Full paper

Using virtualization in teaching in a software laboratory

Dr Albert van Aardt
Principal Academic Staff Member,
Northtec, Northland

avanaardt@northland.ac.nz

Mike Mossom
Senior Academic Staff Member,
Northtec, Northland

mmosom@northland.ac.nz

Abstract
Teaching software development in a computer laboratory with single workstations inevitably leads to problems: different papers have different software needs. For example, one paper may require students to build a database application, while another may require students to install an operating system. Clearly the work of the second group will negate all the work from the first. In this study the use of "virtual machines" using software to alleviate these problems will be described, as well as pointing out some lessons learned.

Keywords: Computing education, quality assured, virtualization, software.

1 Introduction

The use of virtualization to simplify the management of computer systems has received wide interest in the ICT industry in the last few years. Numerous studies have indicated that significant cost and time savings can be achieved using this "new" technology. The technology itself has been around since the 1970's, but it was only within the last four years or so that low / no cost packages became available, running on standard i386 Intel PC architectures. We are using virtualization on desktop computers and not on servers. Every student runs their desktop operating system as a virtual machine.

In the past we got around the problem of clashing requirements of academic papers by using removable disk caddies. These worked fine, but were clumsy and fairly fragile. As each student needed at least three different "machines" to work on, we ended up with a large number of hard drives to store and administer. Handing out the right disks to the right students took some time as well.
However, the caddy system had one major drawback: if a student managed to "break" the OS on that caddy, a time consuming process had to be undertaken to get everything back again. Depending on the configuration, this could typically take about two hours. Multiplied over 22 PC's = 44 hours. The whole idea of a software development paper is for students to try new things, and it is inevitable that they will "break" the operating system. In addition, the caddies are made from rigid plastic and prone to breaking. Therefore, even the caddy system was fairly cumbersome to manage.

This quality assured paper appeared at the *22nd Annual Conference of the National Advisory Committee on Computing Qualifications* (*NACCQ 2009), Auckland, New Zealand. Samuel Mann and Mike Lopez (Eds). Reproduction for academic, not-for profit purposes permitted provided this text is included. www.naccq.ac.nz*

2 What is "virtualization"?

Simply stated, virtualization is running a full Operating System (OS) plus all applications as a separate application on a host OS. As far as the host OS is concerned, the virtual machine is simply another application. The "virtual image" is simply a group of files inside a directory tree, that can be copied or moved as any other directory tree. As far as the guest OS is concerned, it is running natively on the hardware and is oblivious of the host. (See figure 1). IBM pioneered virtual machines on the IBM VM OS (Silberschatz, pp.80). It is also not widely known that Microsoft Windows95 used virtualisation to run DOS programs. (Microsoft).

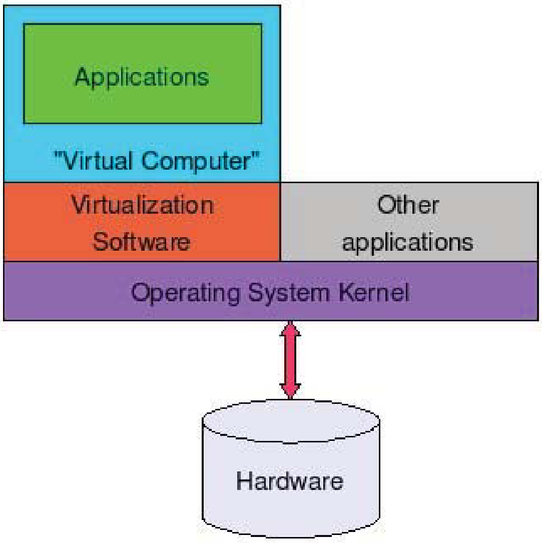


Figure 1: simple diagram illustrating the concept of virtualization.

3 Introduction

The computers in our software and network labs reached the end of their 4 year life cycle in 2007. We were using Dell Optiplex GX270 tower machines up to that point, but a number of them were failing, mainly due to faulty capacitors. Although we successfully repaired a number of these, it was felt that it was a good time to upgrade. After exhaustive testing, we settled on HP xw4600 quad core computers with 4 GB of RAM.
During December 2007 and January 2008 we conducted some small scale experimentation on free standing computers and gathered enough information on how to install and configure the software. We experimented with the Norton "Ghost" software as well as "G4U", but found these inadequate for our needs (see discussion on UUID below). We then wrote, tested and debugged an install script to semi-automate the process. The objective was to roll out the installation to a number of computers on the

LAN in the software laboratory. As all our client PCs are connected on our own, free-standing LAN, we have total control over our software environment.

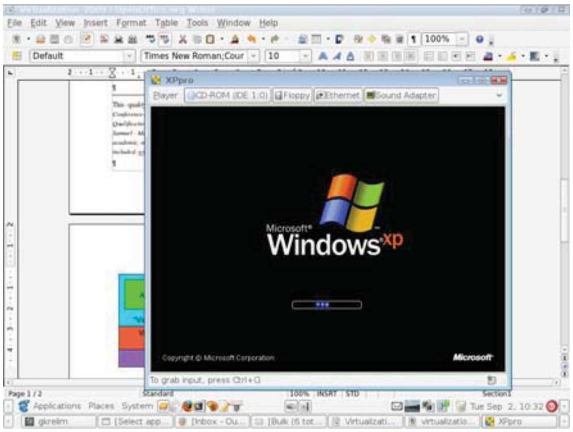

File Edit View Insert Format Table Tools Window Help
XPpro
Player
CD-ROM (IDE 1:0)
Floppy
Ethernet
Sound Adapter
Microsoft
Windows xp
Copyright © Microsoft Corporation
Microsoft
To grab input, press Ctrl+G
Applications Places System

Figure 2: Screenshot of Microsoft Windows XP ® booting up inside a virtual machine.

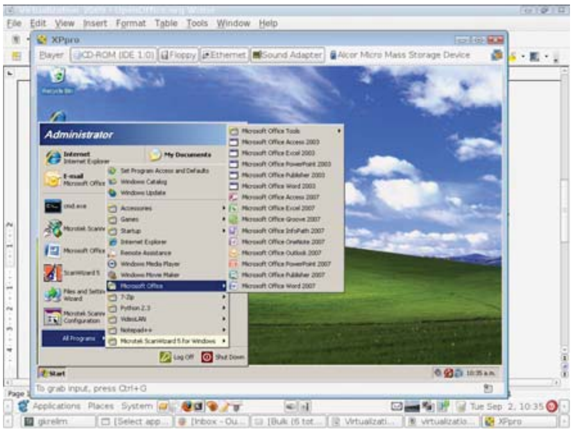

File Edit View Insert Format Table Tools Window Help
XPpro
Player
CD-ROM (IDE 1:0)
Floppy
Ethernet
Sound Adapter
Alcor Micro Mass Storage Device
Administrator
Internet
E-mail
All Programs
Log Off
Shut Down
start
To grab input, press Ctrl+G
Applications Places System

Figure 3: Screenshot of Microsoft Windows XP ® running on Ubuntu Linux host.

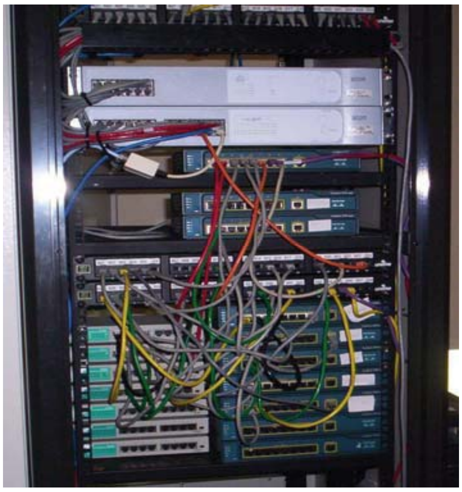


Figure 4: Network switches in Lab B160.

As budgets are always tight, one of the major objectives of the exercise was to limit expenditure to the minimum. More specifically, we believed we could achieve the software roll out without any direct costs.

3.1 Methodology

During 2007 we evaluated various virtualization software, notable VMWare (see "VMWare"), VirtualBox (VirtualBox) and Xen (Xen). We decided on VMWare as it was, at that time, the easiest to use and deploy.

During the last term of 2007 we created four virtual images, using the free VMWare Server software. These images contained:

a) "XPThin": A basic Microsoft XP Professional OS® with Microsoft Office 2007®

b) "XPFat": As above, but with the following software added:

i) Python, Pygame and Livewires;
ii) Notepad++
iii) Visual Studio.NET;
iv) "WAMP"; Apache, MySQL and PHP;
v) 7Zip
vi) GTK and the GIMP
vii) Firefox

c) "03Thin:

i) Microsoft Windows 2003 Server®
ii) Microsoft Office 2007®

d) "03Fat": The above plus:

i) Microsoft SQL Server ®
ii) PHP
iii) Firefox plus web development plug-ins
iv) Visual Studio.NET and ASP.NET

Each image was specified to grow to a maximum of 15 GB in 2GB chunks.

During the course of the year 2008 the installation script was refined and further tests were carried out. We also needed to adapt our VM image and re-install on a few occasions.

The procedure for 2008 was:

a) Install Ubuntu Linux 7.10 from the install CD on the host PC.

b) Copy the tar.gzip file from the network file server onto the host machine.

c) Run the install script.

d) Manually add the user “user” with the password “password” on the host when prompted

e) Continue install script, which finally copies the folder containing the Microsoft Windows XP ® virtual image from the LAN server onto the local host.

This semi-automatic process was followed mainly because Ubuntu allocates a Universally Unique Identifier (UUID) to hard drives, a somewhat controversial practice. (See the discussions at http://ubuntuforums.org/showthread.php?t=223182&highlight=uuid). This prevented us from creating one Ghost or G4U image of the whole hard drive to roll out over the LAN. However, in practice it was found that the total time needed to install using the semi-automatic script was roughly 45 minutes per machine. This was not considered excessive, as one could get more than one process running in parallel, resulting in being able to install all the software on all the machines in one morning.

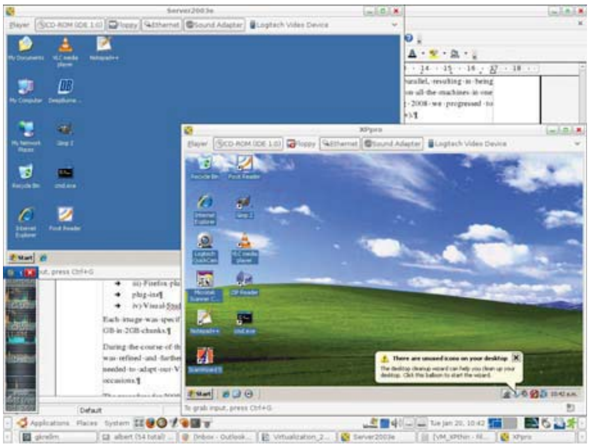
Nonetheless, during 2008 we progressed to network installations (see below).

3.2 Results

Due to administration and budgetary hold-ups, we only received the HP machine two weeks before the start of the first semester 2008. This placed tremendous strain on us to finalize the testing. This was completed the Friday before the start of the semester on the Monday. However, once the initial testing and bug-fixing was done, it took us less than four hours to roll the installation out on 22 PCs.

We found that students and staff had no problem adjusting to this "new" way of doing things, once properly explained. We organized a one hour session with all students and staff in the laboratory, and walked through the system with them.

The use of VMWare images for various papers proved to be very beneficial. In the past we had at least one or two cases every week where there were clashes between the various requirements of papers. In most cases these could be resolved with un-installing and re-installing software, but during 2006/7 we had to reformat hard drives to fix "broken" operating systems at least once a month. Using the VMWare images not only reduced this frequency, but creating a "new" image is as simply as copying the image from the backup directory on the host to the production directory; a mere six minutes process. In addition, students can run more than one image at the same time (see figure 5).



The desktop cleanup wizard can help you clean up your desktop. Click this balloon to start the wizard.

Figure 5: Running virtual images of Windows XP® and Windows Server 2003® on an Ubuntu Linux host.

4 Conclusion

A number of lessons was learned. First and foremost, ample time for testing should be set aside before rolling out. Each and every application installed must be thoroughly tested on the image, and this takes time. The host installation script must also be thoroughly tested; and each time a test must be run on a "clean" machine so as to avoid being tripped up by packages left over from a previous test. All this takes time -- doing it in two weeks is not a good idea.

A bonus benefit we encountered was that Microsoft DirectX® does not work on VMPlayer. This meant that students could NOT install their favorite games and waste time playing games.

One problem we did encounter is that VMPlayer mounts USB memory sticks as USB 1.1, rather than USB 2.0. After a lot of searching, it was found that the **.vmx** file needed to have the following lines appended to it. This solved the problem; nonetheless it took a lot of time.

```
scsi0.pciSlotNumber = "18"
ehci.present = "TRUE"
ehci.pciSlotNumber = "19"
```

In 2009 we issued all students their own portable USB hard drives. On these students copied one or more virtual images, and simply switched to the particular one as needed. All students also have administration rights on the virtual machines.

For managing the host OS in the lab, we started using "SystemImager" from December 2008. This package allowed us to stop using the CD plus install script and rather roll out hosts from a central server. Performance is impressive: 6 minutes to create a backup from a "Golden Client" (the PC with the all the host software installed) and 8 to 12 minutes to restore a client from the server. We have also upgraded our Ethernet switches from 10/100 Mb/s to 1 Gb/s speed. This facilitates the replacement of "broken" images and reduces the time taken.

As far as virtual images are concerned, we started with a clean image and let senior students install the software they'll be using themselves. We believe that his would be a good exercise to give them some very valuable practical experience. We also believe that this will teach them some invaluable lessons with regards to system security and backups.

One problem we did experience is that the SystemsImager software does not create the mount point for the DVD/CD on the image. We had to manually create this on each PC with the command:

```
sudo mkdir /media/cdrom0 /media/floppy0
```

We will keep on exploring the use of virtualization. For example, recent developments in VirtualBox are very promising (being released as Open Source) and we may move from VMWare to VirtualBox.

5 References

Microsoft: http://technet.microsoft.com/en-us/library/cc751120.aspx Accessed 20th January 2009

Silberschatz, A., Galvin, P.B., Gagne, G. 2006: "Operating systems Concepts"Wiley & Sons New York ISBN 0-471-25060-0

SystemImager: **http://wiki.systemimager.org/index.php/HOWTO_Net_Boot_Ubuntu**
Accessed 20th January 2009

VMWare: http://www.vmware.com/
Accessed 29th December 2008.

VirtualBox: http://www.virtualbox.org/
Accessed 29th December 2008.

WAMP: http://www.wampserver.com/en/
Accessed 29th December 2008.

Xen: http://www.xen.org
Accessed 29th December 2008.

Appendix.

<u>Using "Systems Imager" to roll out Ubuntu Linux clients in a software laboratory.</u>

1. Install software on both server and client.

2. Replace "relatime" with "defaults" in /etc/fstab on the client.

3. Check ip-address of server – set MANUALLY to fixed IP address of 172.20.100.xxx (e.g. 253) if necessary.

4. Prepare client :

```
si_prepareclient -–server
172.20.100.253
```

5. Get Image on the server (Where xxx is the IP address of the server)

```
si_getimage –golden-client 172.20.100.xxx –
image image-name –ip-assignment dhcp
```

(This will take about 5 minutes or so, depending on the size of the footprint on the Golden Client)

6. Create a boot CD ISO image

```
si_mkautoinstallcd –append
"IMAGESERVER=172.20.100.253
IMAGENAME=image-name" –-out-file
/home/image-name.iso
```

7. Burn the ISO image to a CD.

---

Restore Image on a client.

1. Check ip-address of server – set MANUALLY to 172.20.100.253 if necessary.

2. Start rsyncd on the server

```
/etc/init.d/systemimager-server-rsyncd start
```

3. Check the boot sequence on the client in the BIOS and enable booting from a CD.

4. Insert boot CD into new client. Start the computer. After about 8 minutes the new client will have been created. Remove CD and reboot.

5. Replace "defaults" with "relatime" in /etc/fstab

6. Stop rsyncd on the server

```
/etc/init.d/systemimager-server-rsyncd stop
```