops to be provisioned, and entitling the users and administrators who could access the desktop pool.

Installation and Configuration of Horizon View Client on a Thin Client Device

Once a pool of virtual desktops has been created, we were then able to pull and use VMs using Horizon View Client [15]. As we discussed in Section 3.5, the Horizon View Client is available for all desktop OS's and most mobile OS's. On launching the Horizon View Client, the user enters the FQDN of the View Connection Server. Once the user logs in, he/she is brought to a screen with his/her entitled virtual desktop pools. Clicking on the pool launches the VM console window and the user is greeted with his or her entitled virtual desktop with the installed applications in that pool. Figure 7 shows Windows 8.1 virtual desktop launched on iPhone iOS device. We have tested Horizon View Client on Apple iPad, Google Android, iMac, Linux Ubuntu Desktop and Windows.

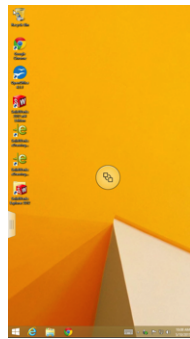


Figure 7: Windows 8.1 Virtual Desktop on Apple iOS iPhone.

4.3 Integrated Cloud Computing Environment (ICCE) - Stage 3

In stage 3 of the project implementation, we integrated our private cloud computing environment with JMU's computing infrastructure (see Figure 8). This enabled the students to use this service from anywhere at anytime, but using their JMU account credentials.

For this stage of the implementation, we were provided with a college owned and maintained server which is located in a secure location. This server came with ESXi and vCenter already installed. Our first step was to use the JMU's AD, as an identity source to the vSphere web client.

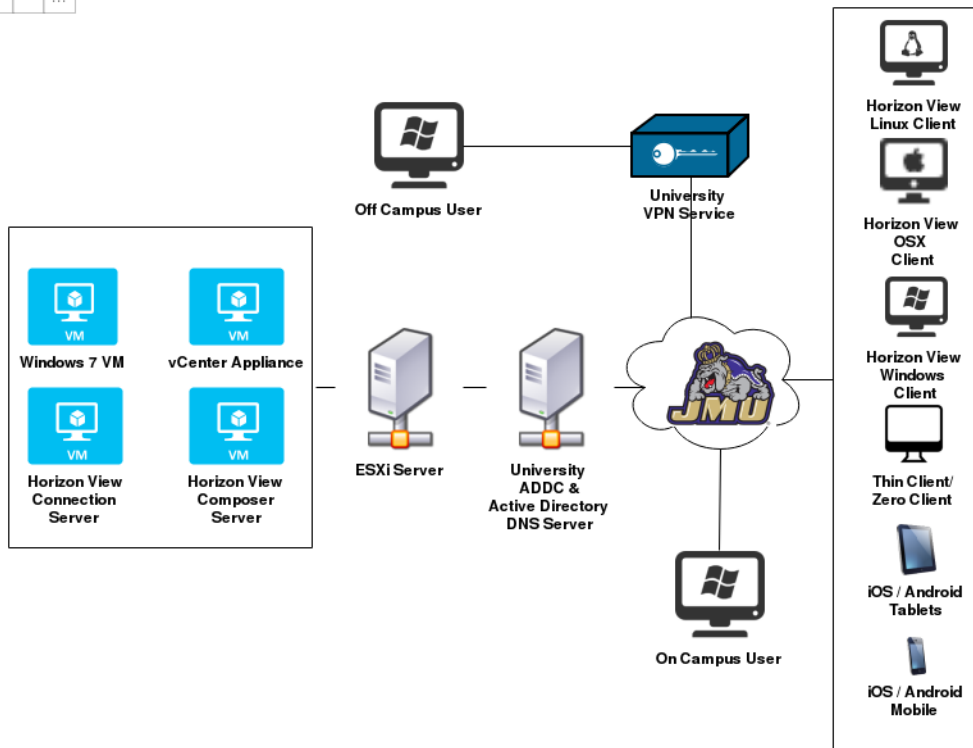


Figure 8: Networking Architecture of Stage 3.

4.3.1 Integration with the University AD

The integration with JMU AD required elevated credentials in order to access certain Administrator consoles of the Horizon environment; this was accomplished by the college network administrators. Next, we had to define a new AD model to meet Horizon View requirements. This task required close collaboration with the JMU's IT and Network Engineering teams. For example, we had to decide what type of Organizational Units (OU) structure to use and implement. As shown in Figure 9, we decided to go with a separate OU structure from the rest of *the college* OUs. This was driven by our requirements to apply different group policies to certain Horizon View virtual desktop templates. Also, in the case of a Horizon View persistent (linked) virtual desktops pool, different accounts are created with different levels of privileges to make changes to the virtual desktops. The final step to integrate with JMU's AD, was to join our virtual desktop template to the AD before creating the virtual desktops pools.

However, JMU's AD presented us with a new security policy setting that prevented users from installing software of their choosing onto the VM. We were not successful in reaching an agreement on this issue with JMU network security team. This was overcome by perfecting the master image and ensuring that the administrator installed the proper software before entitling users to the VM pool. After the integration with *the college* AD, we repeated the steps given in Section 4.2.4.

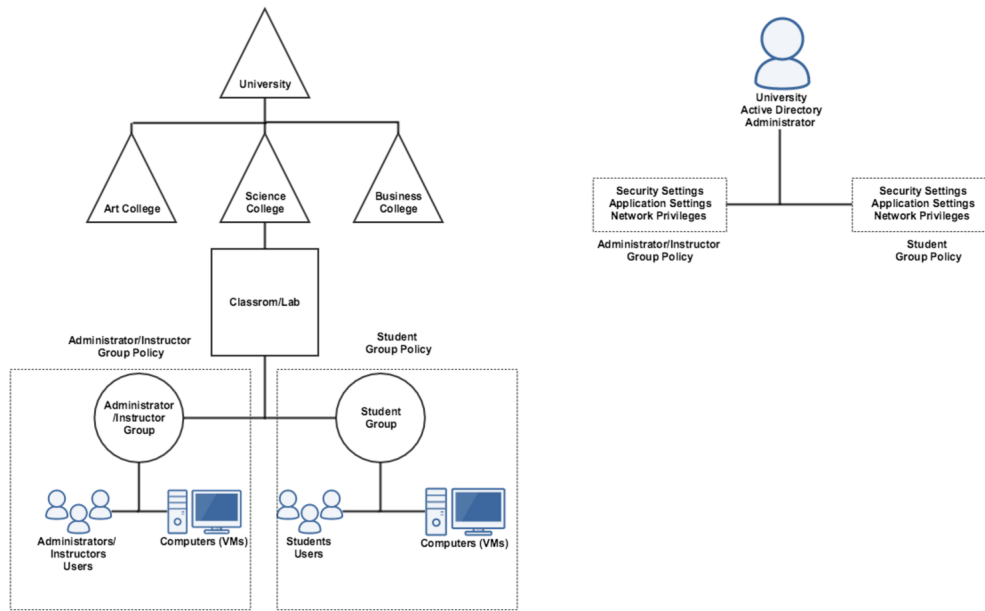


Figure 9: AD Organization Unit (OU) Structure/Tree used for ICCE.

5 Results - Pilot Studies

5.1 Overview

We conducted a total of three (3) pilot studies. Pilot study 1 included three students (technology major) and two potential administrators and lasted for one week. The focus of this pilot was to test the ICCE and identify possible areas of adjustment in preparation of pilot study 2. Pilot study 2 was conducted for one week with ten students (technology major) from a specific class using specialized simulation software (Multisim). This software is available only on the hardware desktops in one specific Network Computing lab. The last pilot study focused on a preliminary stress testing of virtual desktops running ArcGIS, a specialized CPU demanding software application.

5.2 Pilot Studies Procedure and Results

5.2.1 Pilot Study 1

We launched our first pilot study in an attempt to test and gather feedback on the Integrated Cloud Computing Environment (ICCE) from actual users. Our first lesson, came immediately after entitling our first five (5) users. As with any new environment our users attempted to install the applications needed for the classes they were enrolled. As mentioned in 4.3.1, the JMU computer security policies, configured by the JMU AD administrator, prevents any student accounts from installing software. This keeps students from unknowingly installing a virus creating a hole in JMU security scheme.

Our second surprise was that although administrators were able to remotely (via VPN connection) pull virtual desktops, the students were not. The VPN security policies for the students accounts blocked the PCoIP ports (this is the default and VMware recommended desktop display protocol for Horizon View suite).

5.2.2 Pilot Study 2

Pilot Study 2 included ten (10) students from specific technology course. Students in this course make use of Multisim, circuitry simulation software. In this pilot, we created a master image with Multisim and all other software required by the course instructor before provisioning the VMs.

Students were given instructions on "How to Access Horizon View" and had access on and off campus. Students were instructed on how to install the Horizon View client on Linux, Windows, or OSX platforms. Upon completion of the pilot study, we asked for students feedback using an online survey. We received numerous verbal responses which strongly supported the adoption of the ICCE solution. However, the number of students who responded to our online survey was not statistically significant nor conclusive to report on in this paper. Due to security complications, students could only access their virtual desktops on-campus but their feedback was positive in comparison to the physical limitations of lab room access only.

In our attempt to resolve this, we discovered it is possible to provision the virtual desktop pools to support both desktop display protocols: PCoIP and RDP. And that the remote access to the virtual desktops was possible through JMU's VPN connection using RDP.

5.2.3 Pilot Study 3

Pilot study 3 focused on preliminary stress testing of our solution. For this pilot, we used the PCCE environment located in our Networking Computer lab. We created a new master image on which we installed ArcGIS. This is a program that have high processing power requirements and are often requested by Geographical Information System (GIS) students to be remotely accessible on- or off-campus. First, we created a master image that included these programs. We decided on a pool size of six (6) virtual desktops due to the limited resources of the server (see Section 4.2.1) hosting the vSphere and Horizon View suites. The VMs were given a large amount of the server's resources. This was done so that we would be able to see if the Horizon View environment would remain stable during high load. After the pool was provisioned, we connected six (6) laptops to the Horizon View environment via Windows Horizon View Client.

We ran a Python script developed by a professor that was built to perform a watershed analysis using the ArcGIS software. This script can take anywhere from forty five minutes to six hours or more to run depending on the computers hardware and may be regarded as a measure in how well a system can use the ArcGIS software. Our objective was to run high performance applications (ArcGIS and SolidWorks) on a multitude of virtual machines all hosted on the same Horizon View environment to see if the system remained stable.

All six (6) of the virtual desktops finished the script at near identical times and none of them had crashed at any point. Figure 10 show the performance report provided by vSphere Web Client. The CPU resources on the host jumped up to the 80-90% range during the entirety of the test. Despite the huge load on the host, the environment did not sputter, the virtual desktops did not experience any lag, and the overall experience was smooth and functional. Further testing with large number of VMs is required to determine the trades off between buying specialized physical stand alone desktop computers versus the alternative of purchasing high performance hosts for the Horizon View environment.

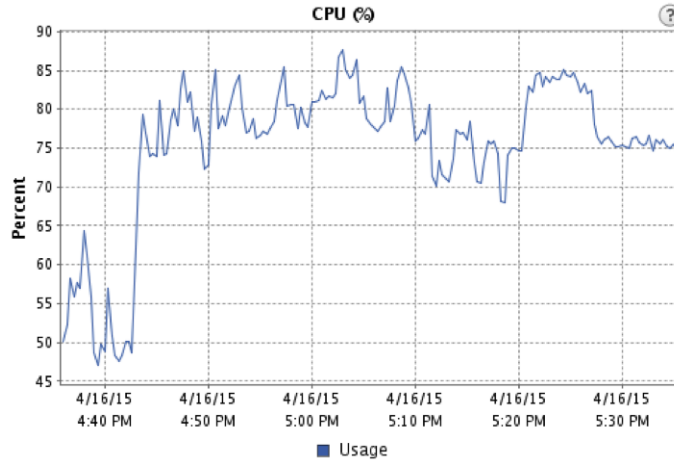


Figure 10: CPU Load (%) of the Horizon View Host during Watershed Testing and Analysis.

6 Discussion & Conclusion

Significant differences exist between a private "proof of concept" lab environment and a university integrated production environment. We were faced with many challenges and operational constraints and priorities in the university IT and network organizations.

However, we feel fortunate to have the opportunity to work with these organizations in conducting two successful pilot studies. In these studies we experienced first hand what it takes to roll out an integrated solution in a university computing and network environment. We learned about integrating our solution with the universities Active Directory, the design of service accounts and privileges they require, and configuring the virtual desktops pools permission policies. It was a rewarding and satisfying experience to see students actively using our Integrated (private) Cloud Computing solution. Witnessing students using their university credentials to access our solution on the university network was an outstanding accomplishment that contributed greatly to our technical, personal and professional growth.

We aimed from the beginning to make use of the Raspberry Pi [24] as our main thin client device. However, after many unsuccessful attempts at installing the Linux Horizon View client onto the early releases of Raspbian OS, we decided to instead move ahead with the low-cost low-power Cubietruck single board computer. We have learned recently that as of February 2015 the Raspberry Pi 2 was released with nearly twice the computing power of the Raspberry Pi that we tested. Also, the latest release of Raspbian OS officially supports the Horizon View Client on the Raspberry Pi 2 [25]. We did not, however, complete a full cost analysis on the energy or hardware savings that would be garnered over time from replacing desktops in computer labs with either the Cubietruck or Raspberry Pi 2.

With the knowledge gained as system administrators of the PCCE and ICCE environments, we also learned that, with the proper guidelines and structure, instructors and administrators can work together to configure a virtual lab for an individual class. This gives instructors the ability to create and structure classes in ways that they have not been available before by creating Master (or

Golden) Images [26] for an entire class to use on a just-in-time basis.

We have partially succeeded in enabling students to Bring Your Own Device for using their virtual desktops. With the problem of the PCoIP protocol being blocked through the university's VPN connection, the students who participated in our first two pilot studies were unable to connect to their virtual desktops from outside of the university network. However, this issue was resolved as discussed in Section 5.2.2.

In conclusion, we have achieved our major objectives laid out in Section 2. However, we also identified numerous topics that could be a fertile ground for many exciting and rewarding research and capstone projects (see Section 7).

7 Next Steps

Here is a list of potential future work that senior students could pursue for capstone projects:

- Research and experiment with Openstack [12, 27] (open source and free VDI platform) as a full replacement of the (high license cost) Horizon View solution,
- Work on achieving seamless integration with the University VPN using PCoIP,
- In collaboration with the university IT organization, conduct a full cost analysis between integrating the Horizon View suite and the four year cycle replacement of lab computers
- Conduct a medium size pilot study (fifty students) using thin client devices such as Cubitruck and Raspberry Pi 2 [25],
- Work with the university IT organization towards replacing all stand alone desktops in one or more computer labs and libraries with low-cost, low-power thin client devices,
- Collaborate with the university IT organization in developing a plan for campus-wide adoption of the Horizon View-based solution presented in this paper
- Investigate the ability to push applications to Virtual Desktops pools using Application Pools [28] (Microsoft Remote Desktop Service (RDS) [29]) to increase the efficiency of provisioning virtual desktop pools with students needed applications,
- Research and experiment with other Horizon View Components: View Transfer Server (for local mode), the View Security Server, Horizon Mirage, Horizon View for Linux and redirection (folder, USB, printer and scanner),
- Continue to investigate and assess the performance and security features of Desktop Display protocols including X11, VNC, RemoteFX (Windows Server 2008 R2 SP1 - part of RDS), Virtual Machine Remote Console (VMRC) and Citrix HDX,
- Provide comparison between existing approaches from other institutions with focus on performance related issues and,
- Move to Integrated "Public" Cloud Computing environment.

Acknowledgment

The authors would like to thank the Department and the University for the fund granted for purchasing the two Dell servers so that this project could be successfully completed. We would like also to thank Chrissy Rothgeb for helping us with the Active Directory integration, CPU performance measurement and spending the time and effort that made this project a success. Special thanks are in order for James Madison University (JMU) IT and network teams. Many thanks go to the ISAT students who participated in the pilot studies 1 and 2 and their informal feedback that has been proven invaluable to us in the successful delivery of this project.

References

- [1] Jose A. Gonzalez-Martínez, Miguel L. Bote-Lorenzo, Eduardo Gomez-Sanchez, Rafael Cano-Parra, "Cloud computing and education: A state-of-the-art survey," *Computers & Education* 80 (2015) 132-151, Spring 2015.
- [2] "vSphere." <http://www.vmware.com/products/vsphere>, Retrieved: 2016-01-25.
- [3] "Horizon 6." <https://pubs.vmware.com/horizon-view-60/topic/com.vmware.ICbase/PDF/horizon-view-60-architecture-planning.pdf>, Retrieved: 2016-01-17.
- [4] Steve Zurier, "Colleges Deploy VDI with an Eye Toward BYOD." <http://www.edtechmagazine.com/higher/article/2013/03/colleges-deploy-vdi-eye-toward-byod>, Posted: March 14, 2013, Retrieved: 2016-01-25.
- [5] Seattle University, "VIRTUAL DESKTOP." <https://www.seattleu.edu/its/labs/vdi/>, Retrieved: 2016-01-25.
- [6] William & Mary University, "Virtual Desktop Infrastructure (VDI)." <http://www.wm.edu/offices/it/services/network/virtualdesktop/>, Retrieved: 2016-01-25.
- [7] Trinity University, "Virtual Desktop Infrastructure (VDI)." <https://inside.trinity.edu/information-technology-services/resources/virtual-desktop-infrastructure-vdi>, Retrieved: 2016-01-25.
- [8] Michelle Woods, Stefanie Poe, Elon University, "Virtual Desktop Infrastructure - VDI." <https://wiki.elon.edu/display/TECH/Virtual+Desktop+Infrastructure+-+VDI>, Posted on Oct 17, 2014, Retrieved: 2016-01-25.
- [9] "Cubieboard4-cc-a80." <http://www.cubieboard.org/model/>, Retrieved: 2016-01-17.
- [10] "PCOIP TECHNOLOGY." <http://www.teradici.com/pcoip-technology>, Retrieved: 2016-01-24.
- [11] Andrea Mauro, "PCoIP vs. RDP." <http://vinfrastructure.it/2013/01/pcoip-vs-rdp/>, Posted: 25 January 2013, Retrieved: 2016-01-25.
- [12] "New vSphere OpenStack Virtual Appliance (VOVA) – Havana Release." <http://blogs.vmware.com/vsphere/2013/11/new-vsphere-openstack-virtual-appliance-vova-havana-release.html>, Retrieved: 2016-01-14.
- [13] William Lam, "Standalone VMRC (VM Remote Console) re-introduced in vSphere 5.5 Update 2b." <http://www.virtuallyghetto.com/2014/10/standalone-vmrc-vm-remote-console-re-introduced-in-vsphere-5-5-update-2b.html>, Posted: 10/10/2014, Retrieved: 2016-01-25.
- [14] VMware, "VMware Workstation Pro." <https://www.vmware.com/products/workstation/>, Retrieved: 2016-01-17.
- [15] VMware, "Download VMware Horizon Clients." https://my.vmware.com/group/vmware/info?slug=desktop-end_user_computing/vmware_horizon_clients/3.0, Retrieved: 2016-01-23.
- [16] William Lam, "What is the VMware Client Integration Plugin (CIP)?" <http://www.virtuallyghetto.com/2015/12/what-is-the-vmware-client-integration-plugin-cip.html>, Posted on 12/10/2015, Retrieved: 2016-01-25.

- [17] Microsoft TechNet, "How DNS Support for Active Directory Works." [https://technet.microsoft.com/en-us/library/cc759550\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc759550(v=ws.10).aspx), Updated: November 19, 2014, Retrieved: 2016-01-25.
- [18] "Horizon 6 for Linux." <https://www.vmware.com/products/horizon-view/features/horizon-linux>, Retrieved: 2016-01-24.
- [19] "Dell FX100 Zero Client." <http://www.dell.com/us/business/p/dell-fx100/pd>, Retrieved: 2016-01-17.
- [20] Alexander Chamberlain, Ian Healey , "Setup and Installation Guides for VMware Horizon View Suite," *JMU Capstone Project, Advised by Dr. E. Salib*, Spring 2015.
- [21] E. H. Salib, Ryan Carter, & Ryan Lutz, "Hands-On Networking & Security Labs On-Demand," *Proceeding ASEE/IEEE*, 2015.
- [22] National Instrument, "NI Multisim." <http://http://www.ni.com/multisim/>, Retrieved: 2016-01-25.
- [23] "VMware ThinApp Reviewer's Guide." <https://www.vmware.com/files/pdf/VMware-ThinApp-Reviewers-Guide.pdf>, Retrieved: 2016-01-24.
- [24] "Raspberry Pi." <http://www.raspberrypi.org/>, Retrieved: 2016-01-25.
- [25] "Raspberry Pi Thin Client project." <http://rpitc.blogspot.com/>, Posted: January 18, 2016, Retrieved: 2016-01-25.
- [26] Tom Rose, "Yes, One Gold Image in VDI is Possible." <http://blog.unidesk.com/one-gold-image>, Posted on Nov 17, 2014, Retrieved: 2016-01-25.
- [27] "VMware Integrated OpenStack." <http://www.vmware.com/products/openstack/>, Retrieved: 2016-01-17.
- [28] Vladan SEGET, "How to publish RDS application in Horizon View 6." <http://www.vladan.fr/publish-rds-application-horizon-view-6/>, Last Updated: April 11, 2014, Retrieved: 2016-01-25.
- [29] Vladan SEGET, "How to add RDS functionality to Horizon View 6." <http://www.vladan.fr/how-to-add-rds-functionality-to-horizon-view-6/>, Last Updated: April 10, 2014, Retrieved: 2016-01-25.