The Development and Deployment of a Multi-User, Remote Access Virtualization System for Networking, Security, and System Administration Classes

Charles Border
Department of Networking, Security, and System Administration
Rochester Institute of Technology
cbbics@rit.edu

ABSTRACT
We have combined four industry standard technologies to create a flexible, modular and easily extensible virtual server environment for both distance and local students to use in our networking, security, and system administration classes. By combining the remote access technologies of Remote Desktop, the multi-user capabilities of Microsoft Terminal Services, the ability to share sessions of Remote Assistance, and the ability to manipulate virtual machines and virtual networks using VMWare Workstation, the Remote Laboratory Emulation System (RLES) allows students to conduct labs very similar to our local labs from either inside or outside our lab facility. While other people have been experimenting with various combinations of these technologies we have aggressively developed and implemented this system and feel that it has become a very important tool in our ability to cost-effectively provide computing infrastructure for local and distant students, and student and faculty research.

Categories and Subject Descriptors
K.3.1 [Computers and Education] Computer Uses in Education
K.3.2 [Computers and Education] Computer and Information Science Education
C.2.0 [Computer-Communication Networks]: Security and Protection

General Terms
Management, Design, Economics, Security

Keywords
Virtual machines, VMWare, computer labs, information technology education, security

1. INTRODUCTION
RIT, like most other universities, sees a great opportunity to expand its educational offerings through the use of distance learning technologies. We have spent lavishly on course management systems and considerable time converting our traditional lecture based courses to either wholly distance or combined distance and traditional lecture (blended) courses. At the same time, we have been working very hard to build our physical lab environment so that it can provide local students with the ability to manipulate both Unix/Linux and MS Windows operating systems (and save the results of their work), as well as various networking hardware in ways that created a very hands-on learning environment. The feedback we have gotten from students on our physical lab environment has always been very positive. Our problem was that we wanted to provide the same opportunities our local students had in our physical labs to distance students. After much investigation we have settled (for the moment, please see the Conclusions and Future Work section) on the architecture outlined below.

RLES is designed around our need to build a modular, scaleable virtual machine environment to support, many users at a given time, several varieties of Microsoft Windows, Unix/Linux as virtual operating systems, and have flexible support for various network configurations and topologies. Our problem was how could we deploy a system that would provide access to these capabilities with minimal incremental resources beyond the servers we had at hand? There were several functionalities that needed to be supported before we could judge any system to be a success. We spent the summer of 2005 going down blind alleys and finding all the ways that we could not do this. Finally near the end of the summer we came up with the idea that has since coalesced to form RLES.

This paper reports on our first year of experience using RLES for six classes and ten separate sections with a total of fifty students. RLES has been used as a platform for the manipulation of virtual servers, to deploy special purpose applications, to create a secure environment for computer forensic analysis, for the investigation of viruses and malware, and for MS level graduate projects. We have overcome some of the problems outlined by others who have developed similar systems. Students have reported being very pleased with the system and have not reported any conceptual problems with the manipulation of virtual computers. We have been very pleased with the results we have obtained and are excited about future uses of the system. Our plans for the future include migrating RLES to a higher performance blade server environment with an attached SAN for image storage. We also plan to develop a management front-end that will allow students, among other things, to dynamically choose the least busy server rather than being assigned to a single server.

2. DESIGN REQUIREMENTS
Any system we adopted needed to live up to the following design requirements:
Although we could require that distance students have access to the Internet over a broadband connection, we could not control that connection.

Whatever client software was to be used on the distance student’s desktop had to be freely available for MS Windows, Mac, and Linux operating systems.

We wanted student’s desktops to be as thin as possible with most of the actual processing happening on our servers. This would allow us to control most of the actual applications and allow us to avoid providing support for student’s desktops.

We needed to build a system that could grow over time to support many classes and provide an environment to support faculty research and outreach to other institutions.

The system needed to be able to support student work that spanned multiple sessions. Projects needed to be able to span an entire quarter with students saving their work as they progressed.

At least in our initial deployment, the system had to be built from industry standard hardware and software. As we are members of the MSDNAA, this included any software available under this alliance.

We wanted students learning activities to be concentrated on the material in their classes, not in how to work with the new system.

We wanted to avoid assigning a single system to a single student. This system had to be at least as cost effective as our current lab environment where each computer supports multiple students.

The courses we initially envisioned deploying this system for required that students have the ability to start, stop and configure operating systems over which they have administrative rights, to access specialized applications with heavy, or very specialized resource requirements (specifically OPNet IT Guru Academic Edition, and EnCase a forensic analysis tool), to create and manipulate network topologies and deploy network services, and to work with and analyze viruses and malware in a secure environment.

3. LITERATURE REVIEW

While there has been previous work in the area of using virtual machines to offer networking and system administration courses that work has been constrained in several ways.

One of the most relevant constraints in the literature is that the computers used to support users access to the virtual machine operating systems were housed in general purpose labs and had to support other uses [1]. In Virtual Machines – An Idea Whose Time Has Returned: Application to Network, Security, and Database Courses, Buller et al. describe their use of VMware GSX server in several classes all of which required that students have administrative rights to the virtual machine operating systems that were then manipulated to create interesting laboratory experiments. The design goals of Buller et al. did not include allowing students to access the machines remotely nor did they attempt to allow multiple students to access one computer at a given time. Bullers et al. also ran into a problem with the networking of the virtual machines. We also ran into this problem, but were able to work around it as will be reported in a later section.

Villanueva and Cook [4] used VMWare GSX server for both server virtualization and remote access to create a remotely accessible virtual learning space. Their solution to the problems addressed was interesting for two reasons, they relied on their administrative computing group to administer the system (as faculty members our ability to provide 24x7 access to a system is limited), and they allowed students to remotely access the system. This was a key idea in our work as well. They also had problems configuring the networking associated with the student’s virtual machines and while they were successful in finding a way to make the system work, they were forced to use an inelegant solution.

The work of Leitner and Cane [2] was interesting as much for the technology that they described as for their theoretical discussion of required laboratory outcomes and contexts. Their goal was to have their Virtual Networked Laboratory (VNL) more closely model the environment of their traditional labs:

The distance laboratory must foster active learning and comprehension construction on the part of the student. In the social setting of the in-lab experience, the learner interacts directly with other students, the instructor, apparatus, activities and other elements. These interactions guide interpretation and construction of mental concepts. They are key aspects to the laboratory experience that must be made a part of the distance education laboratory experience as well.

Leitner and Cane developed an architecture that relied on students remotely accessing a MS Windows Server 2003 physical computer running Dameware and then constructing experiments themselves to test the functionality of web services and other applications. They thus avoided the use of virtualization technologies relying instead on allowing students to remotely access the console of a running server.

4. SYSTEM CAPABILITIES

The flexibility of RLES is very intriguing. As we begin to offer more courses in either a wholly distance format or a combination of in-class activities with traditional lab based exercises and remote labs, RLES becomes a key tool. As long as students do not report conceptual problems utilizing VMs rather than physical servers there are many possible scenarios for the use of RLES.

4.1 Distance Classes

One of the questions that I routinely ask students who take classes in a distance format is “Where are you going to be when you take this class?” The answers have been very surprising. Many students who take distance classes are not at all distant from our campus. In fact, some students actually take distance classes when they live in our dorms. The reasons reported to me that local students take distance classes revolve around problems related to combining their work and class schedules, and a preference for non-lecture based classes. Distance, or it is
probably better to describe them as asynchronous, classes provide these students with opportunities they have never had before. The one missing element for these students has been the lab component. RLES allows these distance/asynchronous students to have a similar lab experience to that provided to our traditional students.

4.2 Combinations of RLES and Traditional Labs
The use of virtualization in RLES imposes an abstraction layer between the physical hardware and the operating systems. This allows virtual machines (VMs) to be moved from one physical machine to another. When students use RLES remotely they can be one of several users accessing the multi-user terminal servers to host their VMs and virtual networks. If students would like to use the same VMs in a traditional lab environment (provided the lab computers were also running the same virtualization software) this could easily be accomplished. Virtual machines could be moved from platform to platform with impunity. This would allow students to begin work on a lab exercise in the traditional lab environment, store their modified VMs on a network based storage system, go home and finish work on the lab exercise by starting their VMs on the remotely accessible terminal servers.

Figure 1. Port Forwarding in Remote Desktop

4.3 Security classes
Students are very interested in computer security, but we have two huge problems related to providing an environment where security related classes can be taught. The first problem relates to reliably providing access to computers running operating systems that have known security problems. It is difficult to maintain the operating systems at just the right upgrade level for many students to access. Also, once a system is upgraded it is no longer useful for the class. By utilizing read-only libraries of virtual servers that can easily be copied, stored and deployed, we allow students in a security class to have access to just the right vulnerability without spending all of our time creating just the right version of vulnerable operating system.

The second problem relating to offering courses in computer security is that sometimes we would like to allow students to see a virus or malware in action. This is impossible using traditional multi-purpose computer labs and in the past has spurred us to create special purpose isolated labs with no connectivity to the campus network or the Internet.

The ability to create virtual networks as well as virtual servers and clients whose connectivity can be easily controlled and filtered allows us to rethink the advisability of offering these previously taboo courses.

5. SYSTEM ARCHITECTURE AND CONFIGURATION
RLES combines four industry standard technologies: multi-user computing through MS Windows Terminal Services, with remote access through MS Remote Desktop Communications, remote support through MS Remote Assistance, and server virtualization with VMWare Workstation to create a system that has more than satisfied our design requirements and is very much appreciated by our local and remote students. The modular design of RLES makes it very adaptable to changing technologies. The current model relies heavily on Microsoft based products due to their ease of implementation and our ability to rely on the Active Directory to authenticate students to the system. Another important reason to rely on Microsoft products was the ease of use, and relative efficiency of the Remote Desktop Communications client. Students with limited bandwidth to the Internet were able to select the amount of bandwidth that the client utilized in ranges between 10 Mbps LAN speed to 28.8 Kbps modem speed.

5.1 Routing and Controlling Remote Access
The best tool that can be used to control what students can do inside RLES was to control how they could get to RLES and how RLES could get back to them. The Remote Desktop Protocol (RDP) by default utilizes port 3389 for both client connection and to reply to connected users. By applying Access Control Lists at the router interface we were able to block any inbound or outbound traffic that did not use port 3389 and RDP if a given class did not require access to such services as HTTP or HTTPS.

We also used port forwarding to connect incoming students to the server they were assigned to. One of our goals for the first year of
RLES was to keep load balancing across servers very simple and easy to administer. We also knew that we did not have adequate centralized storage resources or intra RLES connectivity to allow students to move their image files between servers. We accomplished these goals by assigning students to servers and then used 300 GB local hard drives to store student images. We also wanted to have a single DNS entry advertised throughout the RIT domain to route incoming students to RLES. We accomplished all of this through the use of port forwarding such that students would configure their Remote Desktop client to connect as shown in the following figure. A student connecting to rles netsyslab nssa.rit.edu:5993 would be forwarded by the router to a server named “vmserv03.rles.netsyslab.nssa.rit.edu”.

5.2 Virtual Networking and Virtual Machine Isolation
Our goal in developing RLES was to have a system where students could build a complete network including deploying all the services necessary to make their network useful. A problem that other authors have run into was to find a way to turn off the DHCP, NAT and DNS services provided by VMWare and have those same services provided by VMs that students could configure. We found a way to do this through the use of what VMWare calls “Teams”. By configuring VMs as members of a team, students are able to connect those VMs through a VLAN that they configure. They can then have one of the VMs have two NICs one of which is attached to the VM on the student configured VLAN and the other is attached to the VMWare VLAN that is providing NAT services. The VM hosting two NICs can then be configured as a router to provide NAT and firewall services between the student configured virtual network and the host operating system. Students are then free to use any addressing system (static or dynamic) and to configure any services as though they were in control of a real network (please see the network topology included with the case study).

5.3 Server Sizing
It is difficult to know how many, and what the specifications of individual servers should be. As we planned to use RLES for other classes we developed a very basic server sizing metric. Our metric was based on the number of virtual machines students would need to have active at any given time, the number of hours per week each student should need to access the server to do their assigned tasks(based on past experience), and how large each virtual machine disk partition needed to be.

5.3.2 Virtual Machines per Student
Depending on the material to be covered students will need to work with more or less virtual machines at the same time. A networking or system administration class will require that students have access to more virtual machines than a programming class. This is a good problem as it is one that we can wholly control through the requirements associated with the labs that we assign to students.

5.3.3 Hours per week
Realistically, how many hours per week do you expect students to use the system? While it would be nice to build a system that would allow all students to access the system at the same time that may be prohibitively expensive. If you tell students that the system is going to be busy, and when it is they are best off logging off and trying again at a later time, you are effectively creating a time slicing system. You could, if you wanted to, use a hard time slicing system where you scheduled student access to the system in two to three hour blocks. We did not do this, instead we allowed students to log in whenever they wanted to and gave them access to the terminal services manager with which they could check to see who else was logged into the system. When students found that several other users were using the system, they would simply log off and try later. Students were very understanding of limitations relating to the number of concurrent users. My feeling is that they knew that just as traditional labs have busy times that they may want to avoid, RLES also had busy times that they should probably avoid. The only time that they became angry about availability was when fellow students disconnected without logging off and left their virtual machines hogging CPU cycles when they were not using them. We have since resolved this problem by utilizing a group policy object that is assigned to all the students and runs a script that pauses running virtual machines and logs a disconnected student off the system after they have been disconnected for more than an hour.

5.3.4 Storage requirements
The math for determining how much storage you will need to have to deploy this type of system is fairly basic, but very impressive. Virtual machines are built in virtual partitions. The default size for a virtual partition is four GB. To determine the aggregate amount of local storage required to support student use multiply the number of students by the number of virtual machines per student and the size of each virtual partition. This simple math can lead to some very impressive numbers!

6. RLES IN ACTION: A CASE STUDY
Principles of System Administration is a bridge course in our MS degree in Networking and System Administration. It is designed to introduce students to the basic deployment of services in a networked environment and test their functionality. The course was offered for the first time in the Fall of 2005, but really got going in the Winter quarter of 2005. By this time we had worked out some of the bugs and had a new text book [3] that made for a much better student experience. Enrollment in the class doubled to sixteen and we rebuilt the two servers we had previously used to 3.4 GHz CPUs, with 2 GB of RAM and two hard drives (40 GB for the host operating systems, and 300 GB for student images). The students were surveyed at the beginning of the course to better understand their background in Windows and Unix system administration, and server virtualization. 75% of the respondents reported extensive windows experience, but only as a user, 65% reported having at some UNIX experience, and 70% reported having little if any experience with server virtualization. Student experience with RLES was very positive. The course consisted of a series of readings and lab exercises. The labs asked students to build the network shown in figure four and implement services such as DNS, DHCP, and LDAP. At the end of the course students were asked to complete a practical exam where they met with the faculty member (either locally or via shared remote connection) and demonstrate what they had implemented. Of the 16 students in the class twelve completed all the required labs and four (for different reasons) received an incomplete and two have since completed the assignments.
Administering the practical exam required that both the faculty member and the student have access to the student’s console session at the same time. This was accomplished by using Microsoft’s Remote Assistance capability.

7. CONCLUSION AND FUTURE WORK
There are many avenues available to us as we think about the future of RLES.

7.1 The Xen Project:
The current version of RLES relies on VMWare Workstation to provide both virtual networking and virtual operating systems for students to manipulate. The Xen Project is an open source virtualization server that takes a different approach to virtualization than VMWare. We feel that the Xen licensing model (open source) better fits our needs for the future. We intend to slowly migrate at least a portion of our new and existing infrastructure to this model over the next several months.

7.2 Hardware Infrastructure:
We are in the process of migrating RLES from its existing PC/local storage architecture to a more robust and scalable blade/SAN architecture.

7.3 System monitoring software development:
As an educational system RLES has some requirements that are very different from a standard industry service delivery system. We are in the process of developing tools that will allow us to monitor both the RLES systems and the guest operating systems utilized by students.

7.4 Hosting Issues:
We are in the midst of discussions aimed at finding a way to offload the basic service provision side of RLES to our on campus administrative computing group.

8. ACKNOWLEDGEMENTS
This project would not have come this far if it were not for the hard work and support of my colleagues: Bill Stackpole, Yin Pan, Bo Yuan, and Luther Troell. I am also indebted to the following students: Jon Schulman, Charles Consagra, Matt Lidestri, and Suraaj Guar.

REFERENCES

