UNIVERSITY OF OSLO
Department of Informatics

Virtual Machines In Education

Muhammad Ashfaq
Oslo University College

May 23, 2007

Virtual Machines In Education

Muhammad Ashfaq
Oslo University College

May 23, 2007

Abstract

To provide education and particularly providing practical educational experiences to the students in the field of computing and information technology related courses including practical experience in the field of Networking, System Administration, and Operating Systems needs a lot of resources for the institution. Because this level of technical education can't be provided only theoretically, students also need hands-on practical experience, and providing practical experience faces a lot of problems such as lack of funding and physical space, risks and threats to the network environment when we attempt to provide real, physical laboratory for experiments. This problem can be solved by developing a virtual environment for delivering students practical education. In this report we will look into different technologies used for virtualization today and do a comparative study. We will also explore some of the institutions, which are using virtual machines based environment to provide students practical experience in the field of computing and Information Technology. And see how peoples are getting benefits from using virtual machines. We present how networks of virtual machines can be beneficiary for computing and information technology student and institutions by providing necessary environment in virtual network.

Contents

1 Introduction **3**
- 1.1 Virtual PC . . . 3
- 1.2 Virtualization . . . 3
- 1.3 Why Virtualization: Reasons . . . 4
- 1.4 Introductions to Virtual Machines . . . 6
- 1.5 Standardized Computer System Components . . . 9
- 1.6 Virtual Machine Basics . . . 11
- 1.7 Process VMs . . . 14
  - 1.7.1 Multiprogramming . . . 15
  - 1.7.2 Emulation and Dynamic Binary Translators . . . 15
  - 1.7.3 Dynamic Optimizers . . . 15
  - 1.7.4 High Level VMs: Complete Platform Independence . . . 16
- 1.8 System Virtual Machines . . . 16

2 Virtualization Technologies **18**
- 2.1 History of Virtual Machine . . . 18
- 2.2 Virtualization at the Hardware Abstraction Layer . . . 21
  - 2.2.1 VMWare . . . 21
  - 2.2.2 Virtual PC . . . 22
  - 2.2.3 Denali . . . 22
  - 2.2.4 Xen . . . 23
  - 2.2.5 Plex86 . . . 24
  - 2.2.6 User-mode Linux . . . 24

3 Benefits and challenges for virtual environment **25**
- 3.1 How Virtual environment can be better then real environment . . . 25
- 3.2 Benefits of Virtualization . . . 26
  - 3.2.1 Flexibility and agility . . . 26
  - 3.2.2 Server Consolidation . . . 26
  - 3.2.3 Business continuity and disaster recovery . . . 27
  - 3.2.4 Reduction in Downtime . . . 27
  - 3.2.5 Reduction in Administrative Costs . . . 27
- 3.3 Challenges in Managing a Virtual Environment . . . 27
  - 3.3.1 Bandwidth Implications . . . 28
  - 3.3.2 Policy Management . . . 28
  - 3.3.3 Image propagation . . . 29
  - 3.3.4 Security Considerations . . . 29

4 Literature survey **30**
4.1 Case Study: Using Virtual Machines for teaching System Administration 30
4.1.1 Inroduction 30
4.1.2 University Background Information 31
4.1.3 System Administration Courses 31
4.1.4 Their Vision 33
4.2 Case Study: Virtual Laboratory Usage in IT Security Education 35
4.2.1 Introduction 35
4.3 Case Study: An Open Source Virtual Lab used by The University of Milan 41
4.3.1 Introduction 41
4.3.2 Background for This System 43
4.3.3 Techniques for Virtualization 43
4.3.4 System Structure 44
4.3.5 Framework of Virtual Lab 45
4.3.6 Implementation of Virtual Lab 47
4.4 Case Study: Virtual Machine at Ume University Sweeden 48
4.4.1 Introduction 48
4.4.2 Problem 48
4.4.3 Agenda of new Assignment 49
4.4.4 Solution 49
4.4.5 Basic Virtual Machines Architecture types 49
4.5 Case Study: Virtual Machines at University College of Oslo and the University of Amsterdam 51
4.5.1 Introduction 51
4.5.2 User-mode Linux used at the University of Amsterdam 54
4.5.3 MLN 57
4.5.4 Advantages of MLN 58
4.5.5 MLN Usages 59

5 Conclusion **61**

Chapter 1

Introduction

1.1 Virtual PC

Virtual. It's a term we hear more often today. What does it really mean? It means that we experience something that is not truly real, but that it seems like it is. We may think we are walking across a rocky terrain on another planet with a "ray gun" in our hand looking for alien enemies, but we are really not. The system is setup to make it seem real. Sometimes it is important to have a computer for us own. Whether we need root access that we won't get on the official machines or that we need to work in a different environment in contrast to the environment our organization currently provide. It has been seen that, most applications do not need the full CPU, disc and memory capacity and performance of a dedicated machine. With virtual machines, separate machines can be set up on the same physical host. Usually, we don't see the difference between a physical host and a virtual one. Virtual hosts are quite easy to set up. They offer copy-on-write file systems that can restore a predefined state at any time. Currently virtual host are used for software testing, as servers for security related or otherwise dangerous services, for practical courses which require root access for students especially in the courses of networking, system administration, computer security, for creating larger test setups with 3 or more machines on different networks, and for several other purposes.

1.2 Virtualization

virtualization is a framework or methodology of dividing the resources of a computer into multiple execution environments, by applying one or more concepts or technologies such as hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, and many others. This is a loose definition because it includes concepts such as quality of service, which, even though being a separate field of study, is often used alongside virtualization. Often, such technologies come together in intricate ways to form interesting systems, one of whose properties is virtualization. In other words, the concept of virtualization is related to, or more appropriately synergistic with various paradigms. Even though we defined it as such, the term "Virtualization" is not always used for the concept of partitioning, breaking something down into multiple entities. Here is an example of its different meaning: we

can take N disks and make then appear as one logical disk trough a virtualization layer. Grid computing enables the virtualization of distributed computing: IT resources like storage, bandwidth, and CPU cycles.

PVM (parallel Virtual Machine) is a software package that permits a heterogeneous collection of UNIX and other Windows computer hooked together by a network to be used as a single large parallel computer. PVM are widely used in distributed computing.

1.3 Why Virtualization: Reasons

There are many possibly representative reasons for using virtualization concept and benefits of virtualization. We will discus them briefly.

- Mostly when we are working at home probably we are sitting at a computer. Our computer is a real, a physical machine. On that machine we have got an operating system and we have also got programs we have installed on it. We also have data. When this machine gets old or out-of-date we have to move all our stuff to a new computer. We have to reinstall all those programs. We have to move over all our data. We are doing all this simply because our equipment got old or failed. There is absolutely nothing wrong with the operating system or software. Virtualization separates the machine from the software. When we do that we open up all kinds of possibilities. When we can create "virtual machines" you can have several virtual computers on one physical machine. The virtual machines can be running different operating systems and can have different software running in them. If one virtual machine crashes, the others are just fine. With virtual machines if Dad wants to fiddle with Linux the whole family doesn't have to participate. He can set up a virtual machine and experiment all he wants. Through this virtualization gives us great flexibility.

- Because a virtual machine is contained in a few files it makes it easy to move those files to another physical machine. For example. One computer begins to wear out and I get a new one. Using virtualization I can easily move my virtual machines to the new computer. I can install VMware Workstation or the free VMware Player, or Parallels Workstation, or Microsoft Virtual Machine, and immediately I am back in business. I don't need to install any program, no data to move or transfer. Within minutes of having a new computer arrive I can have everything I need installed on it by way of virtualization. Using VMware Workstation I can clone an existing virtual machine. Then I can make changes to that one and not in the original. If the original machine has children, clone one machine for each child and let them customize it to their hearts content. No matter what they download or install, they are not affecting the real computer. When we want to have backup, we just backup the files that contain the virtual machines.

- To day virtual machines are widely used to unite the workloads of several underutilized servers into few machines, or perhaps to a single machine or server consolidation. There are many benefits for using virtual machines as consolidation.

Some of them are savings on hardware, environmental costs, management, and administration of the server infrastructure.

- It has been seen that virtual machines serve well to run legacy application. Some times it might be possible for legacy applications simply to not run on newer hardware or operating systems. Even if it does, it may under-utilize the server, so as above, it makes sense to consolidate several applications. This is simply not possible without virtualization as such applications are usually not written to co-exist within a single execution environment. One of the trivial examples of such type of applications is applications with hard-coded system V IPC keys.

- " To run un-trusted applications on real machines or servers can be very risky. Virtual machines can be used to provide secure, isolated sandboxes for running un-trusted applications. We could even create such an execution environment dynamically, as we download something from the Internet and run it. We can think of creative schemes, such as that involving address obfuscation. Therefore virtualization is considered to be an important concept in building secure computing platforms.

- With resource limitations Virtual machines can be used to create operating systems, or execution environments, and given the right schedulers and resource guarantees. Partitioning usually goes hand-in-hand with quality of service in the creation of QoS-enabled operating systems.

- Virtual machines can provide us the illusion of hardware, or hardware configuration that we do not have (such as SCSI devices, multiple processors,) to create network simulation of independent computers Virtualization can also be used.

- Some times we need to run multiple operating systems at same time. Virtual machines can also be used to run multiple operating systems simultaneously: different versions, or even entirely different systems. Some such systems may be hard or impossible to run on newer real hardware.

- Through Virtual machine monitor allows us powerful debugging and performance monitoring. We can put such tools in the virtual machine monitor, for example. Operating systems can be debugged without losing productivity and we can also set up more complicated debugging scenarios according to our desires

- As Virtual machines can fully isolate all applications running on, so it provide fault and error containment. To study the software subsequent behaviour we can inject faults proactively into software and this can only be done on virtual machines.

- With Virtual machines migration of software is very easy, which provides aiding application and system mobility.

- With virtual machines it is possible to categorise applications for example we can treat application suites as appliances by "packaging" and then running each package in a different virtual machine.

- There is a great use of virtual machines in research and academic experiments as they provide isolation and therefore it's safer to work with virtual machines in research projects and in academic experiments. They encapsulate the entire state of a running system: we can save the state, examine it, modify it, reload it, and so on. The state also provides an abstraction of the workload being run.
- To run operating systems on shared memory multiprocessors can also be achieved by Virtualization.
- It's very easy with virtual machines to create arbitrary test scenarios and this lead to some very imaginative, effective quality assurance.
- Virtualization can be used to add new technologies and features in existing operating systems without too much work.
- Some of the important tasks like system migration, backup, and recovery can be made very easier and more manageable with virtualization.
- Virtualization can be an effective means of providing binary compatibility.
- Virtualization on commodity hardware has been popular in co-located hosting. Many of the above benefits make such hosting secure and cost-effective.

1.4 Introductions to Virtual Machines

Modern computers are among the most advanced human-engineered structures, and they are possible only because of our ability to manage extreme complexity [1]. To day computer systems consist of many silicon chips and these chips consist of thousands of transistors. These transistors are then connected with input/output (I/O) devices and to the network to have a platform so that software can operate on it. Operating systems, application programs and libraries, and graphics and networking software all cooperate to provide a powerful environment for data management, education, communication, entertainment, and many other applications. The management of computer system complexity is done by its division into different separated interfaces. Through this abstraction all the details of design can be ignored, making possible to simplify the design of components at higher level. The details of hard disk, for example, that it is divided into sectors and tracks are abstracted by operating systems so that the disk appears to application software as a set of variable size files. An application programmer then can create, write, and read files, without knowledge of the way the hard disk is constructed and organized. The management of abstraction is arranged in such away that lower level implemented is done in hardware and higher level implemented is done in software level. In the hardware levels, all the components are physical, have real properties and their interfaces are defined so that the various part can be physically connected. In the software levels, components are logical, with fever restrictions based on physical characteristics. We are concerned with the abstraction levels that are at or near the hardware/software boundary. On these levels the software is kept completely hide from the machine on which it executes. We know that any type of software can only be executed on a piece of hardware or machine. We have to look machine from

two perspective, from the perspective of the operating system, a machine is largely composed of hardware, including one or more processors that run a specific instruction set, some real memory, and I/O devices. However, we do not restrict the use of the term machine to just the hardware components of computer. Where as from the perspective of application programs, the machine consist of an operating system and some portions of the hardware which are directly accessible through user-level binary instructions. We will look now the management of complexity from another perspective i.e. from the perspective of well defined interfaces. Well defined interfaces allow computer design tasks to be decoupled so that teams of hardware and software designers can work more or less independently. The instruction set is one such interface. For example, designers at Intel and AMD develop microprocessors that implement the Intel IA-32 instruction set, while software engineers at Microsoft develop compilers that map high-level languages to the same instruction set. As long as both groups satisfy the instruction set specification, compiled software will execute correctly on a machine incorporating an IA-32 microprocessors [1]. The second important interface in computer system is Operating system interface, which is defined as a set of function calls. As the Intel/Microsoft example suggests, well defined interfaces permit development of interacting computer subsystems at different companies and at different times. Application software developers do not need to be aware of the detailed changes inside the operating system, and hardware and software can be upgraded according to different schedules. Software can run on different platforms implementing the same instruction set. After the hardware and software interfaces, resource consideration of the hardware also limits the flexibility of software systems. Memory and I/O abstraction, both in high-level languages and in operating systems, have removed many hardware resource dependences; some still remain, however. As all the hardware resources are managed by only a single operating system. This limits all hardware resource management in a single regime. And this then automatically limits the flexibility of the system, from security and also from failure isolation point of view. Through virtualization constraints can be relaxed and flexibility can be increased. When a system or subsystem, e.g., a processor, memory, I/O device, is virtualized, its interface and all resources visible through the interface are mapped onto the interface and resources of a real system which actually implements it. Moreover the real system is transformed in such a way, that it appears to be a different, virtual system or even a set of multiple virtual systems. Virtualization deals with the construction of an isomorphism, whose responsibility is to maps a virtual guest system to a real host. The process of this isomorphism is shown in the Figure 1, which maps the guest state to the host state, and for a sequence of operations, e, that modifies the state in the guest there is a corresponding sequences of operations e' in the host that performs an equivalent modification to the host's state. Although such an isomorphism can be used to characterize abstraction as well as virtualization, we distinguish the two: virtualization differs from abstraction in that, virtualization does not necessarily hide details; the level of detail in a virtual system is often the same as that in the underlying real system.

Virtualization concept is not only limited for the sub system, but it can also be applied to the entire machine. A virtual machine (VM) is implemented by adding a layer of software to a real machine to support the desired virtual machine's architecture. For example, virtualization software installed on an Apple Macintosh can provide a Windows/IA-32 virtual machine capable of running PC application programs.

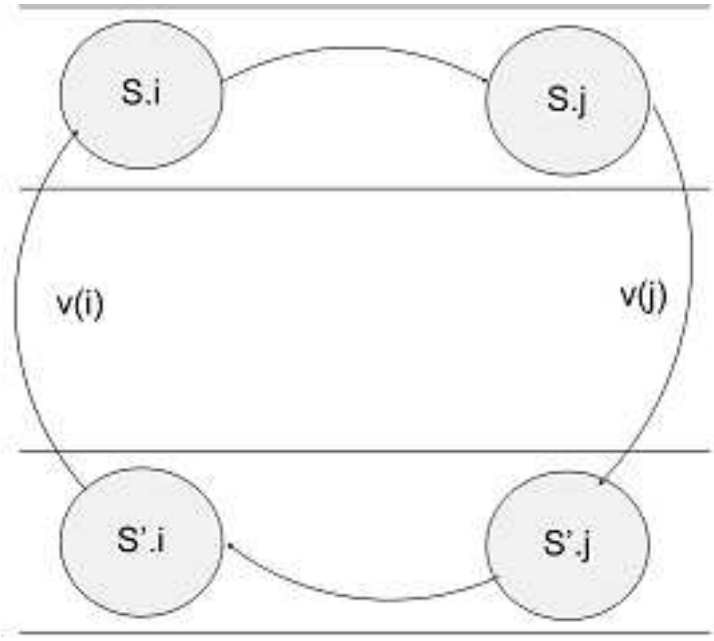

Figure 1.1: Virtualization is construction of an isomorphism between a guest system and a host

In general, a virtual machine can circumvent real machine compatibility constraints and hardware resource constraints to enable a higher degree of software portability and flexibility [1]. A very large number of virtual machines verities exist to provide an equally wide variety of benefits. Multiple, replicated virtual machines can be implemented on a single hardware platform to provide individuals or users groups with their own operating system environments. The different system environments possibly with different operating systems also provide isolation and enhanced security. A large multiprocessor server can be divided into smaller virtual servers, while retaining the ability to balance the use of hardware resources across the system. Cross platform software compatibility can also be supported by the virtual machine emulation techniques. For example, a platform implementing the PowerPC instruction set can be converted into a virtual platform running the IA-32 instruction set. Consequently, software written for one platform will run on the other. This compatibility can be provided either at the system level e.g. to run a Windows OS on a Macintosh or at the program or process level e.g. to run Excel on a Sun Solaris platform. In addition to emulation, virtual machines can provide dynamic, on-the-fly optimization f program binaries [1]. The example of the virtual machine that is described is considered to mach the architecture of a real existing machine. But its not necessary, virtual machines can also exist without correspondence of any real machine. It has become common for language developers to invent a virtual machine tailored to a new high-level language. Programs written in the high-level language are compiled to ”binaries” targeted at the virtual machine. The power of this approach has been clearly demonstrated with the java high-level language and java virtual machine, where a high degree of platform independence has been achieved, thereby enabling a very flexible network computing environment.

1.5 Standardized Computer System Components

There are three major components of a computer through which a computer system is constructed and they are hardware, the operating system, and application programs. The hierarchical nature of the system and the meshing of its major interfaces are illustrated in figure. All major system components are stacked on one another and they reflect the direct interaction of these components that takes place. For example, the operating system and application programs interact directly with hardware during normal instruction execution. As the operating system has special privileges for managing and protecting shared hardware resources, e.g. memory and the I/O system, therefore to interact with hardware resources, application programs need to make operating system calls.

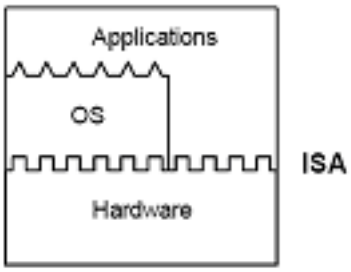

Figure 1.2: Computer system components

Well defined computer system architecture has many advantages over the conventional computer system architecture. Important system design tasks can be separated; hardware and software designers can work more or less independently. In fact, the three major components are often developed at different companies, and at different times, sometimes years apart. Application developers do not need to be aware of changes inside the OS, and hardware and software can be upgraded according to different schedules. Software can run on different hardware platforms implementing the same ISA, either within the same hardware generation or across generations. Because of its many advantages, the architecture model of Figure 1.2 has persisted for several decades, and huge investments have been made to sustain it. There are also significant disadvantages to this approach; however, these have become increasingly evident as software and hardware have continued to grow in complexity [1].

Problems arise because the major components work together only in the proper combinations. Figure1.3 shows three popular desktop computer systems, each constructed of hardware, an operating system and application programs. However, the components that form the three systems are not interoperable. Application software compiled for a particular ISA will not run on hardware that implements a different

ISA. For example, applications compiled for Linux and for Windows use different operating system calls, so a Windows application cannot be run directly on a Linux system and vice versa. In isolated computer systems, lack of software compatibility is bad enough, but in a heavily networked environment the problem becomes even worse. There are a lot of advantages to view a collection of networked computers as a single system, where software can freely migrate. This view is obstructed if the networked computers have incompatible ISAs and operating systems. That is, if a piece of software is restricted to running on only certain nodes of the network, then a great deal of flexibility and transparency is lost, especially when the network is very large.

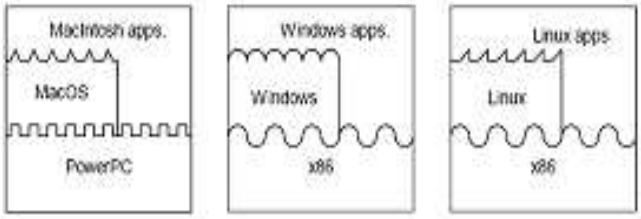

Figure 1.3: Three popular computer systems composed of different ISAs, OSs, and Applications

A second problem that arises is that innovation is sometimes limited by the need to support interfaces developed possibly decades earlier. For example, implementing new software concepts may be inhibited by an old ISA.

Third type of problem is that optimization across important interfaces is difficult. Interfaces allow independent development of the important components, but this can also be problematic. The developers on each side of an interface seldom communicate, so it is very difficult to cooperate closely on optimizations that cross an interface. Finally, in a traditional computer system, one operating system is matched with one hardware platform and all applications co-exist under the management and protection of the single operating system. Not only does this constrain all the system users to the same OS, but it also opens opportunities for exploiting security holes in the operating system. That is, the degree of isolation among the application programs is limited by the shared system software. This may be especially important when a large hardware system, e.g. a server, is to be shared by different groups of users who would like to be assured of a secure environment [1].

1.6 Virtual Machine Basics

The above problems can be solved by implementing a layer of software that provides a virtual machine environment for executing software. One type of virtual machine (VM) is illustrated in Figure 1.4 where virtualizing software is placed between the underlying machine and conventional software. In this example, virtualizing software translates the hardware ISA so that conventional software sees a different ISA from the one supported by hardware. As we shall see, virtualizing at the ISA level is only one possibility, but it is necessary to illustrate the range of VM applications.

The virtualization process involves.

1. The mapping of virtual resources, e.g. registers and memory, to real hardware resources and
2. Using real machine instructions to carry out the actions specified by the virtual machine instructions, e.g. to emulate the virtual machine ISA.

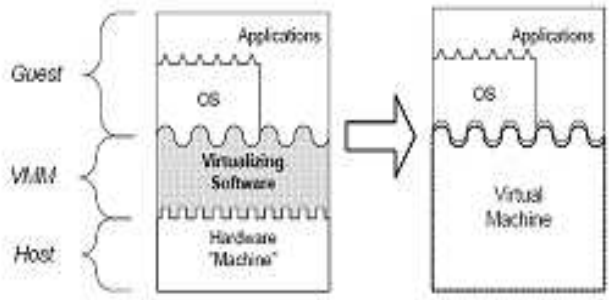

Figure 1.4: Virtualizing software can translate the ISA used by one hardware platform to another, forming a Virtual Machine, capable of executing software developed for a different set of hardware

With regard to terminology, the underlying platform is usually referred as the host, and the software that runs in the VM environment as the guest. The virtualizing software is often referred as the virtual machine monitor (VMM).

Virtualizing software can be applied in several ways to connect and adapt the three most important system components. As mentioned, emulation adds considerable flexibility by permitting cross-platform software portability. Virtualizing software can enhance emulation with optimization, by taking implementation-specific information into consideration as it performs emulation, or it can perform optimization alone, without emulation. Virtualizing software can also provide resource replication, for example by giving a single hardware platform the appearance of multiple platforms, each capable of running a complete operating system or a set of applications. Finally, the various types of virtual machines can be composed to form wide variety of architectures [1].

Given the wide variety of applications, VM technologies are widely used today to allow interoperability of the major system components. Furthermore, because of the heavy reliance on a few standards and consolidation in the computer industry, it seems likely that any major innovation, e.g. a new ISA, new OS, or new programming language will leverage VM technology. Consequently, for constructing modern systems, virtualizing software has essentially become a fourth major system component that merits equal standing with hardware, operating systems and application software.

In computer systems, with their many levels of abstraction, the meaning of machine is a matter of perspective. From the perspective of a process, the machine consists of a memory address space that has been assigned to the process, along with user level registers and instructions that allow the execution of code belonging to the process. The I/O system, as perceived by the process, is rather abstract. Disks and other secondary storage appear as a collection of files to which the process has access permissions. In the desktop environment, the process can interact with a user through a window that it creates within a larger graphical user interface. The only way the process can interact with the I/O system of its machine is via operating system calls, either directly, or through libraries that are supplied to the process. Processes are often transient in nature. They are created, execute for a period of time, perhaps with other processes along the way, and eventually terminate.

From a higher level perspective, an entire system is supported by an underling machine. A system is a full execution environment that can simultaneously support a number of processes potentially belonging to different users. All the processes share a file system and other I/O resources. The system environment persists over time as processes come and go. The system allocates physical memory and I/O resources to the processes, and allows the processes to interact with their resources via an OS that is part of the system [1].

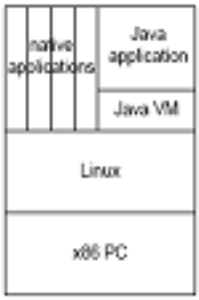

Figure 1.5: A process supported by a guest Java Virtual Machine executes alongside native processes on a Linux/x86 host platform

Just as there is a process perspective and a system perspective of what a machine

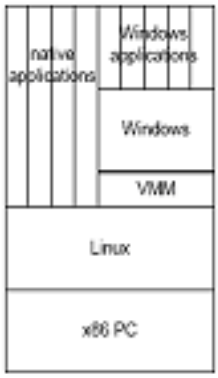

Figure 1.6: A Windows system supported as guest virtual machine running on a Linux/x86 host platform

is, one can also provide virtual machines at either the process level or the system level. As the name suggests, a process virtual machine is capable of supporting an individual process. A process virtual machine is created along with its guest process and terminates when the guest process terminates. For example, consider a host platform that consists of Linux running on Intel x86 PC hardware (Figure1.5). This system clearly can support native x86 applications that are compiled for a Linux system. However, this system can also act as a host for supporting guest Java processes via Java virtual machines. The Java VM environment can execute Java bytecode programs and perform I/O operations through Java libraries.

The same system could also support PowerPC/Linux guest applications via process level virtual machines that are capable of emulating the PowerPC instructions. All these processes can reside simultaneously within the same system environment, and they may interact in the ways that processes sharing a system normally do for example, a PowerPC process could create an x86 process and communicate with it via shared memory. In a desktop system, for example, they would each have individual windows, all appearing within the same X-windows GUI.

A system virtual machine provides a complete system environment. This environment can support multiple user processes, includes a file system, provides the processes with access to I/O devices, and, on the desktop, it supports a GUI. Again, consider a platform that consists of Linux running on x86 hardware (Figure1.6). The native system is Linux/x86, but the platform can also serve as a host for supporting a virtual Windows system, accessed through the Windows GUI, that contains a (virtual) Windows file system and can run Window applications. The Windows processes interact with each other, just as on a real windows system. And, they can interact with the native UNIX system via a network interface, just as if it were a physically separate system. However, the UNIX and Windows processes do not interact directly as if they were part of the same system because they are not part of the same system; for

example, a UNIX process cannot spawn a Windows process.

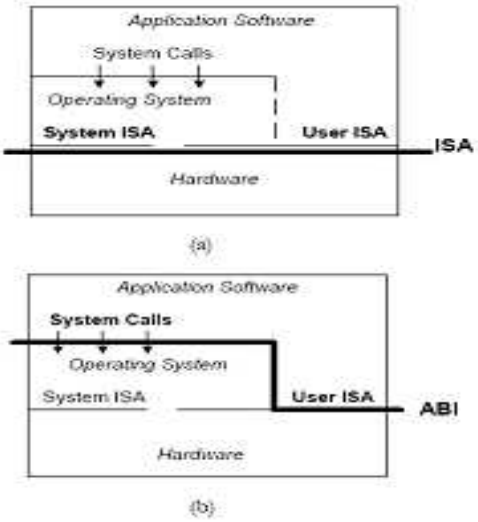

Figure 1.7: System Interfaces: a) Instruction Set Architecture (ISA) interface,b) Application binary Interface (ABI)

For implementing virtual machines, there are two standardized interfaces that will be of interest to us. The interfaces roughly correspond to the process level and the system level. These interfaces are shown in more detail in Figure 1.7. The ISA, shown in Figure 1.7a, includes both user and system instructions. The user instructions are available to both application programs and to the operating system. The system instructions include privileged operations that permit the direct manipulation, allocation, and observation of shared hardware resources which can be the processor, memory and I/O [1].

The system call interface is typically implemented via a system call instruction that transfers control to the operating system in a manner somewhat similar to a subroutine call, except the call target address is forced to be a specific address in the operating system. Arguments for the system call are passed through registers or a stack held in memory, following specific conventions that are part of the system call interface.

Now we will describe some particular type of virtual machine. These virtual machines span a broad spectrum of applications, and we categorized them into two main parts Process VMs and System VMs in our discussion.

1.7 Process VMs

Process VMs support guest software at the ABI level. Often, an important objective is to provide good performance when emulating a guest program binary. In a typical process VM implementation, the guest process and the VM software are bundled together as a single host process. This leads to a number of challenges in providing complete transparency. This is a high standard of compatibility to meet, but it is one that most other varieties of VM strive for. Here, there are opportunities for providing some architecture primitives to allow efficient, transparent Process VM implementations. Process

virtual machine provides user applications with an environment of virtual ABI. Process VMs has ability to provide replication, emulation, and optimization. We will discuss these one by one.

1.7.1 Multiprogramming

In multiprogramming each process itself behaves like a complete machine, as every process have its own address space and also every process is given access to a file structure. The operating system has to share and manage all resources to make this possible. In effect, the OS provides replicated process level virtual machines for each of the concurrently executing applications. And, there are virtually an unlimited number of such processes. So the process virtual machine can be freely replicated.

1.7.2 Emulation and Dynamic Binary Translators

One of the main problems of process level virtual machines is that, process machines have to support program binaries that are compiled on another instruction set. Such an emulating process virtual machine is shown in Figure1.8. Application programs are compiled for a source ISA, but the hardware implements a different target ISA. As shown in the figure, the operating system is the same for both the guest process and the host platform. The example illustrates the Digital FX!32 system. The FX!32 system could run Intel x86 application binaries compiled for Windows NT, on an Alpha hardware platform also running Windows NT [1].

Figure 1.8: A process VM that emulates guest applications. The Digital FX!32 system allows Windows x86 applications to be run on an Alpha Windows platform

1.7.3 Dynamic Optimizers

Besides translating from source to target code, dynamic translators also perform some optimizations in the process. We need dynamic optimization where host and the guest

use the same instruction sets, and optimization is the basic goal of the virtual machine. Dynamic translators are implemented in a manner very similar to emulating virtual machines, including staged optimization and software caching of optimized code. An example of such a dynamic optimizer is the Dynamo system, developed as a research project at Hewlett-Packard. A dynamic optimizer can collect statistics on a running program, i.e. construct a profile, and then use this profile information to optimize the binary on-the-fly [1].

1.7.4 High Level VMs: Complete Platform Independence

Cross-platform portability remains the most important objective of the above mentioned virtual machines. That has been seen very difficult to achieve. By designing the virtual machines in the program first, it can be made easy for virtual machines to be cross-platform portable. This can done and achieved when a process level virtual machine have to design at the same time when an application development environment, including a high level language, is being defined. Through this a Virtual machine does not correspond to a particular hardware. These High Level VMs are similar to the ISA-specific process VMs described above. The focus is to support applications and try to reduce hardware related issues.

One of the advantages of the high level VMs is that, if virtual ritual is implemented on the target platform, then software can be completely portable with high level virtual machine. Though it takes some time, but still it is an easier task than developing a compiler for each platform and re-compiling an application beginning with the HLL whenever it is to be ported. And it is also much simpler than developing a conventional emulating process VM for a typical real world ISA [1].

1.8 System Virtual Machines

A fully system environment in which many processes, which belongs to different users can exist at the same place, a system virtual machine is supposed to provide this environment. These types of virtual machines first time developed in the 1960s and early 1970s. Through these system virtual machines, single host machine hardware has the ability to support multiple guest operating systems simultaneously. At that time computer systems were developed in very large size and very expansive. That's why computer system has to be shared among many users. And this type of multi-user environment, different users mostly wanted different operating systems to run on their shared environment. At that system virtual machine make it possible for different group of users to have different operating systems on the same shared hardware environment. Alternatively, a multiplicity of single-user OSes allowed a convenient way of implementing time-sharing amongst several users. Figure1.9 illustrates these classical system VMs [1].

The main feature that provided by system virtual machine is replication, that's provide through VMM. That main problem for the system virtual machine was the sharing of resources among multiple guest operating system environments. VMM has the ability to access and manage the shared hardware resources. All guest operating systems and every type of application that has to compiled for that operating system

will be managed through the hidden control VMM. What is going behind VMM is hidden; guest software is unaware of the work performance by the VMM. Multiple guest Operating systems can be supported through the use of VMM in system virtual machine; Figure1.9 there how system virtual machine supports two operating system on the same single hardware.

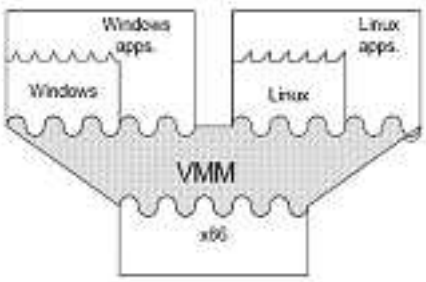

Figure 1.9: An example system VM – supporting multiple OS environments on the same hardware

There are a lot of advantages of this type of virtual machines. The most important advantage of this type of virtual machine is that many operating systems can be simultaneously used at the same single hardware. It means that software developed for different operating system can be simultaneously executed on the same hardware for another operating system. For example, DOS, Windows NT, OS/2, Linux, and Solaris operating systems are all available for the x86-based PC platform. And, different application programs have been developed for each of the operating system. This leads to a problem, if a user wants to run an application written for Windows NT on a PC that is currently running Solaris. There is a similar problem if multiple users are sharing a larger system like a server, and most of them prefer to use different operating systems.

Another important application, and one that could possibly be the most important in the future, is that the separate virtual environments provide a security firewall for protecting independent groups of users and applications. Also, OS software development can be supported simultaneously with production use of a system. This replicated VM approach has been very successful for IBM; it has continued to evolve and is a key part of large multiprocessor servers, today's equivalent to the mainframes of the '60s and'70s [1].

Chapter 2

Virtualization Technologies

2.1 History of Virtual Machine

Virtual Machine concept came into existence first time in 1960s, when IBM developed it to provide concurrent, interactive access to a mainframe computer. Each virtual machine supposed to be used as an instance of the physical machine and it gives users a direct access to the physical machine. It was a transparent way to enable time-sharing and resource-sharing on the highly expensive hardware. Each VM was a fully protected and isolated copy of the underlying system. Users could execute, develop, and test applications without ever having to fear causing a crash to systems used by other users on the same computer. So, virtualization concept reduces the hardware cost and it improves the productivity by letting group of users to work on it at the same time.

With the passage of time as hardware became cheaper and multiprocessing operating systems were being developed. With the emergence of wide varieties of PC based hardware and operating systems in 1990s, the virtualization ideas were in demand again. The main purpose of the virtual machines at that time was to make possible the execution of a range of applications, originally targeted for different hardware and operating systems, on a given machine. The trend of virtualization continuing till now, Virtuality differs from reality only in the formal world, otherwise possessing the similar effect.

In the world of computer science, a virtual environment is perceived to be the same as that of a real environment, though the underlying mechanism is different for the application programs. A typical computer system already uses many such technologies. One such example is the virtual memory implementation in any modern operating system that lets a process use memory typically much more than the amount of physical memory its computer has to offer. This virtual memory also enables the same physical memory to be shared among hundreds of processes. Similarly, multitasking can be thought of as another example where a single CPU is partitioned in a time-shared manner to present some sort of a virtual CPU to each task. In a different setting, a cluster of medium-speed processors can be grouped together to present a single virtualized processor that has a very high clock speed. There are lots and lots of examples in today's world that exploit such methods.

With the increase in applications of virtualization concepts across a wide range of

areas in computer science, the scope of the definition has been increasing even more. However, just for the discussions of virtual technologies here, we use the following relaxed definition:"Virtualization is a technology that combines or divides computing resources to present one or many operating environments using methodologies like hardware and software partitioning or aggregation, partial or complete machine simulation, emulation, time-sharing, and many others".

There can be several reasons how virtualization can be useful in practical scenarios, a few of which are the following:

- Server Consolidation: To consolidate workloads of multiple under-utilized machines to fewer machines to save on hardware, management, and administration of the infrastructure
- Application consolidation: A legacy application might require newer hardware and operating systems. Fulfilment of the need of such legacy applications could be served well by virtualizing the newer hardware and providing its access to others.
- Sandboxing: Virtual machines are useful to provide secure, isolated environments for running foreign or less-trusted applications. Virtualization technology can, thus, help build secure computing platforms.
- Multiple execution environments: Virtualization can be used to create mutiple execution environments in all possible ways and can increase the quality of service by guaranteeing specified amount of resources.
- Virtual hardware: It can provide the hardware one never had, e.g. Virtual SCSI drives, Virtual ethernet adapters, virtual ethernet switches, hubs, and so on.
- Multiple simultaneous OS: It can provide the facility of having multiple simultaneous operating systems that can run many different kind of applications.
- Debugging: It can help debug complicated software such as an operating system or a device driver by letting the user execute them on an emulated PC with full software controls.
- Software Migration: It eases the migration of software and thus helps mobility.
- Testing/QA: Helps produce arbitrary test scenarios that are hard to produce in reality and thus eases the testing of software.

In reality, machines were never designed with the aim to support virtualization. Every computer exposes only one bare machine interface; hence, would support only one instance of an operating system kernel. For example, only one software component can be in control of the processor at a time and be able to execute a privileged instruction. Anything that needs to execute a privileged instruction, e.g. an I/O instruction, would need the help of the currently booted kernel. In such a scenario, the unprivileged software would trap into the kernel when it tries to execute an instruction that requires privilege and the kernel executes the instruction. This technique is often used to virtualize a processor.

In general, virtualizable processor architecture is defined as an architecture that allows any instruction inspecting or modifying machine state to be trapped when executed in any but the most privileged mode. This provides the basis for the isolation of an entity from the rest of the machine. Processors include instructions that can affect the state of a machine, such as I/O instructions, or instructions to modify or manipulate segment registers, processor control registers, flags, etc. These are called sensitive instructions. These instructions can affect the underlying virtualization layer and rest of the machine and thus must be trapped for a correct virtualization implementation. The job of the virtualization layer e.g. the virtual machine monitor is to remember the machine state for each of these independent entities and update the state, when required, only to the set that represents the particular entity. However, the world is not so simple; the most popular architecture, x86, is not virtualizable. It contains instructions that, when executed in a lower-privileged mode, fails silently rather than causing a trap. So virtualizing such architectures are more challenging than it seems.

Beside from architectures point of view, there are many other problems that make virtualization difficult. Since a virtualization layer a Virtual Machine Monitor has little knowledge regarding what goes on inside a virtual machine, it is typically hard for it to know what not to do. For example, a page fault exception caused by a guest OS inside one virtual machine should not be handled by the virtualization layer, and rather, be left to the guest OS to handle by itself. Similarly, a virtualization layer in the OS level should not handle any system call issued by one of the processes in a virtual machine; rather, should be left to its guest OS kernel to handle. Optimizations are also hard to achieve in the virtualization software as it does not know when a virtual machine does not need some resource. For example, it is hard for the VMM to know if the guest OS inside one of its virtual machine (VM) instances is running idle thread and is wasting processor cycles that can be allocated to other VMs for better performance.

Conceptually a virtual machine represents an operating environment for a set of user-level applications, which includes libraries, system call interface, system configurations, and file system state. There can be several levels of abstraction where virtualization can take place: instruction set level, hardware abstraction layer (HAL), operating system level system call interface, user-level library interface, or in the application level. Whatever may be the level of abstraction, the general phenomenon still remains the same; it partitions the lower-level resources using some novel techniques to map to multiple higher level VMs transparently. We will only discuss the hardware abstraction layer (HAL) level abstraction; others are not in the scope of our discussion.

The functionality and abstraction level of a HAL level virtual machine lies between a real machine and an emulator. A virtual machine is an environment created by a VMM, which is the virtualization software lying between the bare hardware and the operating system and gives the operating system a virtualized view of all the hardware. A VMM can create multiple virtual machines (VMs) on a single machine. While an emulator provides a complete layer between the operating system or applications and the hardware, a VMM manages one or more VMs where every VM provides facilities to an operating system or application to believe as if it runs in a normal environment and directly on the hardware.

2.2 Virtualization at the Hardware Abstraction Layer

This virtualization is the most popular virtualization used by the commercial PC emulators today. This technique is used on popular x86 platforms to make it efficient and to make its use, viable and practical. A virtualization technique provides a mapping from the virtual resources to the physical resources, and for the computation in virtual machines the native hardware is used. When ever the virtual machine needs to have access to the physical resources, the simulator takes the control and multiplexes appropriately.

This type virtualization technology to work efficiently, the VM should have ability, so that it can trap every single privileged instruction and can pass it correctly to the underlying VMM that will be responsible for taking care of it. This is because, in a VMM environment, several VMs can exist, each having an OS running. And each operating system wants to issue privileged instructions and get the CPU's attention. During privileged instruction execution, when ever a trap occurs, the instruction is directly sent to VMM, instead of generating an exception and then crashing it.

This gives the VMM a full control of the machine, VMM keeps each VM isolated from other VM. Then it's up to the VMM, either it executes the instruction on the processor, or emulates the results and returns them to the VM. The popular platform, x86, is not fully-virtualizable, that's why some privileged instructions fail silently, when executed with insufficient privileges. So, for a virtualization technique to work properly, it should have ability to be controlled by the VMM, when ever a bad instruction executes. Some popular emulators use such as scanning and dynamic instruction rewriting to overcome such issues.

Now we will explore some of the techniques used to create correct and efficient virtualized machines and some of their features and shortcomings.

2.2.1 VMWare

VMware is an industrial strength virtual machine company with three levels of VM products: VMware Workstation, VMware GSX Server, and VMware ESX server [7]. Here we will concentrate on the VMwareWorkstation product used for normal PC users that is very common and a few features and the differences with the other ones.

VMware's VMMs can be standalone or hosted. Figure. A standalone VMM is basically a software layer on the base hardware that lets users create one or more VMs. These are similar to operating systems, require device drivers for each hardware device, and are typically limited in hardware support. Such VMMs are typically used in servers, VMware ESX server is an example of such architecture. A hosted VMM, however, runs as an application on an existing host operating system.

That's why it can take advantage of the host operating system for memory management, processor scheduling, hardware drivers, and resource management. VMware Workstation use this hosted virtual machine architecture. VMware products are targeted towards x86-based workstations and servers. Thus, it has to deal with the complications that arise as x86 is not a fully-virtualizable architecture. VMware deals with this problem by using a patent-pending technology that dynamically rewrites portions of the hosted machine code to insert traps wherever VMM intervention is required [Xen and the Art of Virtualization]. Although it solves the problem, it adds some over-

head due to the translation and execution costs. VMware tries to reduce the cost by caching the results and reusing them wherever possible.

2.2.2 Virtual PC

Microsoft's Virtual PC, is a product very similar to what is offered by VMware Workstation [8]. It also based on the Virtual Machine Monitor (VMM) architecture and lets the user create and configure one or more virtual machines. Apart from the features supported by VMware, it provides two distinguishing functionalities. It maintains an undo disk that lets the user easily undo some previous operations on the hard disks of a VM. This enables easy data recovery and might come handy in several circumstances. The other distinguishing feature is binary translation, which is used to provide x86 machines on Macintosh-based machines.

There are a number of shortcomings that the Virtual PC possess in terms of features when compared to VMware. Linux, FreeBSD, OpenBSD, Solaris, etc are not supported as guest operating systems in Virtual PC. The Virtual PC VMs do not have support for SCSI devices, unlike VMware workstation, although some SCSI disks are recognized as IDEs by the VMs. It does not let user add or upgrade the hardware set for a VM. Once configured, it makes it impossible to change the hardware devices a VM possesses later on. Linux operating systems are not available as host OS.

2.2.3 Denali

The virtual machines provided by the VMwareWorkstation and Microsoft Virtual PC are very efficient and practical to use, which supports almost all the PC-like features, but due to design limitations it is hard to create and use thousands of virtual machines instances at the same time simultaneously. The mechanism through VMM achieves the virtualization, its very scale virtual machine instances for a high number. For example, mechanisms like, interrupt handling, memory management and world switching are very difficult to scale for more then a few number of active VMs, but some times for various purposes, it might be possible the need of large numbers of VMs working simultaneously. The University of Washington's Denali project [9] ddress this issue and come up with a new virtualization architecture to support thousands of simultaneous machines, which they call Lightweight Virtual Machines. To achieve this they used a technique called paravirtualization. Through this technique it became possible for the VMs to increase scalability and performance, with a very easy implementation mechanism. This new architecture provides new interfaces for the customized guest operating systems. This architecture provides mechanisms to modified architectural features that make implementation of the guest operating systems very simple. The new architecture provides a set of virtual registers for ease of data transfer between the virtualization layer and the virtual machines. Among other things, it supports a modified interrupt delivery, and does not support the virtual memory concept. All these modifications are incorporated aiming at a simpler implementation with low overhead to make the overall system scalable and the VMs lighter.

2.2.4 Xen

The techniques discussed so far concentrates for full virtualization. In this type of virtualization all the applications and the operating systems with in a VM reside completely in a virtual world without any knowledge of the underlying physical machine. But their might be some cases when operating system or the application running with in VM needs to have knowledge of real machine resources. For example for guest operating system it might be helpful to efficiently support time-sensitive tasks, if it can see the time of both real and virtual machine. Similarly to see address of the real machine also improves performance of the guest operating system and applications running on it. On the other habd full virtualization is not easy to achieve specially when implemented on x86 due to its problem of not being a virtualizable architecture. So achieving high performance task with X86 architecture having full resource isolation is very difficult. Also completely hiding the effects of resource virtualization from the guest operating systems creats risks for both correctness and performance. Also there are some other issues like Quality of Service, security, and denial of service. These issues motivate the researchers in University of Cambridge and they come up with a modified architecture for virtualization, called Xen
citeXartVir.

Xen uses a paravirtualized architecture in each of its VMs to maximize performance and resource isolation. It aims at supporting around a hundred VM instances within a single physical machine having a reasonable performance. Although Denali [9] uses a paravirtualized architecture for more or less the same purposes, they both have diffrent targets. Denali is designed to support thousands of running virtual machines network services, but the vast majority of all these services are of small-scale and not much popular. As Denali was designed to host only a single application with single user unprotected guest operating systems, that's why it doesn't have any support of virtual memory concept, which is the most important and common feature of every modern operating system. VMM performs all the paging work to and from the disks that can be vulnerable to thrashing. While Xen is not performing paging, all paging is done by the guest operating system. Denali Virtualizes namespaces, where as Xen provides secure access control with in hypervisor, is enough to ensure all type of protection when making physical resources to be accessible through the guest operating system.

To provide an efficient page table, Xen exists in a 64MB section of every address space. This avoids a TLB flush when entering and leaving the hypervisor. Guest operating systems have the direct access to hardware page tables, however, updates are batched and validated by Xen. This allows Xen to implement a secure but efficient memory management technique when compared to VMware where every update to the page table is done by VMM and updated. Each guest operating system is provided a timer interface and is aware of both real and virtual time. In this way, it tries to build a more robust architecture that preserves all the features that are of importance to application binaries keeping the porting effort of the guest operating system negligible.

2.2.5 Plex86

Plex86 is the open source free-software alternative for VMWare, VirtualPC, and other IA-32 on IA-32 Virtual PC products [10]. Plex86 project works toward an open-source x86 simulator with virtualization. Through the virtualization it improves the efficiency of a virtual machine such as Bochs. It takes advantage of the hardware similarity of the guest and the host machine and through this it makes possible to allow large portions of the simulation to take place with the speed of host hardware. When the simulated machine talks to the hardware, or enters certain privileged modes such as the kernel mode, simulator takes control and simulates the code in software at very slow speed, as the Bochs does. Virtualization helps Plex86 run much faster compared to Bochs.

2.2.6 User-mode Linux

User-Mode Linux is an open source and safe, secure way of running Linux versions and Linux processes on the top of Linux [11]. We can run buggy software, experiment with new Linux kernels or distributions, all without any risk of damaging the main Linux setup.User-Mode Linux gives us a virtual machine that may have more hardware and software virtual resources than our actual, physical computer. Disk storage for the virtual machine is entirely contained inside a single file on our physical machine. We can assign our virtual machine only the hardware access we want it to have. With properly limited access, virtual machines can never change or damage real computer, or its software

Basically, it gives a virtual machine on which a Linux version can execute as it does on a physical machine, and everything implemented in the user-level. Unlike previous ones that use the VMM right on the base hardware, this uses a different implementation being on top of the operating system and in the user-space. But the abstraction level still remains more or less similar to the previous ones. It lets the user configure virtual hardware resources that would be available for the guest Linux kernel. Since everything runs in the user-level, safety is assured. Its hardware support comes in the form of virtual devices that make use of the physical resources. Devices supported are, block devices, consoles, serial lines, network devices, SCSI devices, USB, Sound, and many more. The UML runs its own scheduler independent of the host scheduler, and basically supports anything that is not hardware specific.

The virtual machine and the guest Linux kernel are tighty coupled. Executing totally in the user space, the major challenge it faces is to be able to intercept the system calls in the virtual kernel, as they would naturally go to the real host kernel. Using the Linux ptrace facility to track system calls, it diverts the system calls made by processes running within the Virtual Machine to the user space kernel to execute them. Similarly, traps are implemented through Linux signals. Kernel and the processes within the VM share the same address space; and conflicts with process memory are avoided by placing the kernel text and data in areas that processes are not likely to use. Each process in the virtual machine gets its process in the host kernel. In order for the virtual kernel's data to be shared across all the processes in the VM, its data segment is copied into a file, and the file is mapped shared to all the processes.

Chapter 3

Benefits and challenges for virtual environment

3.1 How Virtual environment can be better then real environment

Most researches in the filed of virtual machines says that Operating system and applications running on a real machine should be transferred into a virtual machine. Only those programs that are necessary to run on real machine should run on real machine and usually these programs are host operating systems, virtual machine monitor, programs that are necessary for local administration and services enabled by this virtual machine structure. All network services would run in virtual machine and real machine just need to forward all the network packets for the virtual machines.

Virtual machine structure enables services to be added below the operating systems and that can be done without trusting or modifying the operating system. There are many services that can take advantage of this technology and demonstrate how virtual machine structure can be better then real structure.

Virtual machine structure allows us to provide services same as providing services in the real hardware machine. As these services are implemented in a layer of software which is VMM (virtual machine monitor). These services can be provided more efficiently, easily and flexibly then they could if they were implemented by modifying the hardware. Particularly these services can be provided below the guest operating system without trusting or modifying it and providing services in the layer of VMM (virtual machine monitor) has many benefits especially it's useful for security and mobility.

Providing services by modifying a virtual machine has similar benefits as providing services by modifying the real machine. In virtual machines all services including the guest operating systems run separately from all processes. This separation of services from processes provides great benefits of security and portability. As in virtual machines services don't need to trust the operating systems and they only have to trust the virtual machine monitor and trusting virtual machine monitor instead of trusting real machine is less risky because virtual machine monitor is considerably smaller and simpler. Services in operating systems are more vulnerable to malicious and random faults, because operating systems are larger and have more security and reliability holes. To separate services from guest operating system also increases portability. As

with virtual machines we can implement services without needing to change the operating system, its means that they can work across multiple operating system vendors and versions.

As we are gaining benefits similar to providing services in a real machine, virtual machines have many advantages over the physical machines. Modification of virtual machine is easier then real machine because the virtual machine monitor that creates the virtual machine abstraction is a layer of software, manipulating the state of virtual machine is much easier then manipulating the state of physical machine. The state of a virtual machine can be saved, cloned, encrypted, moved or restored, none of which is easy to do with physical machines, and a virtual machine has a very fast connection to another computer system, that's the host machine on which virtual machine is running. Where as physical machines are separated by physical networks, which are slower then the memory bus that connects a virtual machine with the host.

3.2 Benefits of Virtualization

There are a number of driving forces behind virtualization, both business needs and availability needs. Some of the most significant drivers behind this technology are as follows:

- Flexibility and agility
- Server consolidation
- Business continuity and disaster recovery
- Reduction in downtime
- Reduction in administrative costs

3.2.1 Flexibility and agility

Deploying virtualization technologies can greatly increase business flexibility and agility. By decoupling business processing from the physical hardware, virtualization enables IT departments to respond rapidly to growing changes in demand. Virtualization technologies also allow businesses to quickly deploy new products and services, to offsite premises, remote offices and contract personnel. This also enables expansion into new markets. There is also a much lower hardware requirement for testing out new applications. Software developers can develop and test their code on multiple Operating Systems, which will reduce the development and testing time. They can also instantly reload their test systems from an image, which can result in a faster build, test, and rebuild cycle. Virtualization can also reduce the routine deployment processes for production implementation from minutes instead of days or even weeks.

3.2.2 Server Consolidation

Server consolidation and improved server utilization is yet another driving force for the adoption of virtualization technologies. Virtualization allows businesses to combine workload from multiple underutilized physical machines into a single physical system.

This can greatly reduce the overall hardware spending, because it requires far fewer physical systems for the same application load. It also has a greater effect on the overhead costs, including, cooling, power, storage, and physical administration.

3.2.3 Business continuity and disaster recovery

Virtualization and streaming allow for easier software migration, including system backup and recovery, which can make it extremely valuable as a disaster recovery or a business continuity planning solution. Virtualization can duplicate critical servers, so that it does not need to maintain expensive physical duplicates of every piece of hardware for disaster recovery purposes. Disaster recovery systems can even run on dissimilar hardware. In addition to this, virtualization can reduce downtime for maintenance, as a virtual image can be migrated from one physical device to another to maintain availability while maintenance is performed on the original physical server. This applies to both severs and desktops, and even mobile devices. Virtualization allows end-users to remain productive and get back to work faster when their hardware fails.

3.2.4 Reduction in Downtime

The reduction in downtime is another key driving force behind virtualization. Virtual images are much easier to restore after a failure, either an operational failure or a hardware failure. The portability of virtual images allows new and different hardware configurations to be used for recovery purposes, thereby reducing downtime. Likewise, from an end-user perspective, desktop failures are critical, but with application virtualization and streaming, end-users are not tied to a specific failing desktop or location, and as a result, can get back to work on any machine in no time, reducing the impact of any downtime.

3.2.5 Reduction in Administrative Costs

With virtualization technologies, administration becomes a lot easier, faster and cost-effective. Visits to the user's place of work can almost be eliminated through application virtualization and streaming, since business applications are being maintained centrally. Any failure in the endusers environment can be fixed quickly and easily. In addition to this, virtual server Operating Systems can for the most part be managed remotely using standard tools and network interfaces, rather than needing physical attention.

3.3 Challenges in Managing a Virtual Environment

Providing services at the virtual machine level also hold challenges, the main challenge is of performance. Running all applications above the virtual machine hurts performance due to virtualization overhead. For example system calls in a virtual machine must be trapped by the virtual machine monitor and re-directed to the guest operating system. Hardware operations issued by the guest must be trapped by the virtual machine monitor, translate, and reissued. Some overhead is unavoidable in

a virtual machine; the services enabled by that machine must outweigh this performance cost. Virtualizing an anx86 processor doesn't trap on some instructions that must be virtualized. One way to implement a virtual machine in the presence of these non-virtualizeable instructions is to re-write the binaries at run time to force these instructions to trap, but this comes with significant overhead.

In virtual machines there is also semantic gap between the virtual machine and the service. Services in the virtual machine operate below the abstraction provided by the guest operating system and applications. This can make it difficult to provide services. For example, it is difficult to provide a service that checks file system integrity without knowledge of on-disk structures. Some services do not need any operating system abstractions; secure logging is an example of such a service. For services that require higher level information, one must re-create this information in any form. Full semantic information requires re-implementing guest Operating systems abstractions in or below the virtual machine. However, there are several abstractions, like virtual address space, threads of control, network, network protocols, and file system formats that are shared across many operating systems. By observing manipulation of virtualized hardware, one can reconstruct these generic abstractions, enabling services that require semantic information.

While virtualization offers a platform of benefits, it also introduces some new management challenges which will have to be considered and planned for by businesses which are considering a virtual stance.

Some of the key business challenges that virtualization brings are as follows:

- Bandwidth implications
- Policy management
- Image propagation
- Security

Some of the other challenges that virtualization bring to the table include: a new level of complexity for capacity planning; a lack of vendor support for applications which are running on a virtual environment; increased reliance on hardware availability; an additional layer of monitoring complexity; and an overall increase in the complexity of the IT environment.

3.3.1 Bandwidth Implications

Businesses will need to ensure that they have the appropriate network bandwidth for their Server virtualization requirements. For example, instead of having one server which uses 100MBPS Ethernet cable, now 10 or even 100 virtual servers will have to share the same physical connection. However, Application streaming actually minimizes load on the network since there is no need to download the entire application. Typically, only 10-15

3.3.2 Policy Management

Businesses will need to look to deploy a form of automated policy based management stance together with their virtualization strategy. For example, resource management

should include some form of automated policy tools for disk allocation and usage, I/O rates, CPU utilization, memory allocation and usage, and network I/O. These management tools will need to be able to push resources in shared environments, to maintain service levels and response times which are appropriate to each virtual environment. The administrators will need to be able to set maximum limits, and allocate resources across virtual environments. Allocations will need to have the capability to change dynamically to respond to peaks and drops in load balancing

3.3.3 Image propagation

OS and server virtualization can and will lead to rapid propagation of system images. This happens for the simple fact that it is much easier and faster to deploy a new virtual image than it is to deploy a new physical server, without management approval or hardware procurement. This can bring with it a high-degree of management and maintenance requirements, and potentially lead to some significant licensing issues, including higher costs and compliance issues. This propagation can also lead to a significant storage challenges, such as competing I/O and extreme fragmentation, multi-disk access, and increased maintenance time, effort, and cost. Businesses will need to manage their environment with the same level of discipline as their physical infrastructure, making use of discovery tools to detect and prevent new systems from being created without following the appropriate processes.

3.3.4 Security Considerations

While virtualization can bring a lot of security benefits, security also becomes a management issue in a virtualized environment. There will be more systems to secure, more points of entry, more vulnerabilities to patch, and more interconnection points to exploit, across virtual systems, as well as across physical systems. Access to the host environment becomes more critical, because it will allow access to multiple guest images and applications. Businesses will need to pay attention for securing their virtual images just as well as they secure their physical systems.

Chapter 4

Literature survey

4.1 Case Study: Using Virtual Machines for teaching System Administration

4.1.1 Inroduction

In this case study we will discuss the virtual machines developed and used by Adam Vollrath and Steven Jenkins, at the Department of Computer and Information Science East Tennessee State University for teaching system administration education. As we know that System administration requires hands on practical experience that's why for System Administration student to get expertise its necessary not only to have a single system for each student, but also a good network lab having all necessary equipments. Using removable hard drives can be effective in minimizing the total number of computers needed. Adam Vollrath and Steven Jenkins used Virtual Machine technology to simulate multiple computers on a single machine for system administration students [2]. Here we will discuss this system and see how this system helps for teaching System Administration.

In the field of Information technology system administration is mostly taught as an upper level in many institutions. This course is very important as it prepares student to deploy and mange the networked infrastructures of computers that are common in the modern business environment. This course is not that can be taught theoretically, it requires practical knowledge of various technologies that are essential in the field of System Administration. Teaching these valuable skills requires hands-on practice in an environment that simulates the real world and for this type of experience usually we need a lab of networked machine with all necessary equipments. The ideal situation for system administration course to create an actual client/server infrastructure each student needs at least two dedicated machines. There are many factors that creates hurdle in this way like cost, space and many other similar factors. That's why institution must leverage their existing technologies effectively so that student can get sufficient hands on practical experience at least cost.

Virtualization is one of the method through which this goal can be achieved by getting the most out of any given machine. Virtual machines are nearly same as real machines and therefore creating a network of virtual machine is not difficult. Thus in scenario where multiple machines were necessary, as one machine can host a vir-

tual network, without having any interaction with the real network or damaging the real network of the host computer [2]. These people got idea from institutions that were using virtualization for teaching security courses and they showed that how virtualization can be used to improve the teaching of System Administration, without a significant increase in cost.

4.1.2 University Background Information

East Tennessee State University is a state supported university and it is located in Johnson City, TN, comprehensive, regional university governed by the Tennessee Board of Regents (TBR) [2]. Its neither a big sized nor a small sized university, its a mid-sized university and the total number of students at any time are about 12,000. Mostly students come from Northeast Tennessee and the adjacent regions of Virginia and North Carolina. The department of computer science (CIS) is one of famous department that's why CIS has large number of enrolments every year. Department has about 19 faculty members 11 pot of them are doctorates, some of them working towards doctorates and others having master's degrees.

Computer science department is very up to date department of the university, CIS department changes its courses and keeps up to date with changing industry standards, and computer science department has type of programs in the field on computer science.

1. Computer Science (CS)
2. Information Systems (IS)
3. Information Technology (IT)

There are two courses of about system administration that were taught in computer science department, one is basic level of course about System Administration and second is advance level of course about System Administration. Every year department has approximate 50% of its undergraduate enrolment in the field of IT and others are in CS and IS. CIS department also runs graduate courses and every year CIS department has approximately 60 graduate students within the three major fields. The primary System Administration course is required for the students in the IT program and is an elective for the other two concentrations

4.1.3 System Administration Courses

In the CIS department has a basic level of course about system administration which is CSCI 4417 [2]: Introduction to System Administration, this dual listed course is mandatory for all information technology fields undergraduates, and is not essential for graduate students in the field of IT, the second course about System Administration that CIS department runs is CSCI 5360: Advanced Topics in System and Network Administration, this course is mandatory for graduate student and also has CSCI 4417 as prerequisite, that's why graduate students have to take also CSCI 4417. At the beginning of the term department usually have enrolment of about 60 student making two sections having 30 students in each sectionin both the Fall and Spring semesters.

The CIS department's main goal for basic level of course about system Administration was that it should be vender neutral. They would like to present the relevant material in real system environment but they want to be tied to a particular vendor, as they are using both Microsoft and Linux operating systems. They also wanted to deliver full hands on practical experience to the student. There is high risk permitting System Administration student to use the same lab for experiments as other classes. Some universities permitted System Administration and Security courses to use the same lab facilities as other classes, it was determined that for security reasons the System Administration courses would be better served by being in a separate lab. And for this purpose university provided a room for the department and a grant was also provided for the initial level of computer lab setting for the System Administration course.

Assignments

The list below shows a typical set of assignments that are required for the System Administration course students.

1. Operating system installation
2. DHCP and DNS
3. Core Services (FTP, NFS, SMB, and HTTP)
4. User management
5. Software management
6. Services (NIS)
7. Security (port scanning, firewalls, change management, password analysis)
8. Backups

Some of these assignments require student to work in groups. Assignments have multiple parts and for most assignments require multiple operating systems to be configured. For example, the Windows portion of the software management requires Active Directory to be configured on a server, and it requires a Windows client to be joined to the domain.

Problems

The major problem they have is those students do not have their dedicated machines. Their lab consists of about 30 computers, and in each semester approximately 60 students take the course. Hence due to small number of machines available in the lab, it's not possible for the department to provide each student a dedicated machine for the whole semester or we can say that it's not possible to dedicate each student, his owned configuration of the system. First they tried to elevate this problem by requiring each student to purchase a hard drive and then assigning each student a removable drive enclosure. This solution some how lets the student to save their work to their hard

derive after finishing each session so that another student can use this system for its own configuration, but this solution was not good, therefore it did not work for them.

The second major problem they encountered was about the logistics of arranging and scheduling lab timing. This was because there was many assignments for which student need to work in groups for example student's system will act as a server while another's will act as a client, and department also encouraged students to work in groups, but student complaints about arranging lab schedules and they said that they would like to work on assignments separately instead of working with a group member.

The third major problem for them was the checkoff, because lab assignments needs to be checked run time in the lab, and for this purpose they used former students of System Administration class as lab monitors, but the checkoffs of the lab assignments requires a lot of resources like resources of the lab time and lab equipments, as these type of checkoofs need to spend a lot of time in lab for every student on a running system and it also requires higher cost for assigning more-skilled lab monitors

One another problem they encountered was the problem of conducting hands on examination, because hands on practical examinations are mostly time consuming and it also require an instructor or lab monitor for the inspection of each student work before students were allowed to leave the lab.

4.1.4 Their Vision

By exploring and finding other education institution that have used virtualization to similar type of problems in education, they tried to explore the option of using virtualization for their situation. As they are using two operating systems, windows operating systems and Linux. There are two types of technologies that can enable both of their operating systems Microsoft's Virtual PC and Vmware.

University has membership agreement with MSDNAA, so they can install any number of Microsoft's Virtual PC copies on lab and student machines. That's why they chose to use Microsoft's Virtual PC instead of Vmware. Later they said that they found it very easy to use, and they require a very short learning course. It should be possible that it doesn't have the flexibility of other virtualization software packages, but it was satisfactory for them because it meets their requirements.

They made necessary changes to their course so that courses should incorporate virtualization technology. They also tested all the assignments again an again and validate the virtualization approach and they worked with many of the issues that arise for the use of virtualization in our environment.

They prepared four virtual machine, Windows server, Windows client, Linux server and Linux client. Now student need only to install Microsoft Windows and Virtual PC on his own drive and then, they can load four virtual machine. All of the four virtual machines are composed of four files; the first of them is .vmc file. This file contains ASCII XML code, which describes location of other three files, their names and setting of virtual machine. The other three files are vhd files and they are virtual hard disks. All these files have the flexibilities of expanding automatically; it means that these files grow as need according to data written to these files. Operating system itself is first of these virtual hard disk. To ensure compatibility with the grading system this first hard drive image should remain same on all of the lab machines, so for this purpose they made this image write-protected. They used the second drive so that I can contain all

of the temporary information for example swap files, /var, and /temp files. The purpose of this second file was to minimize the size of the third drive. The third file is a kind of file that is used as differencing disk, so its main purpose is to logg all of the changes made to first main disk, as this file only stores changes that will be made to main disk, that's why it is a much smaller file, and can be sent to other lab machines for grading purpose. It also many other side benefits, like it ensures that every student starts his assignments from scratch. They got a problem of distributing all of the four virtual hard drives to 60 students, as each of the virtual hard drive size exceeds a gigabyte. But department solved this problem by purchasing machines with DVD-ROM drives. Otherwise they have to use the second option of loading images across the network and they have to split images into different pieces and then re-assemble all the pieces. But with DVD-ROM drives machines they initially created DVDs for distribution and then they installed the Virtual PC software, and each student need only to copy the virtual machines to their hard drives. They distributed their own .vmc files i.e. XML configuration for each assignment and they also distributed virtual hard drive .vhd files, over the network as, these files are small in size. This gave another opportunity to students to configure their virtual machine settings. As they have virtual machine images available, so it provides a quick recovery process in case there will be a problem.

Most of the lab assignments of system administration course require two or more then two systems connected to each other. This communication was made possible between virtual machines through virtualization by simulating a network of virtual machines, but for security reasons this virtual network was isolated from the campus network. During the semester, virtual machines are supposed to form their own infrastructure, every virtual machine will provide DHCP, DNS, file sharing and many other critical services that are necessary for the lab assignments. As there are many assignments that have high risk by doing in a regular lab, now through virtualization, these assignments can be done with out any risk of service interruptions on the campus network. As now through virtualization a single student can use a single host to run simultaneously both client and server (virtually), so now student got the opportunity through this virtualization network to do many of the group projects and assignments by themselves.

By grading and instructor point of view this virtualization system has an important impact. Now department don't need to higher an instructor or trained lab monitor to examine a live system during lab hours, now students are required to submit their work every week to the grader through the network. Students differencing disks can be swapped in and out of the grader's virtual machine configuration very easily because the grader's base image supposed to be identical with the students, having a slight difference each time when the system will boot. For the modification of the XML in the .vmc configuration files and also managing students images they have developed tools. But they experience that still even a manual system will be much more flexible as compared to system they developed.

Lab assignments and group project exams are very easy to conduct, examiner just need to prepare and distribute an image to each of the machine before exams starts, and student have a network drive, so that they can save their work at the end of the exam.

A greater benefit they got is that now instructors can change lab assignments very easily, so they decide to have assignments of basic installation level to the higher level

of troubleshooting. Instructors can create system having known problems so that students can experience with these type problems and try to fix these problems.

With this virtualization system also they also experienced some of the drawbacks; of the most mentioned is the increase in cost. As lab machines with a minimum of 512 MB of RAM are required, and their current systems average RAM was 256MB. So department purchases new machines to meet this requirement. The second drawback would be cost of the Microsoft Virtual PC, as it's a costly commercial product. But they did not cost much for it, because of the department's membership agreement with Microsoft's MSDNAA, which allowed them to install as many copies of Microsoft Virtual PC as they want. But for institution with this special agreement this virtualization package is very expansive.

As virtualization isolates students from the hardware issues, so this is another major drawback of virtualization approach, though virtualization improves many of the logistical problems in the field of system administration education, but it's also not the representative of real environment. Because a large part of the system administrator's duty is addressing many hardware conflicts, and through virtualization approach students can have the deficiencies of hardware skills. But according to their point of view hardware components of System Administration are increasingly less important and they are more of a technician work.

4.2 Case Study: Virtual Laboratory Usage in IT Security Education

In this case study we will discuss how a Virtual laboratory is successful for providing IT Security education. As we know that IT security is becoming a very famous field now days many research areas are opened for researchers in this field, but successfulness of IT Security not only depends on the technologies, but now a days it highly depends on the knowledge of IT Personal and the level of IT Security education they got [3]. In this case study I want to explore and discuss the concept of virtual laboratory and its application in IT Security Education. Providing It Security Education experience in regular labs is very difficult, because it needs a lot of resources like a dedicated test bed Networks and a lot of Administrative efforts also. Due to these recourses providing Security experience is very expansive. Virtual Laboratory is built with virtual machines and it provides online security laboratory. Virtual machines used here consist of a lot of rich security tools and network interfaces and these virtual machines are assigned to user as laboratory platform. This virtual laboratory can be managed in a reliable way, as virtual machines are under monitoring and administration. This reliability of virtual laboratory makes it possible for running on the internet. Experience within the Tele-Lab "IT-Security" project successfully proves that the concept of virtual laboratories effectively eliminates geographical and financial limitations in traditional IT Security education.

4.2.1 Introduction

As we said the successfulness of IT Security not only depends on the technologies, but now a days it highly depends on the knowledge of IT Personal and the level of IT

Security education they got. It has been noticed that many of the attacks succeed due to the lack of knowledge about vulnerabilities of systems and often they do not know how to defend against attacks. That's why many universities now a days providing IT security courses and use to develop specific security laboratories for preparing students to have hands on practical experience of IT security. Although students need a lot theoretical knowledge about many aspects of IT security , but its also important for student to have hands on experience for security exercises on real world systems, using real world tools and practice how to solve practical problem in the field of IT security.

However, compared with other topics in software teaching, providing security experience has been found particularly difficult by conventional means. Firstly, dedicated laboratories are needed. This introduces big administrative problems, e.g. preparation for exercises needs many efforts to install systems and to prepare security tools. Secondly, students might frequently cause system errors because super-user rights have to be given to them in some security tasks. Then, recovery from failures is needed. It is difficult to maintain such an unstable system in practical use. Moreover, due to financial reasons, not many institutions can afford a dedicated test-bed network. Considering security, students might misuse their super-user rights. This leads to serious security risks. The laboratory network has to be physically separated from production networks, which limits its application in a local area and makes on-site training expensive. They have been researching and developing new tools and methods to combine security laboratory with the electronic tutoring systems to facilitate e-learning activities. In this effort the first experimental system they developed was "e-Learning Platform IT Security" (LPF) That was developed in 2002. This tool was used to teach student practical security technologies and skills through a computer based training system. Student can experience with practical exercises using this tool to become familiar with the IT security tbasic technologies. Its very easy to run, just a stand alone Linux machine is required that should be equipped with some basic open source security tools. As these tools are open and free that's why they preferred them, so that students can easily to understand all the basic technique. LPA is a laboratory platform for student where students can complete their security exercises using various open source security tools. It also provides ease for the examiner because all the exercises submitted by the student are automatically prepared and evaluated, so it reduces a lot of tutors work. As its not running in a reliable mode, so it can cause some problems. Because users exercises may have serious errors that can easily corrupt machines. So to recover from the failure increases a lot of burden of administrative expense.

Due to the failure risk later they developed Tele-Lab CD which integrates the entire LPF system, due to this system they got rid to rely on hard-disk, Tele-Lab CD can detect common hardware and can run a Linux completely without hard disk. Due to this achievement now use can use it on any PC, without any fear of hardware and software failure. As compared to LPF Tele-Lab CD can be considered to be more reliable, but on the other hand it's very difficult to do big security task through its exercises, due to limited space and the nature of its local usage.

The concept of Virtual Laboratory on IT-Security came in their mind when wanted to support distance education. Virtual laboratory is an online security laboratory built with virtual machines. These virtual machines simulates real machine on a host. These virtual machines have all necessary IT security tools and network interfaces, so these virtual machines can be used by users as a laboratory platform on internet. As we

know that management and monitoring of virtual machines is very easy, so through virtual machines they achieved reliability in operating laboratory, and due to this reliability providing exercises over internet became possible. Experiences with Tele-Lab "IT-Security" shows that the application of the virtual laboratories proves to be very valuable as they eliminate geographical and financial limitations, which was experienced in the traditional IT Security education.

Previous Developments

The design of LPF based on a web structure, the architecture and components are illustrated in Figure3.1.

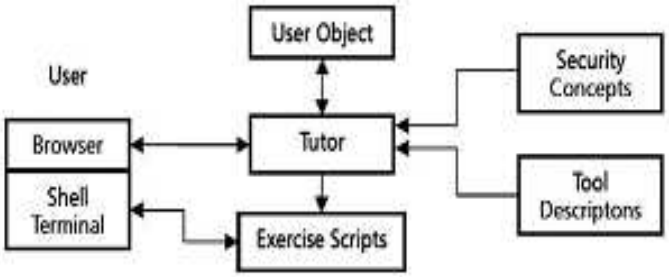

Figure 4.1: The LPF architecture

There are three main objects of the LPF architecture. User object, tutor and exercise scripts. The user object contains student's personal information and learning records. The tutor is a web server which presents teaching materials in the form of web pages to the student. This tutor object is responsible for the execution of scripts to prepare laboratory environments, for preparing student exercises and also to produce a quick evaluation of user completed tasks.

The user interface side consist of a web browser and a Linux shell terminal. When ever a user will log, he will log into the Linux system then he can apply security tools. There are two main connections between LPF and user. First connection for the user data to the web server and second for the user to make changes in the Linux system.

To fulfil some security tasks like security scanning LPF has to assign a user, a super-user right , but in this way a user can miss use the his right and this can lead to corrupt the system But they have backup partition, if it happens the administrator has to reboot the machine using backup partition.

With the new Tele-Lab CD, they made system installation and configuration protected and easy to be restored.

The concept of Virtual Laboratory and Virtual Machine

The virtual laboratory concept introduced by Tele-Lab "IT-Security" proves very valuable, because with virtual laboratory concept they got rid from the limitation that users can only complete their exercises on the local operating system, as with virtual laboratory the concept of physical machines was replaced with virtual machines on one host server. Through this achievement now security exercises can be done by the students from any where via the internet, as now they moved the entire security laboratory with tutoring server to the internet and it has the flexibility to share it among the remote users.

The concept of virtual machine is that, it's a fully separated copy of the hardware's of the physical machine. That's why many users can be given the rite of having a dedicated physical machine through virtual machine concept. As we know that virtual machine is a software application so, by running multiple copies of that application, many virtual machines can be run on a single host machine. Virtual machines can also be connect to the internet and a network of virtual machine can also created, as every virtual machine can be assign an IP address, hence a virtual machine is full fledge a physical machine for the user. If any type of destruction occurs in the virtual machine then it will not affect the host machine's hardware. That's why it's very safe to grant users the super-user rights which are necessary for many security exercises. Virtual machines can be classified into two main types according to the platform on which virtual machines runs, and they are IBM's VM and VMware, both of these can be directly implemented on the physical machine. There are some other which are implemented on the top of the operating system, like, User-Mode Linux (UML is also an open source software and it runs a virtual Linux operating system on a host machine. UML virtual machines runs just as process of an application, so they very easy to manage.

Virtual Laboratory Architecture

According to working functionality of the virtual laboratory they divided the architecture in to three parts, user machine pool, target servers and a control centre [3]. These units are shown in the figure3.2.

User machine pool can run as many virtual machines (VMs) as they needed. A user can only get a dedicated virtual machine when he will be logged into the VMs placed in the user machine pole through a proper login. In this way they are providing a very nice working ground for the students to work on their exercises. They also configure the VMs in such a way, so that size of the file system can not exceed the limit, because it's very important to keep in mind the recourses of the host machine on which virtual machine runs. To achieve this, they divide file system into two main parts, first part which they called local part which holds only kernel of the operating system and some others important programs. Heavy software applications and all security tools will keep in second part of the file system, which they called external part. This external part of the file system will only be executed one when the virtual machine will start, in this way they got efficient performance of VMs because due to the smaller size of the VM they can start, close and recover destructions through virtual machines in a very fast way. A remote execution interface, that's a secure shell (SSH) server is installed

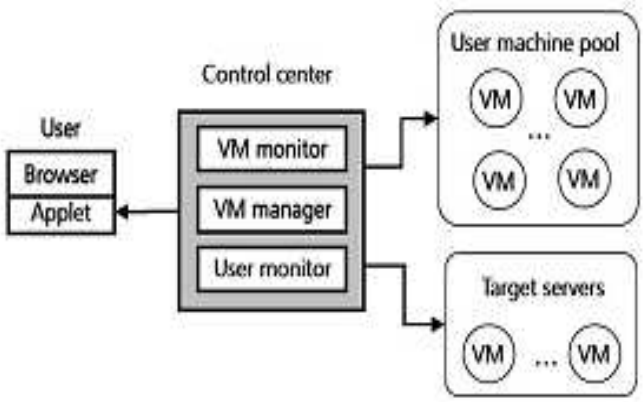

Figure 4.2: virtual laboratory architecture

on each VM. Through SSH one can get a secure remote access to the host. This helps tutor for the evaluation of the student's results on their VM.

Most of the exercises about security need only a virtual machine to complete, but there are some exercises that can't be completed only on a single VM in the pool and to complete them they needed to have a network environment or a target server. These types of exercises that they have are like scanning or attack simulations, for this type of exercise they need a dedicated server which much has vulnerabilities so that the user can get effective results. They prepared these target servers with virtual machines. As there are many risks to complete these exercises, so they kept them separated from the user machine pool. Any crash in the target server will not affect the virtual machines in the pool and its very easy to through VM manager with in time.

They also kept a management tool which is called control centre on the host machine, this tool is very important because it continuously monitor the laboratory status that's make this system reliable because all type of failure can be recovered in time. It has three modules, a VM manager, a VM monitor, a user monitor.

VM manager assigns VMs to users, it starts, close and recover the virtual machines. It also keeps track of the record that which virtual machine have been assigned and to whom, which machine is free, which is in use and which machine needed to be recover. This is possible through each virtual machine has modes, free, assign and recovered, and these entries are kept in a table which is updated by the VM manager while virtual laboratory is in running mode. A virtual machine can only become a workstation for the user, when it enters in the assigned mode. Recovered mode of the VM shoes that the virtual machine is in transitional mode. After the defects have been found, the virtual machine is marked as "recovered". It means that this VM is to be recovered now, so a recovery procedure needed to be called, which runs in the background. All the processes of that VM are stopped and a backup of the file system is kept. When the next time virtual machine is started its mode will automatically be set as free.

The function of VM monitor is to keep track of running VMs and attempts to find

errors and reports these errors to the VM manager at run time. There some necessary services of each virtual machine with out which a virtual machine can not work properly, VM monitor scans these services on each VM. It also checks the connectivity of the VM, if any of these has some problem, then that VM will not work properly and recovery of that VM will be done in the background.

User monitor is very helpful for the prevention of unnecessary occupation of a VM. Users input through the keyboard and mouse are detected continuously. This is done through a detection agent. This detection agent measures each VM's idle time period, they have a threshold time of about half an hour, if any VM accedes this limit, then they are reclaiming the VM from the user by automatically logging out that user.

Compared with other VM solutions, the User-Mode Linux is resource-friendly virtual machine software. Therefore, they decided to choose User-Mode Linux to build virtual machines. Its root file system is not bigger than 300 MB. As many of the applications including X-window based programs and security tools are installed on the external file system which is imported from the host. Each VM is started using a small copy of the root file-system, called the COW. COW files in the UML only record changes to an original image, therefore they are very small andeasy to be restored.

System Applications

As the virtual laboratory provides an infrastructure for the security exercises designed and tested in asecure way, so Tele-Lab "IT-Security" is applicable for providing practical IT security education. As it has been seen that Tele-Lab "IT-Security" designed in modules, so it is very easy to insert these modules into the system. The Tele-Lab "IT Security" is a complete e-learning system that can be used to provide practical IT security education in much safe and secure way.

IT security exercises has three main phases. In first face the basic concepts and principal of security related topics are described to the students. In the second face they use to introduce students information about specific security tools which are related to the topic. In the third and final step students are asked to perform exercise with these tools. All the activities of the students are recorded at each step in to each student own object, so that when a student next time logs on, he should be able to continue from the point where he stops last time.

There are three main activities that are included in the learning process and they are user sign-on, user sign-out and exercises. For logging users in to the systems, the necessary steps include are user authentication, user object should be loaded, then a virtual machine request is sent the control centre, VM manager finds an available VM from user machine pool then at end web server sends java applet to the user's browser. In this way the desk top of the virtual machine is then shown in the browser's window.

When a user sign-out the Tele-Lab, then first of all the user object is updated to the database, the java applet window is closed, the virtual is reclaimed, VM manager recovers the used VM and assignments table is updated and then firewall configuration is updated so that the access permission of the user can be reclaimed to the VM.

Exercises

The entire security exercises tasks are usually finished in three phases. First of all the working environment on the VM is configured, scripts of tasks are activated so that questions can be shown on the web pages and in the thirds face the after completing the tasks have to submit his results, these results are evaluated by the tutor, during preparation of tasks if any type of failure occurs, then help pages are also provided for the information to find solution to these problems. This mechanism provides flexibility for the student, that a student can repeat his exercise until he can't find the correct solution.

There might be some unusual events, and control centre is also responsible for managing these type of unusual events, these events include inappropriate sign-out, machine is running idle for long time, or a virtual machine can be corrupted. As they have VM monitor and user monitor, who are observing all VMs running status therefore as a VM failed, it automatically be detected and recovered.

To test the system students from the department made tests using this new system. Those students did not take security courses before. After the test department requested feedback from them, feedback shows that theoretical part and exercises are very easy to understand and complete without any help from the human instructor. For those students who don't have any back ground in Linux Tele-Lab provides necessary information in all exercises for those students. It has been seen Linux based platform brings some problems for new students, but its not a hurdle for learning process. It's obvious that virtual security laboratories can not completely replace traditional dedicated security laboratories. Security exercises which need a big network environment or those exercises with need a long run time to complete are very difficult to implement in Tele-Lab security laboratory. But still Virtual laboratory is an efficient platform to run practical security exercises on the internet without any risk with less resources, which in turn reduces the over all cost.

4.3 Case Study: An Open Source Virtual Lab used by The University of Milan

4.3.1 Introduction

In this case study we will discuss an open source virtual lab developed and used by the university of Milan Italy, in an online degree course of "Security of Informatics Systems and Networks". As many E-Learning is becoming very popular for the information Technologies degrees. There are a lot of universities that are currently providing many services through E-Learning, like Video lessons, online exercises and exams, having an interaction between teacher and student. University of Milan is also one of from those universities who are providing E-Learning services to the students, but still they said that they need a virtual environment system that can provide students hands on practical experience with in network programming and configuration. To meet this requirement of the university a group of researches designed the virtual lab. So those students can easily experience on network programming and configuration through a complete training environment which is accessible via net using normal web

browser. In this case study we will describe briefly their open source system of virtual environment laboratory and discuss how it helps in online teaching.

As we know that Information Technology education mostly focuses on practical experience of the students in the laboratories, particularly courses like networking, network configuration and network security are those courses which are totally dependent on the hands on practical experience of the students. So providing hands on practical experience my using different technologies and dealing with exercises of network configuration and network security are most popular issues of the current age. Information Technologies Education mostly depends on laboratory activities. That's why most universities, are using security laboratories, through which students can easily get practical experience of security technologies, network programming tools, networking configuration and many other. It is the internet technology through which, it is now possible for the universities to offer courses, not only theoretical courses but also laboratory based practical courses on the web, and these laboratories are providing full functionalities through all mean of a conventional laboratories. There are many issues that are still to be resolved for example, conventional laboratories have the flexibilities to be implemented on an isolated network, and this isolated network allows students to work without any fear on all risky exercises like network programming, network configuration, firewall design, etc.

Though it's obvious that this type of isolated network prevents some dangerous operation by the students and prevents destruction of networks, but there are many problems also for example n case of system failures, which can't be recovered through internet or through virtual laboratory and can only, be recovered if you are working in a congenital physical laboratory. A lot of resources required and for most of the issues its necessary for the virtual machines to be used with administrative privileges, this is not easy to implement through a virtual laboratory and it makes virtual laboratory difficult to develop. No doubt it is very important in case of remote education and e-learning experience, particularly in the Information Technologies related courses. This is the main idea with e-learning process that defines the possibilities for providing online university degrees to the university of Milan and Italian legislation. This is a unique online course that provides students opportunities exploring different issues related to security and provides hands on practical experience through e-Learning experience with the full support of virtual laboratory. This is not just that university is providing course related material with e-Learning experience. It's not only just a normal online version of the university course, but a full fledge online focused course having completely new redesigned teaching context. The e-Learning model they used for this purpose, they called it "blended" model, which has the flexibility of different teaching scenarios, for example: traditional lessons, online video-lessons, forum activities, online exercises and laboratory exercises. This was the reason they thought that they will develop the Virtual Lab project that will only base on the open source technologies. The Linux distribution they used for this purpose was Gentoo Linux, and they develop this fully open source system using Xen platform. In this way they get rid from almost all problems that arise due to virtual networking laboratory and networking programming and configuration on a fully virtual network using efficient Linux scripts. In this case study I am going to examine this fully open source virtual network laboratory solution of e-Learning of the university of Milan courses of information technologies.

4.3.2 Background for This System

The purpose of Remote laboratories is the representation of traditional in-site laboratory experiments to distance learning through e-learning that offers remotely accessibilities of resources, equipments and instruments used for real laboratory. A new technology for e-learning is a virtual network laboratory, which uses a simulation system that has the functionality for replacing a real network laboratory. Mostly Virtual laboratories were made for the purpose of simulation system. There are many simulation software used so far such as, Matlab, LabView or TeleLab, a web-based training system for information technology security that is discussed in earlier. These simulation systems provide students a fully virtual network of virtual machines for particular scenarios. This doesn't mean that these software are only used for virtual simulation, there are many cases where these software were used for a real system. To day this remote, virtual laboratory experiments has been used in various scientific a technical issues like automation control, electronics, chemical and mechanicals and in robotic systems [4].

4.3.3 Techniques for Virtualization

The simple definition of a virtual machine is that, is a copy of underlying real machine that is running in a simulated environment. This simulated environment usually consist of a virtual machine monitor, which is a software component, the purpose of this software is to present a layer technology, which should be similar or compatible with the hardware, and also the management of different virtual instances and controlling access to the system resources is also done by virtual machine monitor. It is the achievement of this hardware replication, that those applications which are running on a virtual machine are completely separated from the applications running in simulated environment. So all the applications can have a environment and it could be created and achieved through many different devices, there are a lot of virtual devices that can be helpful for populating a virtual environment such as virtual disks, emulated processors, virtual network adaptors and many more. So using these types of virtual devices to create virtual environment ensures the virtual machines isolation.

The task of allocating resources to the virtual machines is fully controlled by Virtual Machine Monitor (VMM). All the applications running in simulated environment have their own resource container allocated by virtual machine monitor, and it's impossible for any application running in a virtual simulated environment to access or use those system resources that are assigned by the VMM to that VM container. Two main approaches to virtualization are: full virtualization and Para virtualization.

The main purpose of the full virtualization approach is to construct a virtual execution environment, which should have functionality of running full images of the operating system. It means that it should run an image of the original guest operating system behaviour and it should facilitate on host machine. This approach has a drawback, that to implement this type solution is very slow, because it needs to have a real complete system behaviour and to got resource isolation a software component is required. Products that offer full virtualization are VMWare, Bochs, and QEMU. Para virtualization approach describes performance problem, Para virtualization approach requires that guest operating system must be modified in order to run the virtual en-

vironment. For this approach the operating systems resources should be available, because there is need to make changes to the operating systems kernel. It has been seen that in some emulation system the approached have been mixed, so that the best of both approaches can be used.

4.3.4 System Structure

The basic requirement of an e-learning system is to make sure that the e-learning system is accessible on the web through the internet and that it should have communication channel for the students with full privileged level having functionality to provide student's online tutors and teachers facility. To fulfil this requirement of the e-learning platforms should have a structure that can provide all the necessary services to the students and, can connect them to the resources. So achieve this, e-learning platform should have a three- tiers structure that should consist of a user interface, a user web browser and a connection to the server. The e-learning platform they are using, they called it CDL online e-learning environment follows the above three-tier structure.

The system consist of three main component, the first one is the e-learning platform, this e-learning platform makes possible for the students to access course contents and the virtual lab, and make possible for the students to have an interaction between teachers and tutors. The second component of the e-learning system is the Virtual Server; this virtual server consists of a virtual machine pool, the server that implements virtualization and firewall. This virtual server actually an informatics laboratory and this laboratory provides students a real working environment. The third and final component of the system is a student web browser, through this web browser students can connect to the platform and can interact with it, the browser runs applet to connect, using an ssh connection to the assigned Virtual Machines VM. The identification of each user is made possible through the user name and password, though this identification a user can get access to the platform through the web browser.

Services supplied by the Cdl e-learning platform can be described in three main categories. In first category we have Communication services, these services are responsible for the communication of students to their coursemates tutors having dedicated platform. In second category we have Community services, community services allows students to find profiles of teachers, tutors and also the fellow students. In third category we have Teaching services, these services grant access to the actual contents like to the class exercises and video lessons only to those courses in which students are enrolled. The fourth category of services is called Calendar services, these services are used to inform students the necessary deadlines, so that students can finish their work in time.

With teaching services, their main focus was to implement a virtual lab, so that students can connect and interact with a virtual machine having all administrative authorities and privileges. Its very easy for the students and they could access it just following the specific links that are provided in the platform, through these links a student can connect with the virtual lab server and can populate a ssh shell on his Virtual Machine. A firewall is used between the e-Learning platform and the virtual machines, this firewall is responsible to filter and manage all incoming and out going connections, to ensure the security of virtual server and it also isolates virtual machines pool from the out world.

Each Virtual Machine have its own scaled environment, students have the access to modify the entire Linux System configuration. They also have access to interact with all the virtual machines, but these machines should be active at the same time. A ssh connection allows students to communicate with their virtual machine only through predefined ports. They shaped their teaching activities according to the virtual lab. They have different type of exercises some of them consist of making applications about network programming, for example Socket and RPC (Remote Procedure Calls) libraries. These exercises allows students to try getting practical experience what they learned in the lessons, for example they have exercise like firewall configuration system, routing table related problems and some about networking interfaces.

Through this system now students not only learn the theoretical material about networking related issues, but they will also face the real world systems and they have the opportunity to get practical experience about all real world problems that are necessary for their field.

4.3.5 Framework of Virtual Lab

The system framework of the Virtual Lab they are using focused on three main parts. Hardware of the Virtual Server and the Firewall, virtualization software and the Virtual Machines. We will examine and discuss these aspects one by one.

Hardware

For the implementation of a reliable and scaleable virtual environment, the hardware requirements that was necessary for their choices about hardware were basically depends on two things. A storage unit that should have the flexibility to give students necessary storage that should reflect a complete developing environment and that should be suitable for all courses. To make possible for this system to manage as mush virtual machines as students needs at the same time; the system could need many additional software, a big RAM memory etc. To fulfil these requirements they decide to implement their virtual server with these specifications: a Fujitsu-Siemens Primergy RX-300 S2 with 2 Intel Xeon EM64T CPUs at 3.20Ghz, 8 Gb RAM memory and four 300 Gb SCSI U320 hard disks in RAID 5; [3]. They connect their server with their internal net using a broadband NetXtreme BCM5721 Gigabit Ethernet PCI network interface. They implement the firewall on different machine, so that the performance of the virtual server can't be decreased and also to increase the system security from the out world attacks. They connect the firewall to the external net with an Intel Corporation 82541 GI/PI Gigabit Ethernet network interface.

Currently the maximum numbers of virtual machines that are running at the same time are round about 90, and they occupied storage of about 300 GB. This specification allows the university to have full management of the students not only those who are studying currently but also it has the flexibility for future usage.

Virtualization Technology Used

They decided to use the Xen as virtualization platform. Xen is an open source virtual machine monitor that has the functionality to support para virtualization approach.

Xen basically supports x86/32 and x86/64 platform. The hardware CPU virtualization concept provided by Intel VT and AMD virtual machines technologies, Xen have the functionality to run a full guest operating system kernel as it is. As we discussed that Xen need to port the guest operating system kernel to the environment of x86-xeno architecture, it has the functionality of achiving performance close the native hard ware because of its Para virtualized virtual machine monitor. Xen system consists of several layers, each layer executes in an isolated environment, this virtual isolation environment is called domain. Xeb has a hypervisor that is responsible for the scheduling operation that are necessary for the execution of every domain, where as guest operating system is responsible for management of every virtual machine scheduling. As the system boots, a domain with basic privileges, called domain 0, created automatically. Domain 0 is responsible for the initialization and management of other domains and virtual devices. This special domain is very important domain, because all the management and administration tasks are dependent on this domain. There is special process in Xen called Xend, this process also runs in domain 0 and is responsible for the management of virtual machines and through this access to console of a virtual machine is possible. Some important scenarios in which Xen has been used include, configuration and management operating system and of networking related configuration task, kernel development, consolidation of server and their resource allocation.

Virtual Machines

For the implementation of a virtual machine, after consideration of hardware and virtualization technologies used, it also needs a further analysis that based on some important consideration. Their goal is to develop an efficient virtual machine, and to achieve these virtual machines should be isolated duplicate or be a copy a real system. In order to protect virtual machines from possible faults that can cause VM failure, every virtual machine should work in a sealed and independent environment, to achieve this every virtual machine should have isolated disk and memory address space. Second in order to provide students an environment in which they can easily develop simple systems and can get hands on practical experience on network programming and system configuration, they have to provide students operating systems that should be up to date. There are also some hardware constrains that they should consider, particularly in 64 bit implementation of the server and with the support of Xen technology there is a restriction for some operating systems. Keeping all these points in mind and basing on their experience they decide to implement their VMs Gentoo Linux distribution. Gentoo Linux distribution has a lot of unique characteristics that are essential for virtual machines they want. One of the most important features of Gentoo Linux distribution is its high performance adaptability, Gentoo built on Portage technology and this technology performs some important key functions, some of them are: software distribution system, through this system student don't need to install the whole system, it allows students to install and compile only those component that are needed and these packages can be added at all time with having burden of reinstalling the whole system. The second important feature of the Portage technology is that packages can be built and installed; portage technology has the ability to build a custom version of packages that should be suitable for hardware. Gentoo has the ability of automated updating of the entire system, is fully open source and has support for their 64 bit architecture and

it implements the Xen environment very easily.

4.3.6 Implementation of Virtual Lab

They addressed three major steps of the implementation of Virtual Lab, and they are Network configuration, Firewall configuration and the Platform connection. We are not going to describe these steps in details, as specific description of these steps is out of scope for my topic, how ever I will give a short description.

Network Configuration

The first stage that comes in configuration of Virtual Machine Network Configuration is a virtual machine image. As each virtual machine consists of a root image, this root image shows maximum available disk space and it also holds the operating system and all the necessary installed packages. This root image also contains a swap image, which is responsible for the management of memory swaps. For the installation of the Gentoo Operating system they created a 2 GB image, this image is composed of all necessary services like, a gcc compiler and many essential tools such as iptables, text editors, Perl and Python. Each student have its own root and swap image, these images are created automatically for every student using a shell script, by reading each student id. In next stage they are creating Xen configuration file for each Virtual machine.

Firewall Configuration

The purpose of the firewall in the Virtual Lab project has three important benefits, first of all it provides server full protection from all kinds of external attacks, second it provides isolation for the Virtual Machines from the external net and third one is responsible for the establishing a connection between student computer to the relevant student virtual machine through e-learning platform via ssh. As each virtual machine is assigned a specific local IP address, so for some one to access a virtual machine from the external world, the request should pass through the firewall configuration.Port numbers are used for the connection of identified virtual machine of a particular student. On the basis of this port number, rules are written in the firewall, who forward incoming ssh connection to the specific local IP.

Platform Connectivity

The last part of the Virtual Lab configuration is to define the communication protocols to make possible the connectivity between e-Learning platform and the virtual server. As we discussed that the students get access to the virtual machine through e-Learning platform via ssh protocol. To make possible so that server performance remain increased, the process of Loading virtual machines is not done at the boot time, so that virtual machines can only be started when requested by the platform through ssh protocol. The platform have to call a script to initialize and make necessary arrangements to start a virtual machine, each time a students wants to activate his virtual machine. The same procedure repeats when a student wants to close his connection, again platform have to call script that makes necessary arrangements and shutdown the Virtual Machine of that student. The scripts they are using accepts two types of parameters,

the first one should be either start or stop, depending on whether the virtual machine is going to be start or stop and the second parameter is used for the identification i.e, an id of the student who is requesting the virtual machine.

If we look into the script, then we see that in case if the first parameter is start, then the Xen command xm in order to load the virtual machine that is declared in the configuration file and that also satisfy the second parameter which is identification. Where as if the first parameter is stop, the system calls a Linux command losetup to close the virtual machine. The direct student's connection with the virtual machine can be established, as the platform finishes starting the virtual machine that should be correct and according to configuration. After this an applet starts that's provides the ssh shell; for this purpose the applet they are using is called MindTerm , is a java applet that has ssh1 and ssh2 protocols.

4.4 Case Study: Virtual Machine at Ume University Sweeden

4.4.1 Introduction

At the university of Ume Department of Computer Science Virtual Machines are used to gather with a set of virtual assembler to teach students architectural style of different type of computer processors [5]. Students use to write programs in the virtual assembler and these programs are then compiled the virtual assembler used by the student for writing program. The purpose of using virtual machine so that student can check their programs step by step during execution, that's why programs written by students are then executed using virtual machine, in this way students have the flexibility either to examine the execution of their programs step by step or at once .

Computer architecture is one the core and important course at the department of computer science Ume University that provides student in-depth knowledge of different architecture style of computer processors. This is a mandatory course for the MSC graduate students of the department and is also offered as optional course to the student in undergraduate level computer programs.

4.4.2 Problem

The department faces the problem of rapidly increase in the enrolments of students every year. Last time in computer architecture course they enrolled about over 100 students. Due to this huge enrolment of students they realises the need to change the assignment for the students, this is because of the resource limitations. The first assignment they gave to their students was about the development of a tool called instruction tracer that must have ability to not only execute other programs but also can collect important statistics about data, like size of data used in operation, information about registers used, block and cell basic length, and number of bytes used in instructions. They got problem with this assignment because this assignment require that all students must have access to virtual machines with MIPS processors. The department have total 28 SGI workstations and two SGI servers available, and to run the tests of assignments produces very high load on the machines, this high load is not acceptable for the machines on which assignments have to run. Department also planed to

develop a new assignment, because they wanted students to have hands on practical experience dealing with different issues of architectural style of computer processors. They have three Unix platform SGI Irix, SUN Solaris and IBM aix. They created a virtual machine that runs on all available platforms and also together with virtual assembler, which is responsible for the conversion of assembler code that is written on any architectural style to the architectural style of target machine assembler.

4.4.3 Agenda of new Assignment

As it is discussed in above that they want to prepare new assignment for the students of Computer Architecture course and these assignments should have to meet their goals. They set three main goals for the new assignment. The first is that new assignment should give students necessary task, so that they can have well understanding about the behaviour of different computer architecture, the second main goal is that it should be easy for the students to implement and use, and the third one is that it should have ability to be portable to all the Linux platforms available at the department except to Windows based platform and for the successful implementation of the new assignment all of these goals mentioned should be achieved.

4.4.4 Solution

The hardware solution failed, because hardware for this solution is very expansive and then the maintenance of this type of hardware is also so difficult and require a lot of resources. Further more as there are very few machines with the architectural style of unorthodox system are currently produced in the market and machines based on accumulator and stack are very difficult to find. Due to these hardware limitations they required to think about other options for getting their goals. So the obvious root they have to follow is virtual machine. As the base of a virtual system is also the hardware on which a machine is supposed to be build, and they constructed their virtual machine system in such a way that when it executes a program it looks same as it executes on the real hardware. There are many reasons for choosing virtual machine. The one most important use of virtual machine is to test any new the hardware system before its actual construction to be finalized, this is done by simulation and the second most important use is that through virtual machines system it could be make possible for the system developers to have their own machine that would be available for them without restricting it for any one else.

4.4.5 Basic Virtual Machines Architecture types

Four basic virtual machine architectural styles are: Accumulator, Stack, and Memory-Memory and Load-Store architectures. One option was to develop a virtual machine with user interface specific for one architecture, but they did not adopt this option and they develop a system which is built of one general-purpose register architecture in a virtual machine and this is the only one interactive user interface and they are using four independently working virtual assemblers and their function is to convert the input they receive to the actual virtual machine assembler. In order to make possible the learning process efficient and fast, they created the virtual assemblers with very small

instruction sets. There are also some similarities in some parts of the assembler languages, but there are more parts which have no similarities some of them are branching or jumping, creation of variable, and instructions that are specific to the architectural style of the assembler.

The most basic processor architecture machine is called he accumulator machine. Accumulator based machines have one register which is called accumulator register, this accumulator register is used in all of the instruction as one the operands of the operation and also this accumulator register is used for storing the final result after operation, that's why this register is also called target for loads and source for all the stores.

Stack based machine doesn't have any general purpose registers. A stack is used that handles the data; this stack is based on last-in, first-out strategy. For all type of operations to be completed first of all the operands are taken from the stack and after the operation has been completed and calculation has been done, the result is sent back to the stack through the pushed operation. Only the result that is not pushed to the stack back is the data through the operation in which data flows from memory to the stack or from the stack to memory in reverse. Like the stack based machine memory-memory based machines also doesn't have any general purpose registers, but still there is difference between stack based and memory-memory based machines, one of the main difference in both is that in memory-memory based machine memory cells are used as operands for the operation where as in a stack based machines operands are taken from the stack.

The machine which has a fixed number of registers is called a load-store based machine. They are using in this case 32 fixed registers that are used as operands for all type of operation and calculation, and also the final result of the operation is stored in one of these fixed registers and like stack based machine only the operations whose results are not stored in one of these registers is the operations of transferring data between registers and memory and also in reverse.

In order to increase the usability of the system, they explored that all parts of the system should be portable, and they preferred to use the command line, terminal interface for achieving the highest level of system usability. They also decided to have one starting point for the system and also to have an interactive interface for the user to interact with the virtual machines. To meet these requirements of the system they developed a system of scripts, its purpose is to check the suffix of the given input file and to run the appropriate correct virtual assembler for the given assembler program file. The first one is the ac which is for accumulator and second one is mm, which is for memory-memory and third one is st, which is for stack machine. The resulted output from the virtual assembler is then passed into the appropriate virtual machine, which is responsible for changing its behaviour depending on the command line arguments that are provided to the main script. This main script has a program in the directories having the same UNIX architectures based standard, that's why this main script can handle different underlying architectures.

The purpose of the virtual machine is not only to execute the programme code, but it also gives a debugging interface which is of terminal based, in this debugger the current line of execution is highlighted with colures and all the variables are shown with their values , so its very easy for the students to follow the execution process of the programme code and to understand it. Not only this main script can also show

other things which depends on the suffix of the file which is provided as source file, these things should include accumulator, stack, lowest address and the registers. All the scripts are Bourne shell scripts and all the virtual machines are written in ANSI C, some parts for example memory-memory assembler, are also written in yacc and some parts are also written in lex. Some of the commands are functional commands, i.e., s for stop, n for next, g for go. When ever a stop operation is executed or an uninitialized address is reached the execution stops and the total number of instructions executed both in the symbolic ones which are in the input file and those in the general purpose virtual machine are presented to the user [5].

4.5 Case Study: Virtual Machines at University College of Oslo and the University of Amsterdam

4.5.1 Introduction

As the networks and system administration education is not that can only be provided to the students theoretically, that's why institutions have to face a lot of challenging problem in order to deliver students both theoretical and practical experience in the field of networking and system administration related courses. Some of the main services that are very popular for learning networking and system administration related issues practically require that student must be logged in their real dedicated system with root level of access, so that they can easily install these services and can get practical experience about them. We will explore in this case study how virtual machines networks can be beneficiary in education to provide the environment required for the students, which should not be expensive as compared to provide a real physical environment for the students to get practical experience. A virtual environment can be used to prepare more challenging students assignments and can also provide an environment which could be tested and protested for every single student and also for groups of students. This virtual environment is very flexible and has the ability to be scaleable in future with the larger classes. One of the most important function of the virtual machine environment is that configuration is very easy especially it has the ability to reconfigure a network and to restore it quickly in case of any destruction.

In this case study we will discuss a tool for the building and administration of virtual networks based on User-mode Linux . This a combined effort between the lecturers from the Oslo University College, University of Amsterdam, University of Linkoping and some of the others very experienced lectures in the area of networking and system administration [6].

Networking and system Administration is the process in which structure and configuration of a lot of networking equipment together with a set of computers required, not only this but also team of highly qualified and skilful people is also need in this process. For the students to briefly understand all issues around Networking and System Administration, its very important for them to get specific knowledge of experience, not only this that how these systems work, but also to know how peoples are suing these systems. In the field of Networks and System Administration only theoretical study can not provide all the necessary skills and understanding those are essential in the field of Networking and System Administration. So for the students of Networks

and System Administration its very important to get hands on practical experience of different areas like, configuration, deployment and maintenance of all networking and system administration related equipments, including computer systems. Though for any university to provide such an environment to provide students, experience of all necessary issues is very challenging. But on the other side it's very important for those institutions that are providing education in the field of Networks or system Administration to deliver good education to the students providing hands on practical experience of installing and testing services.

University College of Oslo and the University of Amsterdam, provide education in the field of system and network education, as the key for system administration education is to give students an environment where they have a realistic testbed to test services related to the configuration and management of Networking and System Administration related issues. In this project they also got help from other universities and from some highly skilled professional people in the field of system administration education. Their goal was to provide students and researchers a way, so that they can use and create virtual environment and solve their problems, installing all problems related to the field of network and system administration. So they explored two approaches for using systems in network and system administration related courses. The first one as file system; this file system configures itself on the basis on parameters that are supplied at the boot time. In the second one they build and uses a tool called MLN, it works on the basis of User-mode Linux in the form a configuration language. Their approach toward solving problem using virtual machines shows that, how space and money can be saved and can give more knowledge to students, through the creation of complex new assignments very easily with too much work. Not only this system helpful from the perspective of students, but also from the perspective of universities , institutions and system and network professionals, who are linked with system administration field either through education, or through working as system or network administration, because this system provides them a testbed , where they can install their application, all new technologies and configurations and can test them without fear of destruction or affecting the real physical network.

Introduction To Educational Context

With the passage of time as internet and information technologies is becoming in almost every field, that's why the networking and system administration field is becoming popular and important now a days. As technologies are changing very rapidly, so the education and training of system and network administration is becoming more and more important and also challenging. Some of the institutions and universities provide some courses related to the field of networking and system administration and only few universities provide full degree in networks and system administration, like Oslo University College and University of Amsterdam provides a unique degree in the field of system and network administration.

For constructing a network lab were students can get practical experience about their problem, requires a lot of recourses, like special rooms allocated only for this purpose and all hardware equipments necessary for establishing such a lab, also a large number of machine are also required. Some time also human recourses required for setting up labs for the students. A network Lab for system administration course

should have the following features

- Root password, or administrative privileges are required for some exercises.
- Flexibility in lab scenarios, so that lab can be suitable for every type of exercise and also flexible to change the exercise quickly.
- Lab should have ability in any way to remember the entire configuration, so that for the new students this procedure can be repeated to save the time.
- Lab should be scaleable, so that it can work for many students to work simultaneously on their assignments [6] .

Building such a lab has a lot of challenges, and very difficult to achieve all desired goals through this type lab. Some times there are conflicts between the courses, their exercises and lab setup, and also due to the available resources in the lab. Space and financial issues almost always remain in every institution and these limitations also limit the number of students in the lab at a time and the number of available resources. In Oslo University College teaching firewalls and intrusion detection course in lab consisting of 15 machines, running a simple network, allocating each student group a single machine experienced a lot of challenges, some of them are mentioned below.

- Allocation of a dedicated whole room
- To purchase new hardware is expansive, and also lab space is not enough to place new hardware, so lab had to be consisted of old hardware
- As the network used in the lab is a simple network, students realise that many of the tools can't be tested because of the simplicity of the topology used.
- Some of the hardware broke during the lab work, so department had to repair this hardware and this also applied additional burden on the department and the course staff.
- For rescue operations systems, students need to be present physically in lab,
- To build a lab and then to configure it on all the machines took a lot of time
- As with this Lab scenario, there is limited access networking equipments, like switches, routers, and number of machines, that's why difficult to scale this type of lab.

Virtual Network based environment can solve these problems, it has been seen that, through virtual network based labs environment, students can be provided more complex and up to date network topologies, for testing all types of services related to the course at very low cost then a physical base lab. A significant savings can be realized when the whole virtual lab runs on a single machine. This machine can stand in the college's server room and won't take up much space. There is no need for student access to the main server in order to recover from problems. And so, the old network lab room can now be reallocated [6]. There are also some limitations of using a virtual network lab, because some of the system administration tasks can't

fit into a simulated virtual machine environment, like configuration of some specific hardware or high performance systems is not easily simulated on a virtual machine based network lab. But still department and course staff realises that virtual machine based network proved very valuable in educational point of view, especially in the filed of network and system administration education.

Introduction to User-mode Linux

User-mode Linux (UML), allows multiple virtual Linux systems (known as guests) to run as an application within a normal Linux system (known as the host). As each guest is just a normal application running as a process in user space, this approach provides the user with a way of running multiple virtual Linux machines on a single piece of hardware, offering excellent security and safety without affecting the host environment's configuration or stability. UMlL runs and behaves same as a kernel as compared to instruction set emulator. UML is bound to Linux; where as other technologies like, VMware or Virtual PC have the ability to run as many operating systems as one want in a single virtual machine instance. But it demands additional resources and very complex type of configuration. As its possible to run many User-mode instances as normal application processes running on the same single machine. UML provides a separate kernel for each virtual machine instances, which allows each virtual machine instances to run different configurations. To set up UML network system and to maintain several instances of the virtual machine is a complex task, MUL has a mechanism through which it can automatically set up and maintain as much instances as required.

UML has many ways to connect a network of virtual machines together and also to connect this virtual network to the real world network. Some of these tools are like switch emulator, routing packages and virtual networks. Through the use of these tools virtual machines can be used to create complex network topologies. A tool called Linux bridging software can be used to create a link to the real internet, or to make possible each virtual machine to be connected to the local network. As user mode linux runs as command line, so many system properties can be given as command line option, through this many hosts an be started or stopped at the backend, as students can access the virtual machine using ssh over the network without logging into the server first, so it eliminates the need of each student's account on the server. UML has the ability to keep the file system and memory usage small, this property of UML allows a workstation to run many light weight virtual networks. For building networks of virtual machine, each virtual machine required to be configured individually, and there should be away for starting and stopping virtual servers, this task is very hard, if we have to do by hand. This is a configuration management issue and there are several approaches to deal with these issues. We will discuss in the coming section an approach used by the University of Amsterdam.

4.5.2 User-mode Linux used at the University of Amsterdam

Here we will discuss User-mode Linux used by the University of Amsterdam for teaching one of their masters programs in Network Administration. University has a course called Internetworking and Routing INR, which consist of lectures and practical as-

signments as well. This INR course mainly deals with layer2 i.e. bridging and switching and layer3 i.e. routing networking. For each student to have hands on practical experience with assignment of INR course, it is necessary for them to have dedicated network of multiple machine, routers and bridges. This requirement can not be fulfilled by the institution because of the expansive hardware. As we discussed earlier Linux can be used as a router as well as for bridge purposes. That's why User-mode Linux was ideal choice for them as with Linux it can replace real routers and bridges. INR course basically focussed on theoretical knowledge and it's not concern with specific practical experience, so using software or commercial hardware doesn't matter for them, only required that it should work efficiently. As UML runs the software which runs on the underlying regular Linux, this property of the UML makes suitable for them for approximately completing all task, to install network diagnostic software like tcpdump and ethereal is very easy with UML where as with hardware based solution this can not be achieved.

There was only one server available and it was necessary for the department to not give students root access for the systems and it was also necessary to keep file system usage as low as possible and it was difficult until students build their own network of virtual machines. They used a special filesystem driver for every Uer-mode Linux host so that they can control the usage of the filesystem and they called it hostfs. UML can access this filesystem from the host machine. Several UML instances can mounted at the same time using hostfs. Through this disk space can be saved, which results in less physical space required and that improves significant overall performance of the system. As hostfs driver mounts filesystems only which are read-only mode. Many Linux distributions take root filesystem as writeable and did not accept read-only root filesystem. For this course the Linux system were considered to be as routers and bridges.

Enabling features through boot parameters

Linux kernel is used to be customized in run time; there are many kernel parameters exist many of them are sude for hardware configuration. Some of the most common used parameters are setting up rot filesystem and srtatup runlevel. These boot parameters are then used to configure each of the UML instance by supplying on special parameters, these special parameters are parsed inside the running instance only once when it boots. All the assignments of the course differ from each other in the sense of which routing protocol is being used, and the total number of the network interfaces that are to be configured. One of the important features of the UML is that there is no need to modify configuration of any file of the host in order to pass all of the parameters mentioned above at boot time. When this feature is combined with one read-only filesystem then it gives us platform for building an environment where networks of virtual machines consisting of working routers and bridges can be crated saving space and a lot of hardware resources. A single filesystem based on BusyBox and only consuming around 20MB of space could now be shared among all students [6].

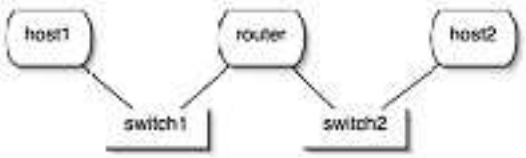

Figure 4.3: A simple UML network consisting of two hosts interconnected by two switches and a router

Creation of virtual networks

For each virtual network to be configured, students need to write a script. This script is basically responsible for handling the starting and stopping of the UML and uml_switch and then to pass the correct parameters to the virtual machine instance, which has to be start or stop. As these assignments was of master level, therefore it was up to the students, that they can use what ever scripting language they want to use.it has been seen that all students were not familiar with the scripting language, that's why student who were not familiar with scripting languages they got problem and students who were familiar with scripting languages they did not face any problem and they went very fast getting their required goals. That's why not all students reached to the required level of expertise many of the students face problems and they did feel comfortable with these assignments. Many of the students who faced problem with this part of the assignment, preferred to have a graphical tool, for the configuration of the virtual machine and the virtual network.

Assignments

First time "Internetworking and routing" INR course was taught in February and March 2004 [6]. The first assignment that was given to the students was just to explore User-mode Linux, as many of the students were not familiar with the UML, so the main purpose of this assignment was to give students opportunity to become familiar with UML and become comfortable with configuring and running networks of virtual machine. This assignment was very helpful for the whole class and particularly for those students, who where new with UML.

After the basic introduction with UML, students were given the task of translating networking diagrams into workable UML configurations with IPv4 number plans. After this week the students were given the assignment to add the support for IPv6 to the configuration they created previous week. After this week students were given

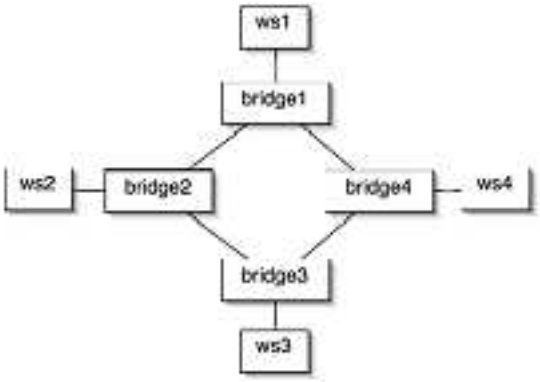

Figure 4.4: virtual network of Spanning Tree Protocol

the assignment to explore and examine the Spanning Tree Protocol, in this assignment students created a network consisting of 4 bridges, using the ring topology, having a normal host connected to each of the bridge. They configured the bridges not to use STP. Through this configuration as one of the machine tries to communicate with another machine, the network starts to be filled up automatically with recycling packets. Through tcpdump students can monitor this process from one the bridge. Students sued last two weeks to explore and examine some of the routing protocols. This first assignment of the routing protocol that was given to the students was to implement RIP and OSPF. The last assignment of the routing protocols that was given to the students was to examine BGP. Border gate way (BGP) protocol is an exterior routing protocol; this exterior routing protocol is used to connect multiple autonomous networks together. To simulate this UML also has this ability that can be achieved by creating an internet exchange. Through this internet exchange students can connect their networks together. BGP is used to exchange external routing information between the student's networks. This assignment proves very valuable, as it provides students fully technical and practical experience.

Student acknowledgement about UML

It has been seen by getting acknowledgement from the students about UML, that most of the students enjoyed while working with UML, and only few students proffered to work with real systems.

4.5.3 MLN

To improve virtual machine administration University College of Oslo developed a virtual machine administration tool called MLN (My Linux Network) [6]. The primary purpose if this tool is to provide such a management tool, so that it can provide help to all aspects of a virtual machine creation, running and configuration. Through MLN one can easily keep track of the different virtual machines and can automate the process of

starting and stopping a virtual machine independently. Each network can be described in the MLN configuration language. A graphical representation of a network is also shown in the Fig below. In this network three hosts are connected to a switch, every host in this network is inheriting properties from a super class called host.

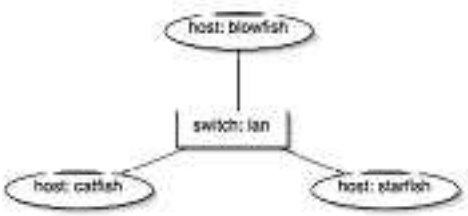

Figure 4.5: A graphical representation of the network described in the MLN language above

Introduction to MLN

MLN generates hosts filesystems through reading the configuration file. Some other scripts like start and stop are also builds by the MLN, MLN has the ability that virtual networks can be build, start, stopped, upgraded and monitored through the MLN commands. As every host's console is accessible from the physical machine, so they can be rebooted individually, this means that, virtual machines which are damaged or some how not working properly, their interface can be accessed by the students without any administrative access to machines. One of the most important feature of the MLN is that, it can support the configuration of a virtual network during run time, it means that more machines can be added to the network, the affected or damaged machines can be rebuild and rebooted without affecting the whole network.

4.5.4 Advantages of MLN

MNL provides a lot of advantages in the filed of teaching networking and system administration concepts, one of the most important is that through the use of MLN virtual networks can be created very fast and easily that can be used in different scenarios of system administrations like, configuration management, intrusion detection and packet filtering. MLN provides an interface through which a whole virtual network can be started and stopped very easily, through the configuration file its possible to make new scenarios for the new class and very easily and fast. As MLN supports to run the system without logging from root, so students are free to create their own test networks. From the security pont of view MLN make sure that a virtual machine

is started only once. Mln2dot is a tool that takes the configuration file and produces a graphical representation of the network topology, so that students can understand the network easily.

4.5.5 MLN Usages

Now we will look in different scenarios, where people are using MLN. MLN can be used in educational institutions to create many Xen or UML instances that students can use for doing their experiments in which they need root level of access to ther network or host and their configuration. MLN can also be used to create testbed systems through which students can test new configurations for system administrator's related issues. It can be used as a part of a production network to separate services into separate instances. MLN was also written with long-time usage in mind. In an educational context through MLN it's possible to administer several virtual networks for a long period of time without much difficulty. Students are also able to build their own networks as a sandbox; through this they can learn many networking and system administration issues by them. Some of the scenarios where people are using MLN for the creation of virtual network are described below.

MLN at Willamette University

At Willamette University, they have a course about basic networking principle. In this introductory networking course, they expect from students to work in groups and create some specific network configuration. The instructors build an MLN simulation for the class assignment. Through MLN this assignment is able show directly, the tasks which are required and also it has flexibility to give students possibility to explore the system in class. The virtual network used at the Willamette University consists of four routers connecting through a ring and each router has a host connect directly to it, which represents the local subnet. A graphical representation of the virtual network is shown in figure below.

MLN at AO Computer Systems

Despite MLN use in educational environment, it can also be used in industries scenarios. At AO Computer Systems, staffs have developed a test simulation in MLN configuration file for the basic level of the system configuration. This test simulation allows them to check any system as experiment very fast and cheaply before putting this into the real production environment. Through this many problems can be resolved before to put the configuration in to production environment on the real machines. This this way MNL has been used for many projects, allowing more quick and secures development reducing the possibilities of any critical system outages and problems. It has been seen that more than 50%of system failures and outages are caused by system administration processes and errors [6]. Through MLN to create a simulation and testing that simulation before putting that system into real environment eliminated the possibilities of several errors.

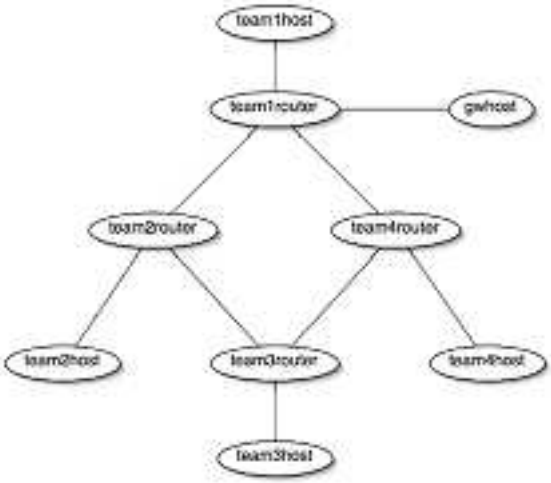

Figure 4.6: Graphical representation of Willamette Universitys virtual network

MLN at Oslo University College

University college of Oslo has a course called Intrusion detection and firewall Security in a master degree program in Networks and System Administration. In fall 2004 college decide to teach this course using virtual machines. Before this college used to teach this course in a regular physical lab, now with virtual machines they developed scenarios in which they are generating about 40 virtual hosts and about 24 switches. MLN allows building this type of scenarios very efficiently and can making changes and generating new test cases very rapidly. For the whole lab college used a single server with two AND Opteron CPUs and 2 GB of RAM. This simulated network gives them very encouraging results. First time when senior students have used MLN in their projects of distributed environment, for testing their problems, they got satisfactory result. Now every year college makes group consisting of two students to use the virtual network environment for getting practical experience in Intrusion detection and firewall Security course and it suites well for the students as well as for the college with respect to the resource limitations.

Chapter 5

Conclusion

Prior to the usage of virtual machines, institutions were forced to fund dedicated computer lab in order to teach advanced classes covering topics such as Networking design, system administration, information security, and operating system. However such labs are often expensive and not feasible for most institutions. The emergence of VM software enables educational institutions to configure shared, multi-course computing labs with cost effective.

VM features such as isolation, compatibility and encapsulation allows instructors to build virtual network topologies, which can consist of multiple independent operating systems and networks. These features allow instructors to create a virtual kernel development environment, where operating systems can be developed, debugged and rebooted in shared computing lab environment without affecting other applications of users. This type of application development and debugging support is particularly important in the field of Networking and Operating Systems and highly valuable for emphasizing troubleshooting in teaching the concepts of System Administration.

We discussed five case studies and explore how people are using virtual machine concept in different areas of education and getting benefits from them. Virtualization technologies provide several important features that make it a very powerful tool to be used across a wide range of applications in education. Virtualization environment provides excellent platform for providing practical experience to the students in the filed of network and system administration, operating system, computer system architecture, IT Security, development of system software and many others. Students can install, configure and experiment with services related to areas mentioned above.

All these features have made these technologies immensely popular in Institutions as well as in industries. As we discuss in case studies different people used different technologies according to their needs, goals and available resources. A closer look reveals that although most of the technologies used present a similar operating environment to the end-user, they greatly vary in their architecture design, implementation, and the level of abstraction at which they operate at.

We did survey of some examples of these categories of virtual machines and study the general design, implementation details, challenging issues involved, and benefits over the physical machines.

Several benefits of Virtual Machine usage accured with in each course delivered through Virtual Environment and for the lab as a whole. First, students were able to

exercise administrative privileges with Virtual Machines while logged into the host operating system as ordinary users. Lab machines were never compromised and always available for the next task. second, the ability to restore a previous configuration, like in VMware through revert commond or by copping a previous saved image make possible for the students to quickly recover from the failures. Third, ability to simultaneously execute multiple Virtual Machines enabled complex exercises, demonstration and side by side comparisons. Finally, the one of the most important benefit of Virtual Machine usage from Instructors point of view is that,now they can increase course contents and use a greater number of more complex exercises, thus providing a richer students learning experience.

Virtualization is an area that has been in existence since 1960s and attracts a heavy attention from the research community even today. So, we believe, such a survey could actually help people analyze new developments in the area and help put them in perspective.

Bibliography

[1] Virtual Machines : James. SMITH . Ravi NAIR: *"Versatile Platforms for Sysytems and Processe"*

[2] Adam Vollrath and Steven Jenkins, *"Using Virtual machines for Teaching System Administration"*.

[3] Ji Hu, Dirk Cordel, Christoph Meinel, *"A Virtual Laboratory for IT Security Education"*, FB IV Informatik Universitaet Trier D-54286 Trier, Germany.

[4] Ernesto Damian, Fulvio Frati and Davide Rebeccani, *" The Open Source Virtual Lab"* A Case Study, Department of Information Technology University of Milan Italy.

[5] Teaching Computer Concepts Using Virtual Machines, *"Department of Computing Science Umegt University SE-901 87 Ume :t, SWEDEN"*.

[6] Kyrre Begnum, Karst Koymans, Arjen Krap, John Sechrest, *"Using Virtual Machines in System Administration Education"*.

[7] VMware, Vmware workstation, http://www.vmware.com/products/desktop/ws features.html.

[8] Microsoft, Microsoft virtual PC 2007
http://www.microsoft.com/windows/products/winfamily/virtualpc/default.mspx.

[9] A. Whitaker, M. Shaw, and S. D. Gribble, *"Denali: Lightweight virtual machines for distributed and networked applications"*.

[10] Plex86,*"Plex86 x86 virtual machine,"*
http://savannah.nongnu.org/projects/plex86.

[11] http://www.usenixorg/publications/library/proceedings/als00/2000papers /papers/full_papers/dike/dike_html/index.html

[12] Jason Nieh, Chris Vaill, *"Experiences Teaching Operating Systems Using Virtual Platforms and Linux "*.
partment of Computer Science Columbia University

[13] Ronald D. Williams, Senior Member, IEEE, Robert H. Klenke, Senior Member, IEEE, and James H. Aylor, Fellow, IEEE, *"Teaching Computer Design Using Virtual Prototyping"*.

[14] Harry Bulbrook, *"Using Virtual Machines to provide a secure Teaching Lab environment "*.
Durham Technical Community College 1637 East Lawson

[15] paul barham,Boris Dragovic,Keir Fraser,Steven Hand,Tim Harris, Alex Ho,Rolf Neugebauer,lan Pratt,Andrew Warfield, *"Xen and the Art of Virtualization"*
Universiy of Cambridge Computer Laboratory

[16] Steve Liu, Willis Marti, Wei Zhao, *"Virtual Networking Lab (VNL): its concepts and implementation"*.
Computer Science Department Texas A and M University College Station

[17] An extensible virtual machine architecture, *"Virtual Networking Lab (VNL): its concepts and implementation"*.
Computer Laboratory,Pembroke Street,Cambridge,UK,October 3, 1999

[18] Joseph A. Driscoll, Ralph M. Butler, Joelle M. Key, *"A Virtual Machine Environment for Teaching the Development of System Software "*.
Department of Computer Science Middle Tennessee State University Murfreesboro

[19] Bruce Kneale, Ain Y. De Horta, Ilona Box, *"VELNET (Virtual Environment for Learning Networking) "*.
School of Computing and Information Technology University of Western Sydney, AUSTRALIA