AC 2007-1986: INCREASING IT LABORATORY FLEXIBILITY USING PORTABLE HARD DRIVES

Michael Bailey, Brigham Young University

Michael Moore, Brigham Young University

Joseph Ekstrom, Brigham Young University

Increasing IT Laboratory Flexibility Using Portable Hard Drives

Abstract

IT students have unique workstation requirements that include complete control of a computer and its configuration, resulting in setups that are often incompatible with other uses of the lab. For example, the system integration and administration thrusts of the IT curricula require that a student be trained week after week on his own system, with his own software installation, configuration and log files. However, few institutions can afford space and hardware to provide students with individual systems. Possible solutions include requiring students to own laptops, creating individual virtual servers on a powerful hardware system, or more recently, to use portable hard drives.

The use of small, portable hard drives for dedicated system instruction was tested this year in a Web-systems course. Students in a study group were provided with portable drives and a second (control) group had notebook computers with large drives, a partition of which was dedicated to the IT coursework. The students compared these options in terms of speed, reliability, ease of use, and in the case of the drives, portability between host computers. Other comparisons such as weight and cost are readily apparent, but were evaluated according to their importance to the students. The study found that portable hard drives are an effective compromise between cost, flexible lab use, and performance.

Introduction

Educating future Information Technology (IT) practitioners can be a very costly endeavor for a university due to the expense of obtaining, updating and maintaining computer hardware for student practicum. Typical IT laboratory curriculum includes networking, web programming, database development, information assurance and security, and system administration, all of which may have unique system and configuration requirements. Many IT programs do not have sufficient space and computational resources to dedicate laboratory rooms to individual topics, much less to dedicate computers to individual students within those labs.

These concerns are universal among educators, and have been addressed with varying degrees of success through various means. Some institutions or programs have required students to in effect, provide their own development systems in the form of a laptop computer. As stated by Campbell[1], “Budget conscious universities are realizing that requiring students to purchase laptops can reduce the need for expensive multimedia classrooms.” The pervasiveness of this practice can be found quickly by means of an internet search engine. While this approach is available, it puts an additional strain on limited student finances.

An approach to laboratory flexibility that has been used at this and other institutions has been to use an imaging server to completely overwrite laboratory computer hard drives between lab sessions, to customize the systems to the needs of the next lab section.[2,3] This approach does provide true laboratory flexibility, at the cost of administrative complexity, time for re-imaging disks, increased network load, and possibly expensive storage area network equipment.

One other approach to this problem utilizes virtualization. Most IT curriculums use hardware-level virtualizations, such as vmWare or Virtual PC, to allow multiple operating systems and software configurations to run concurrently on a single system. These are effective for their intended uses, but performance degrades significantly with each additional virtualization running concurrently. Operating system-level virtualizations, such as Virtuozzo, avoid the performance degradation associated with hardware-level virtualizations, allowing a system to efficiently run over one hundred virtual systems. This done by sharing common images and resources between virtualizations while maintaining the illusion, within each virtualization, that the system is stand-alone.

The operating system-level virtualization technique has been used successfully at this institution for a Web-systems course for the past 2 years. For this course, a large server running Linux, Apache, mySQL, Perl and PHP (LAMP) has been configured to run 100 virtualizations simultaneously. Each virtualization is assigned a unique IP address, so that students are able to log in and upload or edit files on their 'own' server. Port 80 on each IP address is available for viewing web pages. This type of configuration has several advantages, among them being the fact that this is identical to the setup of many LAMP servers provided by commercial web-hosting services.

However, these servers are less ideal in other aspects. First, the students are unable to get the experience of installing their own application packages such as mySQL, since for efficiency they must be pre-installed for all virtualizations. In addition, the virtualization technology does not work well for simultaneous operation of virtualizations that are running highly RAM-intensive applications such as the Apache Tomcat web server.

In view of the limitations of these laboratory technologies, a new approach was developed: requiring students to have and use bootable portable hard drives. These drives allow students to take their development system with them into the lab, or even to do their assignments at home, anywhere there is a computer that can boot from a USB 2.0 port. This solution is more economical for students that requiring them to purchase laptops, since such drives are easily obtained for less than the cost of a textbook. Unlike disk imaging systems, this approach requires no additional capital hardware or administration load on the department, and also doesn't require download latency before beginning to work on an assignment. Finally, portable drives allow students to learn through completely administering their own systems, unlike operating system-level virtual servers.

Objectives

Going in to this project, there were several development objectives that we strove to meet. These objectives were determined by the associated web development course, but the research performed has a much broader applicability.

One of the most challenging aspects of developing a software configuration for these hard drives is a goal to have the drive configured so that it could boot any modern computer with no more effort that selecting it as the boot drive from a BIOS menu.

To support web development, the system developed needs to support a variety of development, client, and server software. To accomplish this, the drives should run both Windows and Linux. This could be accomplished through dual-boot partitions, but would preferably be done using hardware-level virtualization so that both operating systems could be operational simultaneously. Such simultaneous operation allows the student to use the host OS as a server system for testing, and the virtual OS as the client system, as shown in the rightmost system in Figure 1.

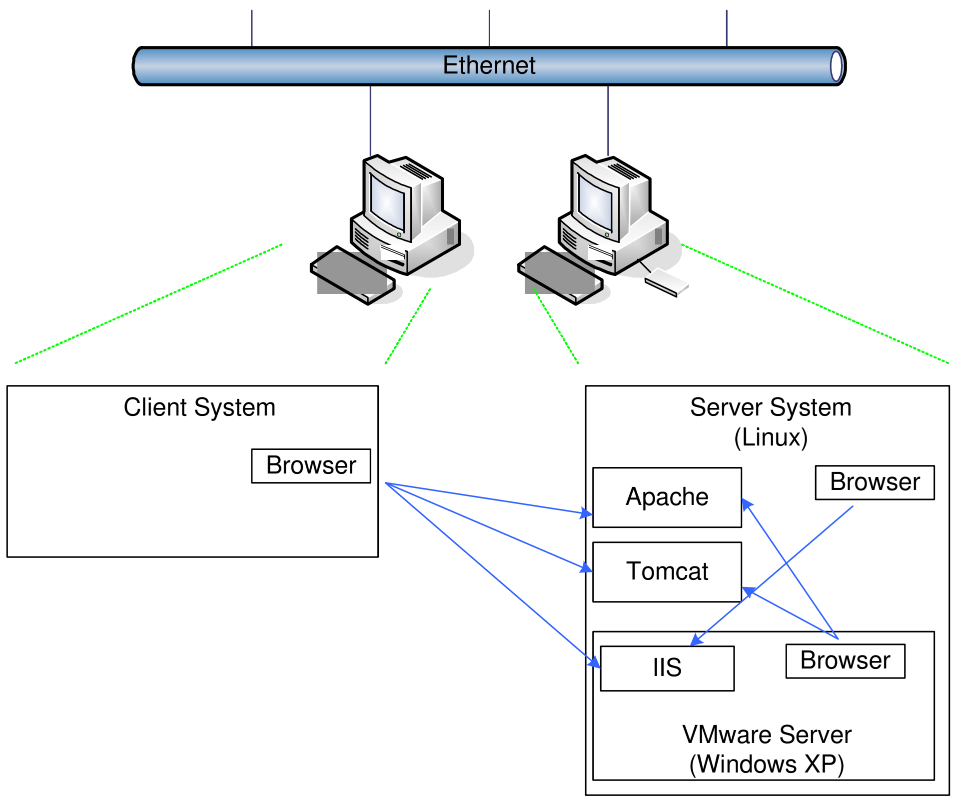

Figure 1. Internal and External Client-Server Configuration

Installation of the software configuration should be fairly straight-forward, since if each student were required to develop a portable, bootable external drive with a virtual machine and two operating systems, there would be much less lab time available for the other web development course objectives.

Finally, this research examines many of the tradeoffs associated with portable hard drive technologies, as compared to other approaches to the limited-lab problem. The tradeoffs examined are system performance, drive durability, size and weight and ease of use.

Results

As is good practice, the highest risk aspects of the project were investigated first. To begin with, this was to determine the plausibility of a portable, bootable hard disk. Setting up an external drive to boot from is not difficult, but this project required that the disk be mobile to various systems, and easily boot them all. The operating systems of interest to the project were Windows XP and Linux, so these were both investigated for suitability.

When attempting to install Windows XP on a USB drive, problems immediately arose, when the installer halted before loading the OS. A call to Microsoft Technical Support revealed that installing to an external disk is purposely disabled. A moment's reflection brought the realization that boot drive portability would be incompatible with Microsoft's copy protection scheme, which attempts to uniquely identify the hardware on which a given copy of the OS is installed, and to prevent it being installed on other hardware. In view of this, Microsoft's decision to disable its installer from loading Windows XP onto an external drive is logical.

Linux, on the other hand, has no such licensing issues. For this research, experiments with downloading various Linux distributions to portable hard drives were performed, with Ubuntu Linux 6.10 working the best of those tried. The main criterion for this judgment was the ability of the boot loader to be placed on the portable disk, and for the boot loader to be configured to boot from a partition on that portable disk. This set of capabilities is included in GRUB 0.97, a boot loader that is included in the Ubuntu distribution. GRUB typically refers to disks by their BIOS name. Since these disks are intended to be portable and to be booted on machines with many different BIOS configurations, this does not work. However, the GRUB configuration file can be modified to refer to the boot disk by it's UUID, a unique number assigned when the disk was formatted. Other new distributions with the same boot loader, such as Suse, were also examined, but proved more difficult to configure than Ubuntu. Please refer to the installation instructions in the appendix for an example of the procedure used.

Using this procedure, a portable USB drive was able to be formatted to allow it to boot an arbitrary computer in Linux, by simply plugging the drive into a USB 2.0 port, powering up the computer, and during power-up selecting the proper drive from the BIOS boot menu. An additional issue was quickly discovered when testing the boot-ability of the drive on a wide variety of machines. Some of the test machines had older, lower-resolution video displays than those in use on the development systems. Thanks to the open nature of Linux we were able to study the code from the Ubuntu Live CD and discover the command flags that the Live CD uses to surmount this problem, and to incorporate them into our scripts. The Live CD uses the command "dpkg-reconfigure -fnoninteractive --no-reload -phigh xserver-xorg" to tell the dpkg package manager to automatically set what it thinks are the best settings for the current video hardware. These flags were added to the Gnome Desktop Manager startup script.

Using the configuration that was thus developed, the disk has been tested on a variety of systems from high-end workstations to presentation stations with video projectors, to notebook computers, and even to Tablet PC's. In all cases, as long as the computer supported USB 2.0, the drive was able to boot the system in a usable mode with little effort.

In spite of the early setback installing Windows XP as a bootable OS, the desire remained to have Windows operational on the system. Since a bootable partition was not achievable, a virtual machine solution was sought instead. That is, a VMware Server image with Windows XP installed as its operating system was developed and placed on the portable drive, to run from within Linux. Due to our department's Microsoft Academic Alliance membership, licensing issues did not arise.

The final testing that was performed was to assure that the portable disk image that had been developed would be completely suitable for the Web Development course. To do this, Apache and Tomcat were installed as Linux-based web servers, and Microsoft's IIS was installed as a Windows based web server. Using this configuration, it was shown that a single system can be used as both client and server for web testing and development, as was shown in Figure 1, or an external client can be used to probe both the Windows and Linux web servers, as was also shown in the figure.

In addition to functionality, the speed of the portable hard drive was also tested. USB 2.0 isn't as fast as an internal hard drive connection such as ATA, and this interface is the most obvious bottleneck in an externally booted system. USB 2.0 is rated at 60 MB/s, while the Serial ATA drives common in new desktop computers can reach at least 150 MB/s, depending on which version of SATA is being used. While it seems that this speed difference should severely affect system performance, the only time that testers noted a difference was during boot time, probably due to caching. To confirm this observation, a quantitative test was performed on a typical 2 GHz Dell notebook computer with 1 GB of RAM installed. When booting from USB instead of a notebook computer internal hard disk, a 26% time penalty was observed (53 seconds versus 42 seconds). While disk-intensive applications would be slowed likewise, applications typical to web development and testing showed no such effects. Even while running Windows via VMWare, and the Apache 2 and Tomcat web servers, the user interface continued to respond quickly, allowing students to work on the web page and database they were testing.

Portable hard drives are designed to be durable, since they are designed to be placed in much-abused notebook computers. While the sample size of drives used in this study was not large enough for a statistical analysis, one student damaged his drive by dropping the backpack that it was in from a moderate height. Others have reported dropping the drives with no ill effects. Dropping any electronic equipment, including notebook computers, is risky, so durability remains a concern.

In addition to durability, the compact size and weight of the drives make them attractive for student application. A typical, inexpensive model used by many of our students measures 0.6 x 3.1 x 6.4 inches, and weighs just 7 ounces. These drives require no power supply, operating off of what the USB bus supplies.

Other Issues

One area of concern in the external hard-drive boot scenario is authentication and security. In our previous laboratory environment, we imaged the lab machines so that students logged into the machines using their ID, as managed by the College of Engineering and Technology. The customized login process then created a local user with admin privileges, providing us a binding between the student and the IP address for network traceability purposes. Since the students control the images on the external drive, there is now no way to guarantee that we know who is using the physical hardware in the lab, and no way to trace network activity back to a student ID. Since traceability is a university requirement, in the long term we must find a solution to this problem.

The issue is the ability to bind a network address to a student. We have been using the end station to perform the binding and this is no longer possible. We do control the switches that are in the labs, and could therefore force authentication to gain access to the network. Since there is physical access control to the lab, we are not concerned about use of the machines, but only traceability of a network address. We are investigating using 802.1x on the switches and forcing the students to supply credentials for network access. Another approach would be to allow access to the LAN without authentication and to implement authentication on the gateway routers that control access between our labs and the university-controlled infrastructure.

Conclusions

With the advent of inexpensive, compact, durable, bus-powered external drives, it is now possible to provide each student with a portable environment that can be easily moved from computer to computer. This portability provides increased laboratory flexibility for the institution, and develop-anywhere convenience for the student. While notebook computers provide many of the same advantages, portable drives are an affordable alternative for financially challenged individuals.

Issues regarding the portability of the drive, or more explicitly of allowing the disk to seamlessly boot most computers were addressed, including specifying the external drive boot partition unambiguously for a boot loader as well as auto-selecting the proper video driver for a system. User network authentication within the university lab has not yet been addressed, but research is in process.

Bibliography

1. Campbell, A., Pargas, R., "Laptops in the Classroom", *Proceedings of SIGCSE'03*, Feb. 2003, pp. 98-102
2. Leone, J., Hartpence, B., Weber, B., "A Networking and System Administration Laboratory Infrastructure", *Proceedings of CITC3*, Rochester, NY, 2002, session 261
3. Higby, C., Blackham, N., Rogers, B., Bailey, M., "Re-imaging Computers for Multi-purpose Labs", *Computers in Education Journal*, v 15, no. 3, 2005, pp. 34-40
4. Tarnoff, D., "Shifting Students' Financial Responsibilities form Textbooks to Laboratory Resources", *Journal of Computing Sciences in Colleges,* v 22, no. 3, January 2007, pp. 237-243

Appendix: Creating a bootable portable drive

Materials used:

- HP Compaq dc7600 computer
- Western Digital 80 GB USB laptop hard drive (bus powered)
- Ubuntu Edgy (6.10) Alternative Install CD
    - The CD can be downloaded in ISO format from ubuntu.com

Overview:
We will use the Ubuntu Alternative Install CD to do an OEM install to an external hard drive. We will configure the install, and use the disk as an image to quickly configure student drives. After prepping an OEM install, a command is run so that on next boot the user is prompted to create a new username and password. This will allow us to avoid issues of having a default username and password.

Terms:

- Host computer: The computer which is used to boot the USB disk.
- GRUB: *GRand Unified Bootloader* – The default boot loader used by Ubuntu
- Ubuntu: A popular desktop Linux distribution based on Debian. http://www.ubuntu.com/
- UUID: *Universally Unique Identifier*—A long HEX number used to uniquely identify partitions or other hardware.
- GDM: *Gnome Display Manager* The login manager used by Ubuntu
- X, X11, Xorg, xserver: The graphics layer used by Linux.

Procedures:

1. Remove all hard drives from the host computer.
    a. This can be done by either physically removing the drives from the computer, or by completely disabling them in the BIOS. Disabling them in BIOS works on most machines, especially new ones, but is not guaranteed to work. Removing the drive is recommended to minimize risks such as overwriting the internal Hard Drive's MBR.
2. Boot from the Ubuntu Alternative Install CD, with the USB drive connected
    a. At the boot prompt, choose "Install in OEM mode" and press ENTER
3. Install process:
    a. Navigation in the install process is done using the arrow and tab keys. Selections can be made using the space bar. You can accept the default values for most steps, with the following exceptions:
    b. Choose to manually partition the disk
        i. Delete any existing partitions
        ii. Create a 20 GB partition for Linux.
            1. The filesystem should be ext3, it should be bootable, and it should be mounted on /.
        iii. Create a 2 GB swap partition
        iv. Leave the rest of the disk space free (this will be explained in the imaging conclusion.)
    c. Choose a password for the OEM user. This will be used during the image customization

   d. When the installer is done, you will be prompted to shutdown or reboot. Select reboot.
4. Upon reboot from the external HD you may have to use one or more of the following methods:
   a. Use a BIOS hotkey to choose your boot device
   b. Enter into the BIOS, and change the boot device order to have USB before the first hard drive.
   c. Note, there are a few computers that will not allow a USB boot period. These are typically older models that do not support USB 2.0.
5. When you are given a login screen, use the username *OEM* and the password you created in the installer.
6. Customizing the install:
   a. This step is required for functionality!
      i. Edit the file `/boot/grub/menu.lst` as superuser (using sudo).
      ii. Find the line in the menu.lst file that says "`kopt=root=UUID=(Long HEX number here)`". Copy from "`root=UUID=`" through the end of the HEX number.
      iii. Replace all instances of `root=/dev/sda1` (where sda1 is the partition you installed to) and replace them with the "`root=HEX-NUMBER`" line.
         1. This can be done in vim or any other editor using a find/replace command.
      iv. This will allow grub to find the correct partition regardless of what the drive is numbered by BIOS.
   b. This step is required for functionality!
      i. Edit the GDM startup script at `/etc/init.d/gdm` as superuser (using sudo)
      ii. Find the line that says "start)"
      iii. Create a new line below the "start)" line and put:

```
dpkg-reconfigure -fnoninteractive --no-reload -phigh xserver-xorg
```

      iv. This line will cause the graphics card to be automatically detected at each boot.
   c. This step is highly recommended!
      i. Edit the package manager source list `/etc/apt/sources.list` as superuser (using sudo)
      ii. Delete all lines, and replace them with:

```
# Default /etc/apt/sources.list file for it210
deb http://us.archive.ubuntu.com/ubuntu/ edgy-updates main restricted
universe multiverse
deb http://us.archive.ubuntu.com/ubuntu/ edgy main restricted universe
multiverse
deb http://security.ubuntu.com/ubuntu edgy-security main restricted
universe multiverse
```

      iii. This will allow the students to more easily install the dependencies for the servers they will need.
   d. This step is highly recommended!
      i. In a terminal, run the following as super user `sudo apt-get update && sudo apt-get dist-upgrade && sudo apt-get upgrade`

    ii. Press *Y*, then *Enter* when prompted if you should install any packages
    iii. This updates the image with the latest security patches.
e. This step is optional
    i. Run the following commands in a terminal: `sudo apt-get install build-essential linux-headers-generic xinetd`
    ii. This installs the packages that VMWare will need to function. It isn't needed, but will make installing VMWare much easier for the students.
f. Install any other packages you want available to the students by default.
g. Finalize the image
    i. Run the following command as superuser: `sudo oem-config-prepare`
    ii. Once this command is run, you should not boot the image again. On next boot, it will prompt for a username, and the OEM user will be deleted.
h. Shutdown and remove the drive.

The drive is now ready to be used as an image. You can use it as a source for professional imaging software such as Altiris or Ghost, or you can use dd on a Linux live CD to copy the partitions. Using dd will not work if the destination drive has a smaller cylinder size than the image source drive.

The purpose of the empty space at the end of the drive is to make the image more portable. With the empty space, the image can be copied to a drive as small as 22 Gbs. The recommended use of the free space is to create a FAT32 partition on it after the imaging is complete. This FAT32 space can be used under Linux, Windows or Macintosh operating systems, allowing the user to easily move files around.