VDPilot: Feasibility Study of Hosting Virtual Desktops for Classroom Labs within a Federated University System¹

Prasad Calyam², Alex Berryman, David Welling, Saravanan Mohan
Ohio Supercomputer Center/OARnet, The Ohio State University
{pcalyam, berryman, dwelling, smohan}@oar.net

Rajiv Ramnanth, Jay Ramnathan
Department of Computer Science and Engineering, The Ohio State University,
{ramnath, jayram}@cse.ohio-state.edu

Abstract
Providing pervasive access to expensive, computational software such as Matlab and SPSS has always been a logistical and licensing challenge for faculty who want to train their students with industry standard software. In addition, faculty who want to manage lab exercises, assignments and exams are led to use e-mail to send and receive large files, thus limiting their ability to access and assist in the work-in-progress of students. In this paper, we present methodology and results from our “VDPilot” feasibility study to address such challenges by hosting virtual desktops for classroom labs within a federated university system in Ohio. The study leverages universities’ pre-existing high-speed network access at the regional-level in order to: (i) assess the user Quality of Experience (QoE) of accessing desktop applications remotely compared to physically going to a computing lab, and (ii) analyze the challenges and requirements for a shared service amongst collaborating universities. Salient results from both subjective and objective testing in the VDPilot study indicated that over 50% of the participants found the virtual desktop user QoE to be comparable to their home computer’s user QoE, and 8% found the virtual desktop user QoE to be better than their home computer user QoE. We also describe the findings in terms of resource characteristics, trusted identity, SaaS/DaaS applications and licensing, end-to-end performance monitoring and user support - that are informing the development of a virtual classroom lab “cloud service” at the regional-level to serve diverse training needs, and support multi-tenant and self-service capabilities.

¹ This article is a revised and expanded version of a paper entitled “VDPilot: Feasibility Study of Hosting Virtual Desktops for Classroom Labs within a Federated University System” presented at IBM Cloud Academy Conference (ICA CON), USA, April 2012.

² Corresponding Author.
P. Calyam, et.al.,

Keywords: Virtual Desktops, Classroom Labs, VMware VDI, Apache VCL, User QoE Measurement, Federated Cloud Service

1. Introduction
Providing access to expensive, computational software such as Matlab, AutoCAD, SAS and SPSS has always been a logistical and licensing challenge for faculty who want to train their students with industry standard software. Although universities have labs with pre-licensed versions of the software available, lab access for some students is still inconvenient. Furthermore, many students need pervasive access to the software, and have trouble obtaining a license and installing the software correctly on their home computers. Faculty who want to manage lab exercises, assignments and exams are led to using e-mail to send and receive large files, and are limited in their ability to access and assist in the work-in-progress of students.

In this paper, we present methodology and results from our “VDPilot” feasibility study to address such challenges by hosting virtual desktops and shared storage for classroom labs within a federated university system in Ohio. The VDPilot project was initiated by the Ohio Board of Reagents CIO Advisory Council, as part of the OSC/OARnet VMLab [1] testbed efforts for desktop virtualization experimentation. The experiments focus on leveraging desktop virtualization beyond the boundaries of a single institution, and seek to federate shared infrastructure resources across a network of collaborating educational institutions to deliver hosted virtual desktops for use in virtual classroom labs. Furthermore, the VDPilot study leverages universities’ pre-existing high-speed network access at the regional-level in order to: (i) assess the user Quality of Experience (QoE) of accessing desktop applications remotely compared to physically going to a computing lab, and (ii) analyze the challenges and requirements for a shared service amongst collaborating universities.

As part of the study, we implemented a “virtual desktop cloud” infrastructure that concurrently provides ~50 faculty and students with secure remote access to lab software using thin-clients over the Internet. In this paper, we describe: (a) how the virtual desktop cloud infrastructure has been setup to meet the resource and security requirements of users, (b) the remote desktop technologies evaluated viz., Apache VCL [2] and VMware VDI [3], (c) demographics and system/network characteristics at the study participant sites, and (d) the workflow followed by participants in order to determine user acceptance of the virtual classroom lab access.
We also describe in this paper, the results from one of the early VDPilot trials with faculty and students, as well as some IT administrator participants. As part of subjective testing activities, participants were asked to compare going to a physical lab versus using Apache VCL and VMware VDI remote thin-clients while performing tasks in the virtual desktops using applications such as Excel, Matlab, SPSS, Windows Media Player, and Internet Explorer. After completing subjective testing, participants were asked to complete an online survey to provide feedback about their perceived user QoE while accessing virtual desktop applications with Apache VCL and VMware VDI.

Subsequently, the participants were asked to conduct objective testing, where they downloaded, installed and ran the OSC/OARnet VDBench software [4] for both Apache VCL and VMware VDI remote thin-clients. The VDBench software executes a series of automated tests within the participant’s remote thin-client in order to simulate or mimic the actions performed by participants during the subjective testing. While performing the tests, the software records quantitative performance information in terms of interactive application response times, and video playback quality metrics that can be used to identify bottlenecks and to correlate with subjective opinion scores of participants. It can thus provide a more ‘apples-to-apples’ comparison of user QoE at the different sites, and eliminate any outlier-biases or mood-effects of participants that reflect in their subjective opinion of virtual desktop application user QoE.

Results from subjective testing and online surveys in early trials showed that both Apache VCL and VMware VDI technologies were evenly liked and the participants would use either. In addition, usability results for the applications-under-test show that users preferring VMware VDI scored all of the application user QoE at high levels, whereas the participants who preferred Apache VCL scored application user QoE over a wider range of values. Further, results from objective testing showed that the quantitative user QoE measurements over the wide-area network were similar to some of the best local lab i.e., local-area network conditions, and had a high level of correlation with participants’ subjective QoE rankings.

Lastly, we discuss the next steps for the study based on the encouraging results gathered from the initial trials. Particularly, we describe findings in terms of distributed resource characteristics, trusted identity within federations, SaaS/DaaS applications and licensing, end-to-end performance monitoring and user support - that are informing the
development of a virtual classroom lab “cloud service” to serve diverse training needs, and support related multi-tenant and self-service capabilities. Our ultimate aim of this cloud service is to use federated shared infrastructure resources at the regional-level in a manner that would simplify classroom lab computing for faculty and students, and reduce costs for collaborating universities that are interconnected with a high-speed network backbone.

The remainder of the paper is organized as follows: Section 2 reviews related literature and works to compare with the VDPilot study approach. Section 3 describes the VDPilot infrastructure setup for subjective and objective testing. Section 4 describes the subjective testing methodology, survey design and objective testing methodology. Section 5 presents results and discussion from the initial trials with participants. Section 6 details the plans based on our initial trials to transition the VDPilot into a regional-level cloud service. Section 7 concludes the paper.

2. Related Work
Diverse, small-to-large educational institutions such as Tamasek Polytechnics [5], Georgia Southwestern State [6], Prairie View A&M [7], Buffalo City School [8], and University of Tennessee at Knoxville [9] have started offering virtual desktop access to industry-standard software, and are pushing custom images for class content to be accessible by students on their mobile computing devices. Many do not offer shared storage and require students to use USB drives and other means to save and manage their virtual application data.

The Council of Australia University Directors of IT conducted a survey [10] in several universities across Australia, New Zealand and other participating international universities (e.g., UK, Singapore, Canada) to document the different maturity levels of hosting virtual desktops as an option to update open-access for students in the library. They found several institutions are conducting pilot trials and are in the early days of adopting desktop virtualization. No university system had a large-scale service being offered yet, and there were challenges and barriers with Linux/Mac platform support, cloud service compatible licensing, user QoE delivery and cost-effectiveness. North Carolina State University’s development, implementation and hosting of VCL [1] is an exemplar case of a successful large-scale cloud service that has been in production for over 8 years across 4 data centers and 2000 computational platforms, serving tens of thousands of students.
Our VDPilot study is different from campus-specific trials [5] – [9], and is looking at a federated cloud service amongst collaborating universities at the regional-level. In addition, we adopt both the open-source Apache VCL technology being used in the North Carolina State University implementation, and compare it with VMware VDI, a widely-used commercial technology for desktop virtualization. Our work is unique from existing work in terms of: (a) collecting and publishing user QoE measurements from subjective and objective testing, (b) conducting an online participant survey in conjunction to the subjective testing to evaluate pilot readiness, and (c) identifying detailed challenges and requirements for the cloud service to deliver SaaS and DaaS applications in virtual classroom labs over a high-speed regional-level network.

3. VDPilot Testbed Setup
The testbed established for the VDPilot project, shown in Figure 1 consisted of separate VDI-based and VCL-based environments with shared components that were used by both. The VDI environment utilized a connection broker for the creation and allocation of virtual desktops into a desktop pool. The VCL environment, on the other hand, utilized a web portal that allowed participants to reserve time on pre-created virtual desktops in a desktop pool. The thin-client protocols configured for VDI and VCL virtual desktop accesses were the default PCoIP and RDP protocols, respectively.

Although there were unique differences between the environments, a few shared components were required by both. A common Active Directory was used to create user accounts and their virtual desktop entitlements. This allowed participants to sign-in to the VDI connection broker and VCL web-portal with the same login accounts that they had established on the VDPilot user registration website. Aside from the VDPilot user registration website, the other shared components were the VDBench broker for objective testing, and the File Server that had virtual desktop application-specific content for both subjective and objective testing. The File Server was also responsible for hosting the images installed on the virtual desktops as well as for hosting the database where the test subjective and objective results were stored. The VDPilot website had separate participant user and administrator interfaces. The testing instructions, thin-client downloads and online survey link were accessible via the participant user interface, and the detailed subjective and objective results as well as survey responses/user comments were accessible via the administrator interface.
P. Calyam, et.al.,

Figure 1: VDPilot testbed showing VDI and VCL environments and shared components for virtual desktop application access via thin-clients

In the following sub-sections, we present details regarding the image creation for the virtual desktops, as well as the VCL and VDI environment setup steps for the subjective and objective testing.

3.1 Base Image Creation and Licensing

Once VCL and VDI environments were installed on 2 separate hosts in the VDPilot infrastructure, we next populated the file server with deployable base images. Similar to the testbed environment, common as well as some unique steps had to be followed to create the corresponding VCL and VDI base images. To host and manage the base images, the VMware vSphere package was used. When creating a new base image for either VCL or VDI environments, the first step in the process was to create new virtual machines in the vSphere client. The virtual machines for each environment were created and organized on separate hardware clusters as shown in Figure 2.
After issuing the command to create a new virtual machine on the designated host, vSphere then required the corresponding hardware specifications to be configured. For each virtual desktop in the VDPilot infrastructure, we provisioned 2 GB RAM, a single-core 2.0 GHz processor, and a 32 GB storage. The final steps of the configuration process configured which operating system the virtual desktop would run, what network hardware it would have, and whether or not the machine would have persistent storage. After these configuration steps were completed, the new virtual machine (i.e., virtual desktop) was booted and the operating system was manually set up. Following this, software installations of SPSS, Matlab, and MS Office were performed by mounting corresponding ISO files to the virtual desktops and completing the installation instructions. In order to deploy separate licenses for SPSS and Matlab in each of the virtual desktops, we setup pertinent licensing servers and delegators provided by IBM and Mathworks, and ensured a concurrent licensing that satisfied peak access of 50 participants.

3.2 VCL Setup

When creating a base image for VCL, the instructions provided at [1] were followed in order. An additional step of a DHCP server setup was needed to generate host names and allocate static leases for the newly created images. This additional step also meant that every time a new image was created, the ‘dhcpd.conf’ file on the management node had to be configured with a new entry containing the machine’s private and public MAC addresses. The ‘hosts’ file in the management server also had to be updated in order to properly route the hostname of the machine to the correct IP address as shown in Figure 3.
P. Calyam, et.al.,

```plaintext
host vcl-apps-v12-clone-four {
    option host-name "vcl-apps-v12-clone-four";
    hardware ethernet 00:50:56:bb:00:29;
    fixed-address 192.168.2.49;
    filename "/tftpboot/pixelinux.0";
    option dhcp-server-identifier 192.168.2.11;
    next-server 192.168.2.11;
}

host pvcl-apps-v12-clone-four {
    option host-name "pvcl-apps-v12-clone-four";
    hardware ethernet 00:50:56:bb:00:2a;
    fixed-address 131.187.127.49;
    filename "/tftpboot/pixelinux.0";
    option dhcp-server-identifier 131.187.127.11;
    next-server 131.187.127.11;
}
192.168.2.49 vcl-apps-v12-clone-four.vdpilot.domain vcl-apps-v12-clone-four
```

Figure 3: Illustration of ‘hosts’ file configured to map hostnames to correct machine IP addresses

With regards to the Apache VCL setup instructions provided at [1], it is relevant to note that when creating a base image for VCL, a SSHD server must be setup. In order to do this on a Windows machine, Cygwin, a Unix-like environment and command-line interface, had to be installed. Through Cygwin, an SSHD server was installed as a Windows service and set to run at startup. Once the SSHD service had been established, SSH keys were generated for the new image using a script provided with the Apache VCL package. The management server in VCL uses SSH to prepare a desktop for image creation such as: running certain system commands to minimize the file size, and making various registry changes. The SSH server is also used to create a local user account when a desktop registration is made. After the keys had been generated and the virtual desktop was ready for deployment, the next step was to prepare the VCL web interface.

Our VCL web interface was made accessible at [1] and served two purposes. The web interface allowed participants to reserve time on virtual desktops and also served as an administration panel for configuring and deploying computers with associated images. The first step in configuring the web interface for a new base image involved creating a new computer. During the computer configuration step, the hostname and IP Address established in the ‘hosts’ and DHCP file from earlier were input into the corresponding fields. The ‘Type’, ‘Provisioning Engine’ and other fields were configured as shown in Figure 4.
After the computer had been created, it was assigned to a virtual host. The final step in the VCL base image creation process was to run the ‘vcld’ setup command from within the management node. A ‘vcld’ setup script was provided with the Apache VCL package and was responsible for taking the initially created base image and making it available for deployment. After the script completed successfully, the image appeared in the ‘Images’ configuration window where it could finally be made accessible to the VDPilot participants.

3.3 VDI Setup
Similar to the VCL installation, VDI installation also required following unique steps. Once the correct software had been installed, a VMware View Agent application had to be installed on the host. This application was required in order to allow VDPilot participants to connect to the image using the VMware View Client software. After the installation, the virtual machine was powered off and a snapshot of the image was captured from within the vSphere client application. Once the snapshot had been saved, the final steps required making the image accessible within the VMware View Administrator.
The instance of the VMware View Administrator for the VDPilot, whose screenshot is shown in Figure 5, was hosted on the VDI hardware cluster. It allowed existing and new image pools to be created and monitored. For the VDPilot study, a ‘VDI-VDI’ pool was deployed with the new base image created earlier, and made accessible for participants. It is interesting to note that if the pool used during experimentation was ever updated with a new image, special instructions had to be followed to assure consistency within the pool. First, a new snapshot of the virtual machine would be made to capture the new software and changes. Next, the new snapshot would be selected as the default for the desktop pool. These steps assured that new desktops would spawn with the new base image. However, in order to update existing desktops, a “recompose” option had to be used. The recompose feature would kick any existing users off of their desktop, shut off the machine, and recreate it with the new image.

4. Testing Methodology
In this section, we describe the participant population, workflow followed by the participants in the subjective testing, the related online survey completion, and objective testing steps.

4.1 Participant Population
In order to effectively measure the user QoE of the VDPilot infrastructure, faculty, students and IT administrators from a diverse selection of universities in Ohio were asked to participate in the study. The participant institutions include: The Ohio State University, The University of Dayton, The University of Akron, Ohio University, Denison University, Walsh University, Sinclair University, Ashland University and Baldwin Wallace...
College. Students from the Ohio Supercomputer Center and OARnet also participated in the study. Over a period of three months, 38 participants registered and participated in the study.

### 4.2 Subjective Testing Workflow

All of the registered VDPilot participants were made to follow the same procedure when completing the subjective testing. After the participants had registered an account on the VDPilot website located at [http://vdpilot.oar.net](http://vdpilot.oar.net), they were asked to complete subjective testing of the VCL and VDI virtual desktops over a period of around ~20 minutes. The test procedure, whose screenshot is shown in Figure 6, was the same for both of the virtual desktops and required participants to perform simple tasks and assess user QoE satisfaction for the exemplar applications: Excel, SPSS, Matlab, Internet Explorer, and Windows Media Player. The application tasks were made simple enough to be completed quickly but also in a way to simulate typical use of the application. Application-specific instructions were provided to the participants as follows:

- **Excel** - Please create sample data and then create a chart for the data. Pay close attention to the response time of your actions.
- **Windows Media Player** - Please play a 30 second video from the virtual hard drive and observe the audiovisual playback quality.
- **Internet Explorer** - Please go to [http://www.cnn.com](http://www.cnn.com) and observe how well the web page performs.
- **Matlab** - Please load a pre-created script which creates a 3D horse model and then manipulate the model in the graph viewer. You should focus on the time it takes to create the graph, how responsive the graph tools are and overall response times while using the program.
- **SPSS** - Please load pre-created data, add an entry, and run a pre-created script to analyze the data; focus on how well SPSS runs, and how quickly issued commands were processed.
4.3 Survey Design

After completing the subjective testing, the participants were asked to complete an online survey on the VDPilot website as shown in Figure 7. The participants responded to questions on the overall user QoE, per-application user QoE for each of the VDI and VCL technologies, and also regarding the information such as operating system of the thin-client, and the last-mile network connection details (e.g., campus network, cable modem, DSL, wired/wireless adapter). These questions were used to determine which virtual desktop technology the participant preferred and how well the applications performed in the virtual desktop compared to their home computer. Mean Opinion Score (MOS) rankings that quantify the subjective satisfaction opinion on a scale of 1-to-5, with 1 being Bad and 5 being Good, were obtained for the overall user QoE as well as per-application user QoE.

4.4 Objective Testing Workflow

After completing the subjective testing and related online survey, participants finally completed the objective testing. The procedure for completing the objective tests consisted of three main steps. First, participants were asked to download Wireshark, the latest Java runtime environment and two versions of the OSC/OARnet VDBench Java client software, one for VCL and one for VDI. We remark that the VDBench software whose screenshot is shown in Figure 8 can run on both Windows and Linux platforms, and has capabilities for NIC selection for test initiation, and interactions with the benchmarking engine within the VDBench Broker shown earlier in Figure 1 at the server-side.
After correctly installing the applications and verifying that they worked as directed in the VDPilot website, participants created a new reservation for a VCL desktop. Once the reservation had been created and the participants had successfully logged into the VCL virtual desktop, they were asked to open the VDBench VCL Java client software. Next, participants would log into the software using their same account information from the VDPilot website. After hitting the “Run Tests”
P. Calyam, et.al.,

button, participants would bring their VCL desktop back into focus and click a confirmation button within a message box in order to start the tests. A scripted workflow of interacting with various desktop applications was then replayed and displayed to the user. During these tests, the response times and network utilization were captured on the thin-client side. Once the tests had finished, another message box would appear indicating that the virtual desktop connection could be closed. Participants were then shown the results of their tests from within the VDBench software, and the same results were stored in a database automatically at the server-side. Thereafter, participants followed the same set of instructions for the VDI environment by using the VDBench VDI software version, which ran through the same automated workflow as in the case of the VCL version.

5. Results and Discussion
In this section, we present results obtained from the subjective testing, online survey responses compilation, and objective testing. Also, we examine the correlation between the results to assess the VDPilot usability for supporting virtual classroom labs involving faculty and students.

5.1 Subjective Testing Results
Based on the analysis of the completed online surveys, it can be seen from Figure 9 that most of the participants used the Windows 7 operating system for thin-client access on a wired network, when performing the testing. The results also show that overall, both VDI and VCL technologies were evenly liked and the participants were willing to use either of the technologies for virtual classroom lab purposes.

An important question answered by the survey results was whether or not participants liked the virtual desktop experience and if they preferred the virtual desktop experience to their home computer experience. Based on the results shown below in Figure 10, it can be seen that 50% of the participants found the virtual desktop user QoE to be comparable to their home computer user QoE, and in fact 8% found the virtual desktop user QoE to be better than their home computer user QoE (particularly in the case of resource intensive applications such as SPSS). In contrast, only 25% of the participants preferred their home computer better, and 17% could not tell the difference between the virtual desktop and home computer user QoE.
Figure 9: Participants’ virtual desktop technology preferences grouped by their thin-client operating system and last-mile network connection

Figure 10: Preference of virtual desktop compared to home computer

Another analysis was performed to show participants’ user QoE satisfaction while using each of the five applications with the virtual desktop technology that they preferred. Figures 11 and 12 show the results for both VDI and VCL technologies, respectively. It can be seen that on average, participants had a good user QoE within all of the applications in the virtual desktop technology they preferred. It can also be seen that participants preferring VDI technology scored all of the application user QoE rankings at high levels, and those participants who preferred VCL scored the application user QoE rankings over a wider range of values.
Throughout the VDPilot study, participants’ comments from the online survey and from email communications were archived based on Institutional Review Board (IRB) guidelines at The Ohio State University. The quotes recorded indicated that the faculty and students liked the virtual classroom lab access in its current form. They also document an interest by two of the faculty who participated in the study about whether their students could use the VDPilot virtual desktops in their on-going course immediately. These comments demonstrate that *there is a real and current need for virtual desktop access with industry-standard software at universities.*
5.2 Objective Testing Results
Recall from earlier Figure 8 that the VDBench client software was used for recording and storing the application response times for an automated workflow that mimicked what the participants were instructed to do in the subjective testing in both VDI and VCL environments. The metrics considered in the objective testing with VDBench software included: Windows Media Playback Time, Windows Media Player Video Quality, Internet Explorer Text Page Load Time, Internet Explorer High Quality Image Page Load Time, Excel Open Time, and Excel Response Time.

In our laboratory in an isolated network, we collected a baseline of salient metrics shown in Table 1, under ideal and increasing amounts of emulated delay and loss conditions. The network health conditions were introduced via the Netem network emulator installed between a thin-client and a virtual desktop in the VDPilot infrastructure. Table 1 metrics are representative of the download performance of a single image, as well as a playback quality of a series of images i.e., a video clip playback. We normalized our baseline results so that 100% corresponds to the ideal user QoE, and lower percentages indicate relative degradation of user QoE.

The averaged and normalized values of the data collected from participants running the VDBench software for RDP and PCoIP thin-client protocols are shown in Table 2. The collected data shows that remote users QoE correlates with our baseline results for a network health with ~50ms delay, and 0% packet loss, which is expected given the participant locations over a regional-level wide-area network. We can observe that the objective results also show a high-level of correlation with the subjective user QoE rankings that inherently indicated there were no usability bottlenecks. Thus, we concluded that the VDPilot infrastructure is capable of effectively delivering satisfactory user QoE for virtual classroom lab needs of faculty and students in Ohio-based universities.

Table 1: VDBench baseline test results under varying network conditions – Normalized to the ideal case

<table>
<thead>
<tr>
<th>Network Health</th>
<th>PCoIP</th>
<th>RDP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Image Load Time</td>
<td>Video Playback Time</td>
</tr>
<tr>
<td>&lt;Delay ms, Loss %&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;0, 0&gt;</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>&lt;50, 0&gt;</td>
<td>79.27%</td>
<td>70.07%</td>
</tr>
<tr>
<td>&lt;0, 3&gt;</td>
<td>93.98%</td>
<td>99.12%</td>
</tr>
<tr>
<td>&lt;50, 3&gt;</td>
<td>74.25%</td>
<td>68.45%</td>
</tr>
</tbody>
</table>
Table 2: VDBench participant thin-client site results
– Normalized to the ideal case

<table>
<thead>
<tr>
<th></th>
<th>PCoIP</th>
<th>RDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Load Time</td>
<td>Video Playback Time</td>
<td>Video Playback Quality</td>
</tr>
<tr>
<td>82.07%</td>
<td>75.47%</td>
<td>54.13%</td>
</tr>
</tbody>
</table>

6. Transitioning VDPilot to a Cloud Service

Based on the encouraging results gathered from the initial trials, we are currently extending and transitioning our VDPilot to a regional-level federated virtual classroom lab “cloud service” to serve diverse training needs, and support related multi-tenant and self-service capabilities. Findings from our initial trials have informed us on cloud service issues relating to distributed resource characteristics, trusted identity within federations, SaaS/DaaS applications and licensing, end-to-end performance monitoring and user support.

In our current efforts, we are working towards supporting real classroom labs over the course of a full-semester - versus asking faculty and students to evaluate the VDPilot infrastructure for structured, repeatable application tasks over short testing durations. We are expecting to support peaks of up to 150 concurrent virtual desktop sessions, and plan to involve wider demographics as well as smaller institutions that have limited personnel resources, who have particularly expressed higher interest in the cloud service than relatively larger institutions.

To allow scaling of the system resources and shared storage, we are deploying extensible blade environments and storage solutions that satisfy I/O requirements for the peak scale of VDs, and that have RAID capability to avoid data loss due to disk failure. We are configuring network switches to provide redundant inter-connect between the servers and our core switches, and are planning to stress test them so that they are capable of reliably handling large amount of concurrent access of VD users. On the client-side enhancements of the VDPilot infrastructure, we are planning to distribute both physical thin-clients and zero-clients to remote faculty and students. With this, we aim to evaluate and compare the VDPilot offering user QoE for a standardized thin-client/zero-client approach versus ‘Bring Your Own Device’ (BYOD) scenarios, where faculty and students use their own computing devices to access the cloud service.
The next major enhancement involves integration of trusted identity management, which is key for authentication and entitlements of both applications and data folders in the cloud service within the federated university system. In our initial trials, user accounts had to be registered within the VDPilot website, and identical entitlements were configured, irrespective of whether a participant was a faculty or a student. However, in the future classroom lab use cases, we will leverage the IAMOhio federated identity program [11] within Ohio-based universities. In this case, entitlements will be based up on the particular base image being deployed corresponding to various offline applications (e.g., Matlab, SPSS, AutoCAD, or SAS) for a particular classroom lab. Also, the entitlements will be applied at the data folder and specific-application level access for different individual or project group roles. For example, faculty must have access to all data folders, whereas students or student groups must have access only to their respective workspaces.

Further, we plan to provision SaaS applications (e.g., Dropbox, WebEx) and explore “pay-as-you-go” licensing/app-provisioning, and provide capabilities of collaboration tools for online training, and online data storage/management. Such online application provisioning in conjunction with offline applications will be accomplished using emerging technologies such as VMware Horizon [12], which uses features of SAML federation found in the Shibboleth framework [13] that is widely adopted within academia and particularly, in the IAMOhio federated identity program. With such a setup, faculty and student passwords are protected as they do not cross institutional-boundaries, and a single account can be seamlessly used to access both offline and online applications.

We plan to integrate end-to-end application and resource monitoring for right-sizing and troubleshooting virtual desktop access. There are integrated monitoring capabilities within the VDI and VCL packages that we plan to further augment with commercial tools (e.g., LiquidWareLabs, EG Innovations). We also plan to enhance OSC/OARnet VDBench with more metrics collection and analysis. Further, we plan to investigate setting up an architecture that uses a “Unified Resource Broker” to provision virtual desktops using federated virtual desktop infrastructures at multiple campuses. Lastly, we plan to define service management and support procedures for multi-tenant and self-service capabilities, where university IT teams can setup dedicated sandboxes for staging, handling custom images, and operations of virtual classroom lab instances for their faculty and student groups.
7. Conclusion

In this paper, we presented methodology and results from our “VDPilot” feasibility study to address challenges of hosting virtual desktops for classroom labs within a federated university system. Through subjective and objective testing to assess the user Quality of Experience (QoE) of accessing desktop applications remotely, we were able to validate the usability and readiness of the pilot infrastructure in its current form. In addition, we were able to analyze the challenges and requirements for a virtual classroom lab “cloud service” at the regional-level with multi-tenant and self-service capabilities. As part of the VDPilot extension efforts, we are starting to address aspects of resource characteristics, trusted identity, SaaS/DaaS applications and licensing, end-to-end performance monitoring and user support.

Our ultimate aim of this cloud service is to use federated shared infrastructure resources at the regional-level in a manner that would simplify classroom lab computing for faculty and students, and reduce costs for collaborating universities that are interconnected with a high-speed network backbone.

Acknowledgement

We thank the peer-reviewers of the ICA CON 2012 and the Ohio Board of Reagents CIO Advisory Council for their helpful comments. We also thank Denis Walsh for helping us in licensing issues as well as participant selection, and Mark Beadles for inputs on integration of federated identity management. This material is based upon work supported by the National Science Foundation under award number CNS-1205658, and VMware. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

https://cwiki.apache.org/VCL/apache-vcl.html

[2] VMware Virtual Desktop Infrastructure (VDI) and VMware View –  
http://www.vmware.com


