EXPERIENCES WITH VIRTUALIZATION TECHNOLOGY IN EDUCATION

Timothy Bower
Kansas State University at Salina
785-826-2920
tim@ksu.edu

ABSTRACT

This paper describes how virtualization technology was used to facilitate the teaching of three classes and presents heuristic results describing when certain technologies were preferred over others.

INTRODUCTION

Virtualization technology has relevance to the class room as well as to the server room. Software virtualization technology is currently one of the most frequently discussed topics in information technology (IT) and systems administration trade publications. Virtualization technology allows additional modularization of applications, which makes services easier to maintain. The bottom line result of using virtualization technology in the server room is increased reliability and better utilization of high end servers, which results in reduced operating costs.[4] This paper, however, considers a completely different application of virtualization technology. It addresses the utilization of virtualization technology in the class room and shares first hand experiences from using virtualization to facilitate student access to the Linux operating system at both the user and super-user (root) security levels. Students were given options to use either traditional methods to access Linux based computers or to use one of two different virtualization technologies to complete assignments in three different classes. In many classroom situations, virtualization offered a preferred solution, but in other situations, a traditional approach was preferred. The objective of this report is present information to assist instructors and systems administration staff in planning for situations where students need to use other operating systems than what is available on laboratory computers.

An informal, *path of least resistance,* experiment was conducted in three classes with distinct assignment objectives. Each class has now been conducted two or more times with students given the same options each time and quite consistent patterns were observed regarding the technology choices made by the students. The primary factor affecting the technology choices was the requirements of the work being performed, which includes the required security level for the assignment, the amount of flexibility required and the duration of the activity. The abilities and preferences of the student was surprisingly a secondary factor with much less impact on the choices made. The students' level of Unix / Linux experience varied quite a bit, but experience tended to only impact their initial attempt, but not their final mode of access. The ease-of-use afforded by a graphical user interface was consistently less important to students than work flow issues. Especially important was the ability to save data so that in future sessions work can be quickly resumed. A few students in this observation pool were non-traditional students with full time jobs. These students were constrained regarding how much time they could spend on campus using the school's laboratory computers. However, the need to use their own computer to complete the assignments while mostly off-campus did influence their technology choices.

OUR COMPUTING ENVIRONMENT

The computing environment at K-State at Salina is quite typical of small under-graduate college settings. The campus has multiple computing laboratories available for student use. The laboratories used for most student course work are equipped with late model desktop computers running up-to-date operating systems from the Microsoft Windows family of operating systems. These lab machines are protected from unauthorized software installation and configuration changes by a product called Deep Freeze.[2] Whenever a user logs out, Deep Freeze deletes files added and restores any files that were changed or deleted by the user.

There is also a networking lab, which is behind a very restrictive firewall. This lab is used by the computer networking and systems administration classes that need administrative privileges. Several of the computers in this lab have removable hard drive systems, which allows students in certain classes to have their own hard drive to use to complete the class assignments. There is also one server grade computer, which runs Linux, available to students in certain classes via remote SSH connection. A few students and faculty have Linux or Apple Macintosh computers for their own use, but most students use Windows exclusively. The servers supporting the campus information technology needs use a mixture of Windows, Linux and Sun Solaris operating systems.

VIRTUALIZAION AND TRADITIONAL TECHNOLOGIES

In simple terms, **virtualization** means the simultaneous execution of more than one operating system (OS) instance on a single computer. This differs from traditional multi-programming, which is just multiple processes executing at the same time. There are several advantages to simultaneous execution of multiple operating systems. Today's computers are so fast that the CPU of most servers are under-utilized if it is running a single OS, even with multiple services running. Yet at the same time, services are increasing complex in terms of their software configuration, so there is a push to run only one service on each OS instance. With virtualization, multiple, single service OS instances can run from a single computer. It is also possible to migrate these modular OS instances from one computer to another to support maintenance activities.

The available vitalization software products can be grouped into two technology categories. The simpler model uses a host operating system and the virtual OS instances, also called virtual machines (VM), run as an user level processes of the host operating system. The second, newer architecture relies on technology called Hypervisor to eliminate the host operating system. In the later architecture, there is a small piece of operating system like software that controls the immediate access of each virtual machine to the hardware, but it is not a full operating system and uses only a small amount of the available memory and CPU time. With Hypervisor technology, the collective performance of the virtual machines is improved and better features are available to manage the virtual machines. The first, and most well know, Hypervisor product is VMWare's ESX Server.[8]

Hypervisor technology is preferred for network servers run by large organizations. We are interested in teaching students how to work with the products using Hypervisor technology. However, from an education perspective, the older, hosted technology is actually preferred in most cases. This is primarily due to security concerns, but the administrative issues are also simpler when a host operating system is used. When the virtual machines are run using a host operating system, the whole VM has no more privileges than any other user level process. Thus, even if the user is logged in as root (super-user) using a VM, they have no additional capability to engage in malicious activities or to do harm to the computer or the network.

A number of host based virtualization products are available. From the perspective of the user, many

of the products provide very similar capabilities. Our criteria for use was that it had to be free and that it had to allow Linux to run as a virtual machine. We used the VMWare Player product to run Linux using Windows as the host computer. We also used User Mode Linux (UML), which is an open source project, to run a Linux VM with Linux as the host operating system. VMWare Player was run using the laboratory computers while the use of UML used a remote SSH connection to a Linux server. As described below, these two products offer opposite ends of the spectrum in terms of ease-of-use and flexibility. Other available software products offer similar capabilities to these two products.

VMWare Player

VMWare Player is a free, desktop application for running a virtual machine.[7][8] The user has full access to the graphical desktop environment of the VM. The virtual disk drive of the VM is a file on the host computer. This file is an image of a drive containing a Linux file system. VMWare's web site contains a number of virtual machine images that be downloaded for immediate use. With the use of the VMWare Server product, which is also free, a custom image can be created. A virtual private network is created between the host computer and the VM. Thus, using the host computer as a router between the virtual network and the computer's network interface, the VM has full access to the network. Other computer drives, such the CD/DVD player and removable USB drives, are also accessable to the VM. Users have the option of logging in as a regular user or as a super-user.

The primary advantages of VMWare Player for education are that it runs locally on the lab Windows computers and that students have access to the graphical user interface of the Linux VM. Students with limited Unix / Linux experience were attracted to the graphical interface, especially the graphical text editors.

In our environment, we did have the disadvantage of not being able to permanently save data to the disk drive image. The Deep Freeze program restores the disk image after the student logs out. Students are accustomed to saving all of their work on USB based removable storage drives. However, in some cases, students may have to make configuration changes to the Linux environment in order to complete assignments. Thus, assignments that can not be completed in one session required repetition of some configuration changes when the student returned to continue working on an assignment. Printing from within the VM is also not possible. To print a file, the student would need to either transfer the file to the host computer or save it to their removable drive to print later.

User Mode Linux

User mode Linux (UML) is the implementation of the Linux kernel as an user level Linux process. [3][6] Thus, unlike most visualization products, the UML system calls are not trapped by the virtualization software and translated to modified native OS system calls. Because the UML virtual machines run as native Linux processes they have very good performance. However, there are more steps which must be completed before the UML may be used. As with VMWare Player, a disk image of the Linux system must be created. Additionally, a custom Linux kernel must be configured and compiled from the Linux kernel source code. If the UML is run within the Linux desktop environment, it is possible to use the graphical environment of the VM; however, in our environment, which accessed the host computer via a SSH remote connection from the Windows based computers, users were restricted to a command line shell environment. Despite the additional effort required to get started using the UML and the lack of a graphical user interface, UML offers some advantages that make it particularly appealing for many tasks.

Copy on Write (COW) Files

When the UML is started, multiple disk image files may be specified along with the mount points for each. Moreover, for each image file a copy-on-write (COW) file may also be specified. Using COW files, the disk image files can be read-only, with any changes made to the file system being recorded in the COW file. The file system presented to the user, appear to function as any other file system, but any changes made to the file system are recorded only to the COW file. Thus, each student starts with an identical file system, yet the changes made from session to session are re-applied each time UML is started. Students may also start multiple UML instances with each machine using its own COW file and thus having an unique configuration. If mistakes are made to where it would be easier to start over than to correct previous mistakes, the user can shutdown the UML instance, delete the COW file and start over with a fresh COW file. It is also possible to run commands from the host Linux system to merge a read-only disk image with a COW file. This is a convenient way for system administration staff to test new software or configuration changes. If the changes are acceptable, the file system changes from the COW file may be merged. In an educational setting; however, merging images would not often be needed.

Virtual Network Configuration

In addition to being able to run distinct UML virtual machines with unique configurations, each VM may also use a flexible combination of network interfaces. Rather than starting each VM with a standard, pre-configured virtual network interface, a virtual device called a TUN/TAP, which is effectively a virtual Ethernet cable, is created before the UML is started.[1] In the simplest configuration, one end of the TUN/TAP might be connected to the host system and the other end connected to a VM. Another program, which is a virtual network switch, may also be started. So, a student might build a virtual network using a switch connected to the host system and several virtual machines. One VM might also have multiple network connections and function as a router or firewall between different sub-nets. Thus, the advanced UML user can configure an endless combination of virtual networks and computers to test a configuration in a completely safe, virtual environment.

SSH Client (Putty)

Putty is a simple Windows application that provides a basic console and SSH client for remote connections to Unix / Linux based computers running a *sshd* server.[5] With *Putty*, students are able to use a Windows based computer to access Linux using the command line interface. One advantage of *Putty* to students on our campus is simply that it is a Windows application which can use the Windows clipboard to facilitate copying and pasting data between Windows applications. How students use this feature is discussed in the description of the *Introduction to Unix* class below.

Individual Workstation Installation

Another option is to install Linux on individual lab computers. In years past, this was done by configuring the boot manager of the computer to provide a menu for selection of booting either Windows or Linux. The advantage of duel booting is that it provides a way to use a basic lab computer for use with either Windows or Linux. The Linux user also has the benefit of the Linux desktop graphical environment. However, the duel boot configuration has been discontinued on our campus in favor of the other technologies presented in this report. The most significant problem with duel booting is that it is impossible to protect the disk partitions belonging to the operating system not

1 A TUN/TAP is functionally similar to a named loopback interface or named pipe. A data structure is created in the kernel of the host OS and kernel code is used to allow inter-process communication via this device.

currently booted. Also, duel booting causes confusion for the less experienced students and the mechanism for switching between Windows and Linux is slow and cumbersome.

If a Linux desktop environment is needed to complete assignments, a better alternative is to use removable hard drives. These drives use locking trays with all connectors on the back side of the tray, so the user simply slides the drive into the tray, locks it and turns the computer on. For relatively small classes, each student can have their own drive for the semester. This way, the same computer can be used for both Windows and Linux. A compelling advantage of this approach is that the student can build their configuration as the semester progresses. This, of course, implies that the student has administrative privileges on their workstation, which requires that the computer be in a *Networking Laboratory*, which is behind a restrictive firewall.

CLASSES, ASSIGNMENT REQUIREMENTS AND TECHNOLOGIES USED

Three classes were required to use Linux to complete varied assignments. Here the nature of how Linux was used is described and which access technologies worked best is discussed.

Introduction to Unix Class

This class teaches students how to use Unix / Linux. The objective is to teach new skills, which are needed to work in the information technology field. Thus, the focus is on the command line shell interface. The only real advantage to using a graphical interface for this class is to use a graphical text editor, although students would still be expected to become familiar with a console based text editor.[2] Students only need user-level access to a Unix / Linux computer.

Early in the semester, most students in this class try *VMWare Player* because of the graphical interface, but when they understand that the graphical interface offers few advantages towards completing the assignments, most students find that *Putty* allows the most effective means of completing assignments. The assignments for this class are task oriented. The students are asked to complete a number of activities that are similar to the daily activities of a programmer or a systems administrator and make use of specific standard Unix utilities. Most individual activities can be completed in fifteen minutes or less, but to keep up with the course pace, students must complete three to four activities by the end of each class period. Students must turn in documentation showing that they have completed each activity. The documentation shows the commands they entered including the content of any shell scripts, which they wrote, and the output produced by the computer. Thus, the simplest work flow seems to be to open the assignment page with a Windows based text editor, such as *Notepad,* and use *Putty* to access the Linux system remotely. Once each activity is successfully completed, the student uses the Windows clipboard to copy from the *putty* console to the text editor to fill in the answer for each activity.

Operating Systems Class

Students in the *Operating Systems* class have used Linux to write, install and use Linux Kernel Modules with the objective of learning more about the internal working and data structures of a modern operating system.[1] As with the assignments for *Introduction to Unix*, a graphical user interface is

2 Students are introduced to the *vi* editor, which is the one editor that can always be found on a Unix like operating system. However, most students feel more comfortable using an editor where they are always in insert mode, so the simpler, but less functional editor *nano* is the text editor most often used by students. *Nano* is a clone of the *pico* text editor, which is used in the *pine* e-mail client. But because of licensing issues, *pine* and *pico* are not readily available for most Linux distributions, while *nano* is available as a standard package.

only a small benefit to the student. However, some students in the *Operating Systems* class have not previously taken *Introduction to Unix*, and thus were quite inexperienced regarding how to use Linux. To install the kernel modules, students need root level access to the Linux system, thus they were not able to simply use *Putty* by itself. For this activity, most students found that *Putty*, in combination with *User Mode Linux* (UML) was the simplest to use. The primary advantages of the *UML* over *VMWare Player* was the ability to save work completed using a COW file versus the requirements of copying all files to removable storage since *Deep Freeze* would restore the disk image to its previous state upon log-out. One variation, which might have made using VMWare Player easier to use for this class would have been to hold the entire disk image on removable storage. However, these disk images can be quite large, so the availability and price of larger (8 GB) USB removable drives, as well as the speed of the USB 2.0 interface, limited the appeal of this option. A few students that were quite uncertain about using Unix / Linux had an initial preference for *VMWare Player*, but when observing the productivity of students using *UML*, most of them eventually switched to *UML*. The exception to this was the non-traditional students with full-time jobs and their own laptop computers. These students took the extra steps of installing *VMWare Player* on their computer. For these students, the driving force of their choice was the need to work on assignments when not on-campus.

Unix Administration Class

This class is a continuation of *Introduction to Unix* and focuses on how to perform Unix / Linux systems administration tasks rather than just using Unix. These students were given the option of installing their own Linux system using a removable hard drive and all students have chosen to do so. Use of the *VMWare Player* in our public labs would not be practical for this class because of *Deep Freeze*. Modern Linux systems include a number of graphical administrative tools that were useful to the students, however, students also learn how to perform administrative tasks by modifying configuration files and issuing shell commands. *User Mode Linux* has also been successfully used in this class for certain projects. *UML* was most useful to this class for networking related projects, such as building firewall systems.

CONCLUSION

Virtualization technologies used in conjunction with traditional technologies offer appealing options to access Unix / Linux systems as needed to complete course assignments. Having multiple access technologies available is important because the nature of different assignments can significantly impact which technology is preferred. It is also important to note that for certain activities the traditional options may be preferred. Key factors which universally influence the overall ease of use relate more to storing work between sessions than to any convenience offered from a graphical user interface.

BIBLIOGRAPHY

[1] Bower, T., Using Linux Kernel Modules for Operating Systems Class Projects, *Proceeding of the 2006 ASEE Annual Conference (ASEE'06)*, 2006.

[2] Faronics, Deep Freeze Product Page, *http://www.faronics.com/html/deepfreeze.asp,* retrieved July 18, 2009.

[3] Dike, J., *User Mode Linux,* Upper Saddle River, New Jersey, Prentice Hall, 2006.

[4] Virtualization Meets Reality, *Network World*, Special Report, 2007.

[5] Tatham, Simon, Putty Web Page, *http://www.putty.org/,* retrieved July 18, 2009.

[6] UML Project Team, UML Project Home Page, *http://user-mode-linux.sourceforge.net/,* retrieved July 18, 2009.

[7] VMWare Corporation, VMWare Player Product Page, *http://www.vmware.com/products/player/,* retrieved July 18, 2009.

[8] Zimmer, Dennis. *VMware Server and VMware Player.* Sunny Edition, Maur, UK, 2006.