



Presenter Name : Asish Chattopadhyay
Application Architect, IBM India Private Limited,
Date : 22 & 23 July 2013



Introduction to Virtualization Technique



Agenda

- Introduction to virtualization Technique Instructor's Slides

Introduction to Virtualization

▪ Overview

To build an abstraction of the existing system, a generic logic definition of the system is built into software. With a generic definition in place, it is possible to replicate the behavior of the cloned system over multiple instances. These multiple instances can then be made to run on the same hardware system it clones, and can share resources through a hypervisor that multiplexes the requests to the single system real hardware instance. Additional features provided by virtualization allow for emulation of totally unrelated or legacy hardware on the common off the shelf hardware systems.

Virtualization eliminates most of the inflexibilities inherent in the hardware systems and allows for better manageability leading to a better utilization of the system.

Traditional IT Infrastructures

Traditionally, organizations have relied on physical infrastructures to manage data and information. With the explosion of information and related data, the physical footprint required to manage the information base has grown tremendously.

Introduction to Virtualization

The common shortcomings of physical infrastructures are listed below :

- **Asset Management:** It is becoming increasingly difficult to keep track of physical server assets. The rapid growth in information base and the need to store tremendous amount of data led to an adhoc increase in servers, datacenters, storage dispersed in diverse geographical locations. A typical organization runs at-least 2-3 datacenters to meet their IT computing and availability needs. The heterogeneity of the infrastructure also makes it difficult to run any common manageability applications to track the physical assets.
- **Tracking Utilization:** Utilization tracking has become a major concern in recent years. With the increase in physical sprawl, the unmanageability of physical systems made it difficult to track the utilization on each system. As a result, a datacenter either ended up with systems that are simply powered on but put to no good use, or systems that are running at maximum load and low performance, in spite of available free capacity in the datacenter.
- **Security and Compliance:** With no efficient datacenter management and monitoring in place, security lapses and complex compliance requirements have resulted in concerns over the cost involved to ensure security/compliance in the infrastructure.

Introduction to Virtualization

- **Provisioning:** The provisioning in traditional IT datacenter takes on an average 6-8 weeks given the time and complexity involved in creating rack-space, setting up the physical server and storage, installation of OS and patches, setting up the applications and allied components. You also need to factor in additional time to test the setup after configuration. A large part of testing is manual to test things that were configured manually. There's a good scope for errors to creep-in at various stages of configuration and testing.
- **Staff for Administration:** There has been an exponential increase in staff requirements for managing the ever-growing and complex physical infrastructure.
- **Sizing:** For any new requirement, the physical infrastructure has to be resized well in advance. In future, if the requirement goes down, the surplus infrastructure remains largely unutilized for a long time, before it is marked for another purpose. This is largely due to the inflexibility inherent in the physical infrastructure.
- **Optimization:** Due to the lack of any granular mechanism to monitor the complex infrastructure for utilization and performance, it is difficult to optimize on an ongoing basis. The effort involved in optimization is generally too high to justify any cost-savings.

Introduction to Virtualization

Benefits of Virtualization

Traditionally, applications have been run on physical systems with limited amount of CPU, memory and IO resources. This has been fine until the system capacity closely matched the requirement of the application. The end result was close to optimum utilization of the infrastructure. Also, the application were not very complex, hence advanced manageability was not required. A dedicated group of system administrators were enough to handle the infrastructure.

With the rapid adoption of IT in business, the complexity of the system has gone up. Also, with the advancement of technology, the resources available on an average hardware system surpass the requirements of an application. This has resulted in systems that are not effectively utilized to their maximum capacity. This leaves lot of capacity underutilized on these systems. These systems still consume power and cooling and take up space in the data center. Also, since a single hardware system only offers limited isolation between applications, this poses a limit on the number of application you may want to put on a single hardware system. Buying an additional system to host an application for security reasons have resulted in further sprawl of physical infrastructure. With the increasing physical systems footprint, the infrastructure becomes too complex to manage since each physical system would require installation, regular updates to the system and application software and this has to be repeated over the entire infrastructure. So, the turnaround times to service these requests began to run into weeks and months. Virtualization aims to mitigate most of these issues by building a more flexible and manageable IT infrastructure.

Introduction to Virtualization

The benefits of virtualization can be classified in three categories :

- **Utilization:** Virtualization builds an abstraction of hardware system resources in software. Each OS/Application can be run in its own isolated environment called the virtual machine. This allows sharing of system resources at a much granular level and brings up the utilization of the physical system. Also, consolidating applications running on separate systems onto one system would lead to saving in datacenter space, power and cooling.
- **Security:** Since, each application runs in its own virtual machine, there is a strict isolation of system resources and sharing only happens in a controlled manner, thus allowing for higher security in the system
- **Manageability:** It is possible to move the virtual machine and abstracted system components around in the infrastructure based on load distribution and save a snapshot of virtual machine to be restored later. Patched VMs images can be maintained in a VM library and can be cloned and activated on demand. This saves a lot of time spent in system administration.

Introduction to Virtualization

A tabular comparison of traditional IT infrastructures with virtualized infrastructures is as follows:

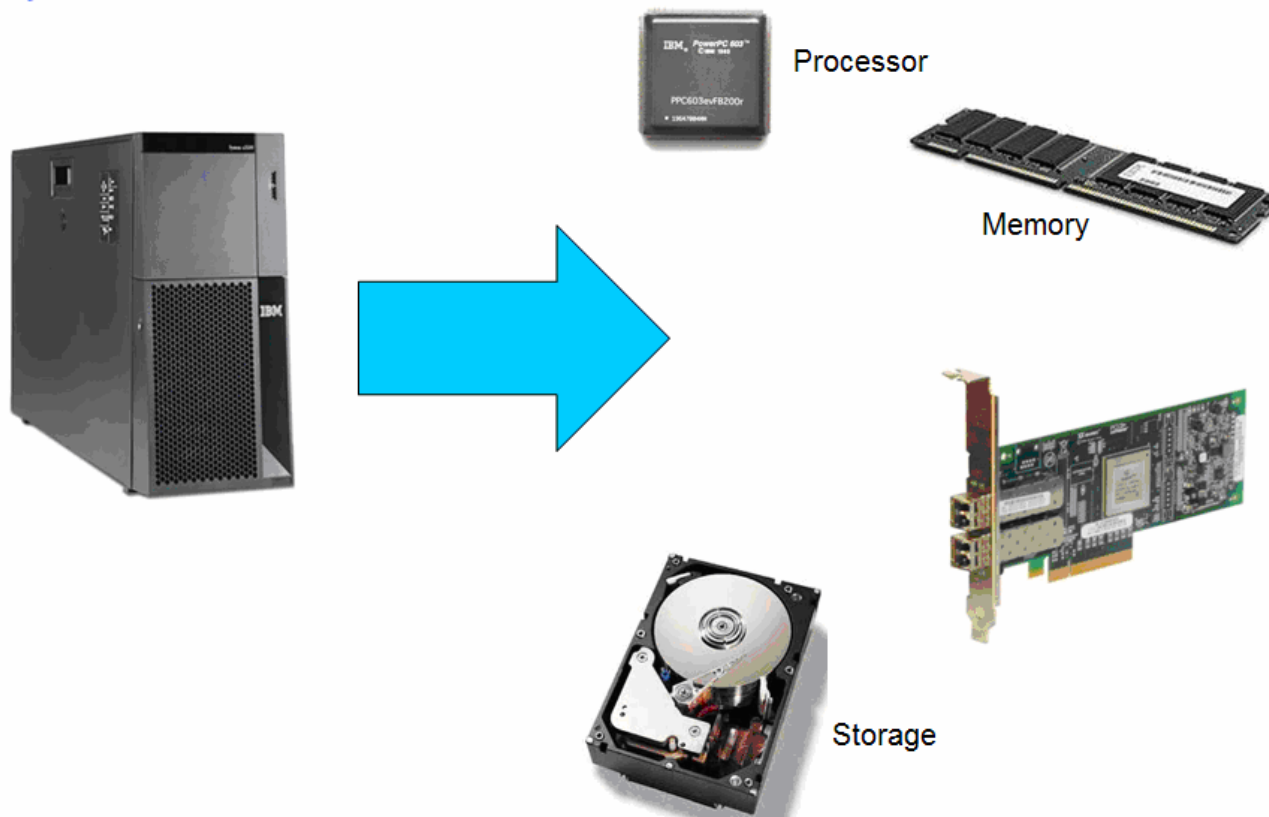
Parameter	Traditional IT	Virtualization
Utilization	0-20%	Typically 60-70%
Provisioning	Typically takes 6-8 weeks	1 day
Monitoring	Usage of monitoring tools. However, need manual intervention to take care of any hardware failures	Comparative ease in monitoring using automated tools. However, need manual intervention to take care of any failures
Sizing	Sizing needs to be completed before deployment. Re-sizing involves procuring new hardware and planned downtimes	Easier to resize. However, manual intervention required to resize
Staff for Administration	Require larger number of Full Time employees to manage the infrastructure	Reduced number of Full Time employees
Cost	Upfront costs involved in outright purchase of hardware	Initial hardware cost reduced due to sharing of hardware assets and increased utilization. There is a typical reduction of 40% in hardware
Optimization	Difficult to do as there is no easy way to monitor and load balance across machines	Easy to share resources and re-balance loads on the virtual machines on the same host. However, re-balancing across physical hosts require advanced features and planned downtime

Introduction to Virtualization

Implementing Virtualization

A physical server is typically composed of four major physical components – Processor, memory, network and storage (adapters and disk drives).

Physical Server

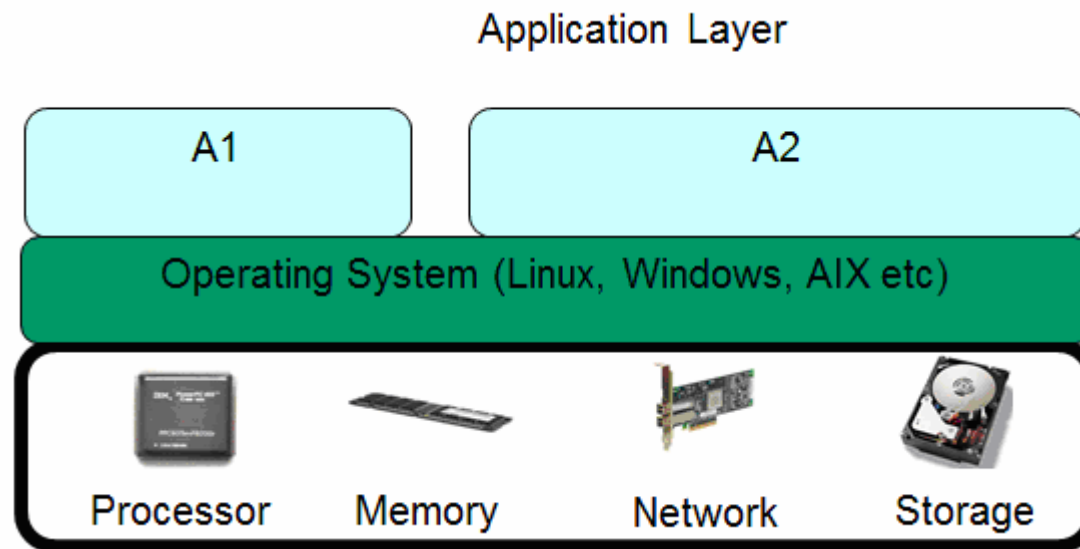


Dedicated Hardware Components

Introduction to Virtualization

To enable an application to perform a useful function, the four hardware components are required – CPU, Memory, Network and Storage. In addition, the application requires support from the Operating System and allied components.

A typical hardware/software server stack is show below. The components marked in black box are the hardware components. The Operating System resides over the hardware. The applications are stacked over the operating system and use services provided by the OS.



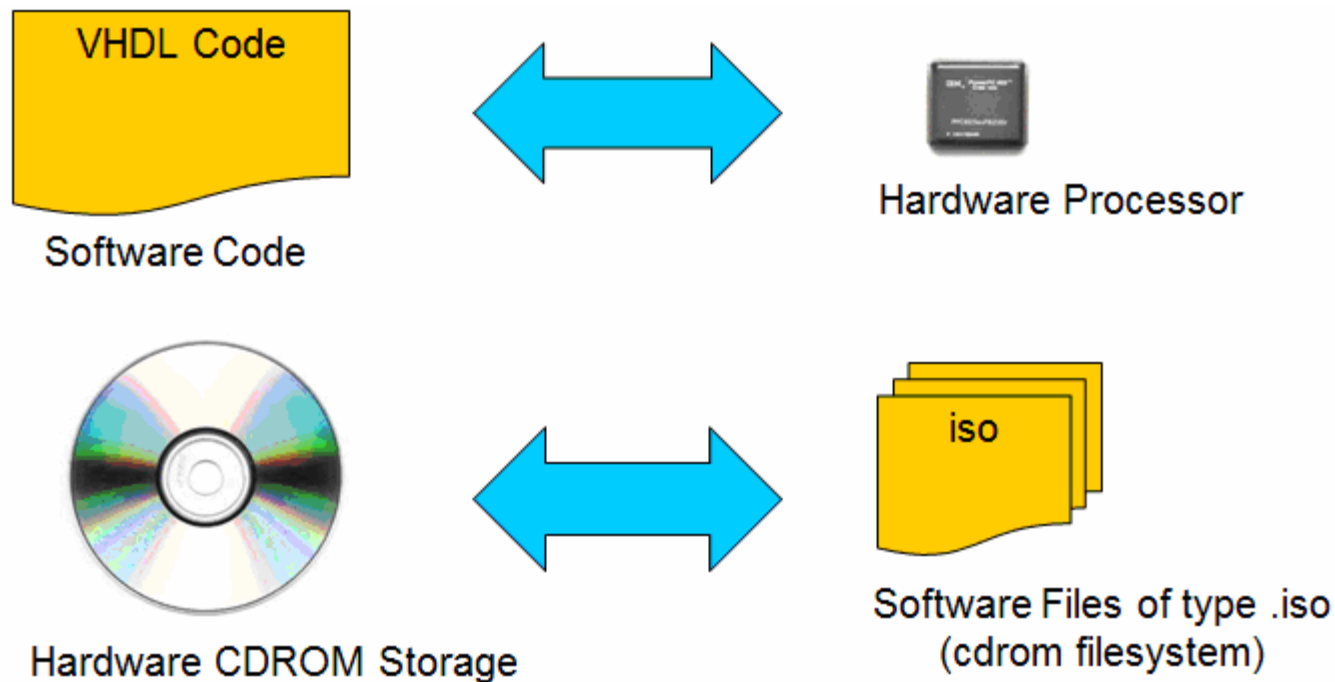
A typical server stack

Introduction to Virtualization

The principle of hardware-software logical equivalence works as follows:

Any logic written in software can be easily converted to a hardware equivalent. Any existing hardware can be easily converted to software.

The examples shown below illustrate the logical equivalence:



Introduction to Virtualization

The VHDL code (software) is logically equivalent to a hardware processor. The VHDL code can be synthesized and fabricated into a physical processor. On the other hand, given a physical processor, it is possible to represent the logic function in an equivalent VHDL code.

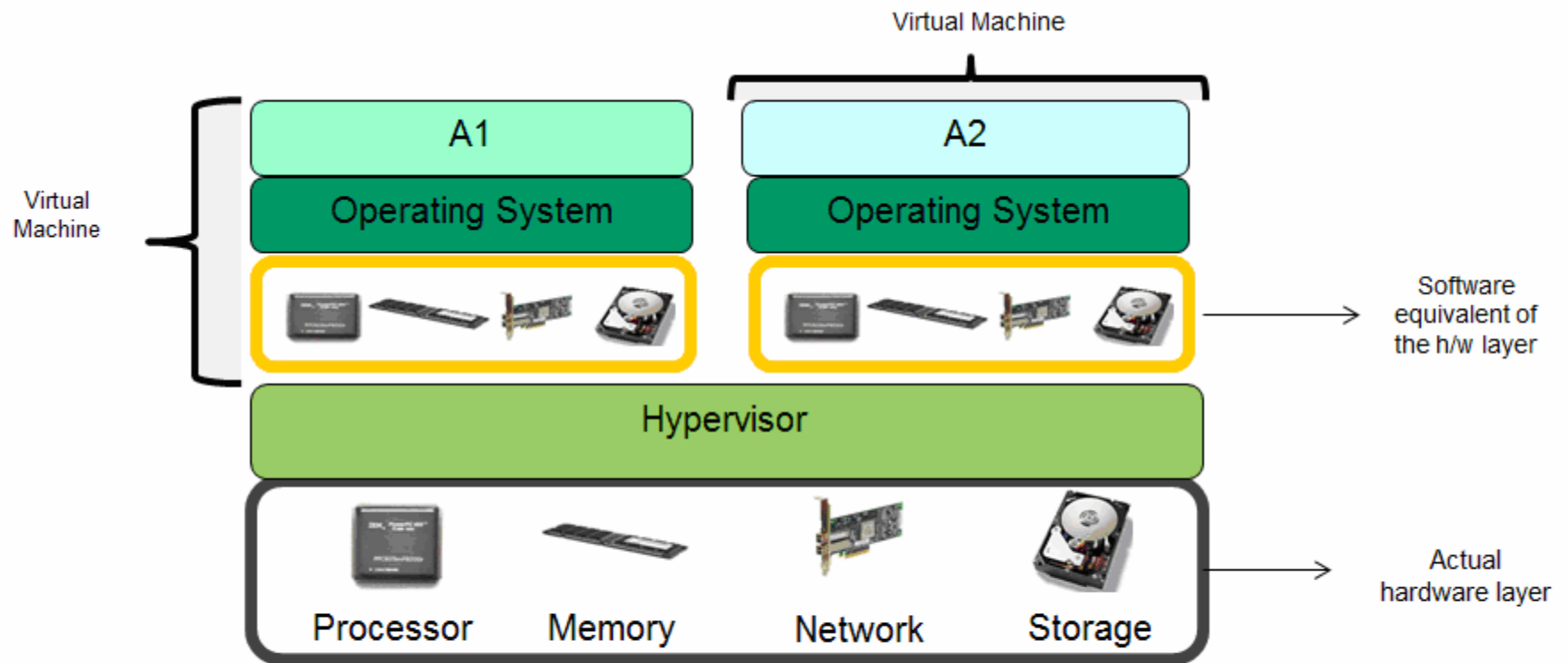
Similarly, in the second example, we see a CDROM storage that can be converted into an iso9660 software image. It is also possible to convert/burn the image back to a CDROM. Logically these two are equivalent.

The hardware-software logical conversion is generally a tradeoff between performance and flexibility.

VHDL code is easier to modify and simulate than a physical processor. A physical processor once fabricated cannot be changed. However, the performance of simulation using VHDL is much slower than a physical processor.

Based on the principle of hardware-software equivalence, it is possible to logically convert the system hardware components required to run an OS/application, to their software equivalent or a virtual machine (orange box in the diagram). The virtual machine can then be replicated to create multiple virtual machines each running its own instance of the OS and the application. A hypervisor layer (marked in light-green) is required to multiplex the actual real hardware resources among the virtual machines. The hypervisor is responsible for allocating memory, CPU resources, network and storage to each virtual machine.

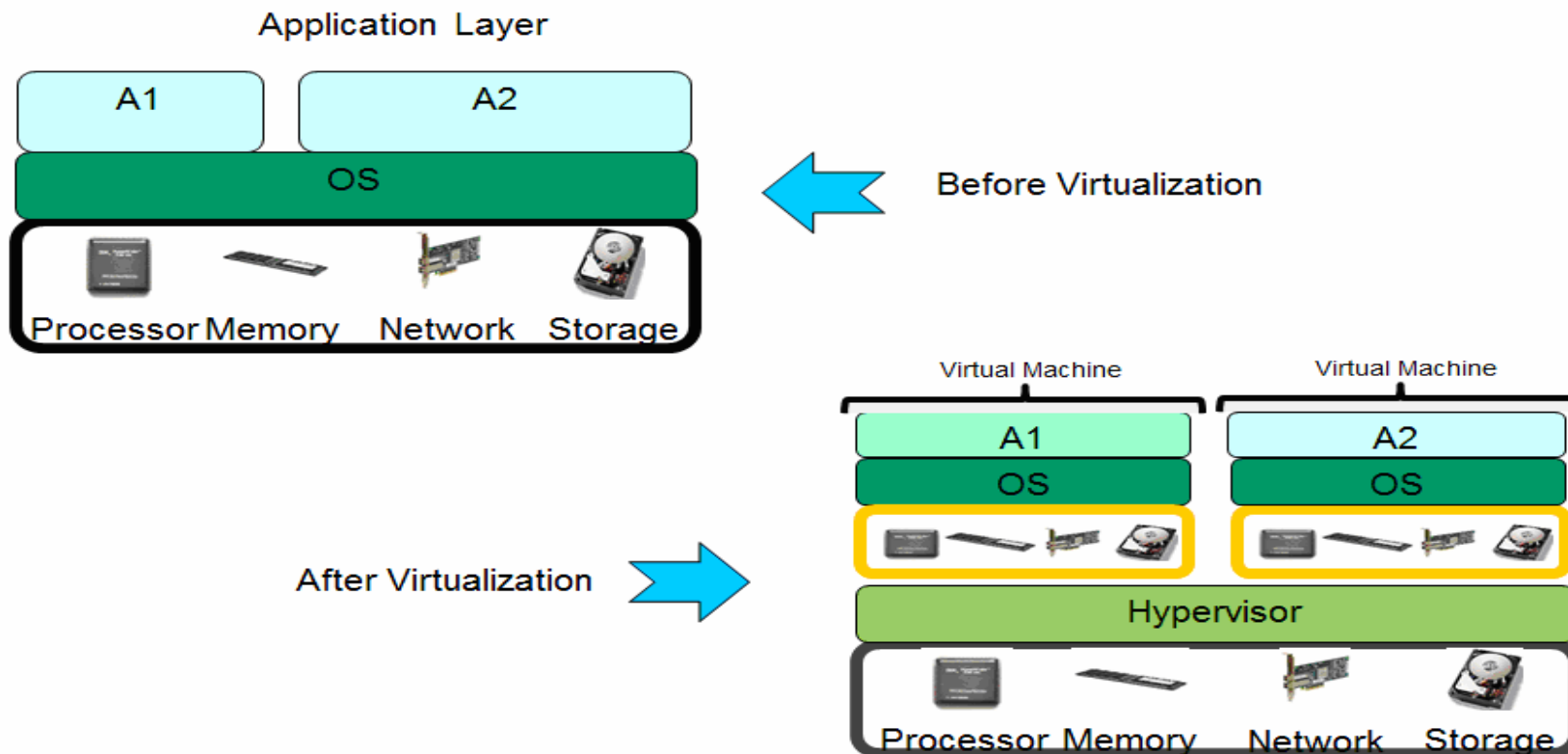
Introduction to Virtualization



Introduction to Virtualization

The figure below shows the pre- and post- virtualization server stacks.

The boxes in orange are an exact replica of the hardware underneath. The figure shows two instances of the abstraction. Each abstraction is known as a virtual machine. Each virtual machine is capable of running its own OS and applications, very similar to the original hardware. The virtual machine in most cases, imitate the underlying hardware, however it is also possible to emulate a different hardware altogether.



Introduction to Virtualization

Types of Virtualization

Virtualization is classified based on the extent of hardware emulation.

- **Full Emulation:** The virtual machine emulates complete set of hardware and peripheral components. This allows for an emulation of a completely different type of hardware distinct from the real hardware. For example: A PowerPC architecture can be emulated over x86 hardware. The guest OS designed for PowerPC, in this case can be run unmodified on the emulated hardware. A few examples of emulated hardware are Qemu-PPC Emulator, Alpha ES40 emulator, Qemu-ARM emulator etc.
- **Full/Native Virtualization:** It is possible to emulate only a section of hardware and use the real-hardware for the rest. For example: The x86 Xen Full-Virtualized machine would generally emulate the hard drive and use the real x86 cpu with no emulation. This offers higher performance as you bypass the emulation layer and use/multiplex the real hardware (with aid from the hypervisor) wherever possible. This also places a constraint on what kind of virtual machine can be defined. For example: The full-virtualized virtual machine can only be a x86 instruction compatible when running on a x86 hardware. For example Xen-Full Virtualization, Qemu-x86
- **Para-Virtualization:** There is no hardware simulation/emulation done by the Virtual machine. The virtual machine uses hypercall API to communicate with the hypervisor for instruction dispatch and other purposes. The guest OS must be modified to work with the hypercall API. Only opensource OS/Certified OS that can be modified are suited for para-virtualization. This method places additional constraints on the OS. For example : Xen-ParaVirtualization
- **OS Virtualization:** This is a technique in which a single OS instance allows multiple sub-instances of OS to be run in a secure environment. The goal is to multiplex OS resources among different instances. For example: BSD Jails.

Introduction to Virtualization

- **Application Virtualization:** Application as a virtue of their design and implementation allows virtualization by creating a separate sub-application instance with components that are not shared within the application (files, memory objects). For example: Java Virtual Machine.

Virtualization can be also be classified based on the technology or the area that is being virtualized – Server, Storage, Network

- **Server Virtualization:** Refers to virtualization of the server hardware components so that each virtual machine can run its own logical server instance. A virtual machine provides a complete operating environment regardless of others. All server components for the virtual machine are virtualized.
- **Storage Virtualization:** A technique which creates a logical abstraction of the underlying storage hardware/firmware components and provides a uniform view of the heterogeneous storage hardware used within a storage infrastructure. This allows for enough flexibility in managing the storage hardware
- **Network Virtualization:** A technique which creates a logical abstraction of the underlying network hardware and resources and provides a uniform view of the heterogeneous network hardware used within the infrastructure. This allows for enough visibility and flexibility in managing the network hardware.

History of Virtualization

Introduction

Virtualization started out as a concept of time-sharing in the 1960s. The machines were large and there was need to split up the machines into multiple portions that can be used for different purposes simultaneously. This resulted in evolution of newer innovations related to hardware sharing, paging techniques and multiprogramming.

Time Sharing Systems

Earlier systems in the 1960s allowed only one user/job at a time on the system. This came as a major disadvantage to larger computers that were technically powerful enough to execute multiple jobs at a time, but not capable due to hardware/software constraints of the system. Simultaneous time-sharing of the system was seen as a solution to this. However, modifying legacy batch systems to accommodate multiple simultaneous users on the system made the operating system too complex to maintain and support.

The first supercomputer to take advantage of the concepts of shared physical hardware was the Atlas Computer. The computer was developed at Manchester University. The computer allowed for separation of the supervisory (OS) components from the user components. The supervisory code monitored and managed system resources (CPU, memory and IO). The supervisory component responded to special instructions that enabled it to provision and monitor the user computing environment.

History of Virtualization

A very rudimentary virtual memory system was also introduced by Atlas in the form of one-level store and paging techniques. This created an abstraction of how memory gets used by the user programs and made managing memory less dependent on hardware.

IBM at the time was looking for a time-sharing implementation on their Mainframe systems. The IBM Engineering Team at Cambridge, Massachusetts came up with a novel approach to solve the problem. They provided each user with a virtual machine (VM) with an operating system. The operating system running in the VM still supported one user, so it does not have to be complex. In fact the existing operating system could be used without any changes. The real ingenuity was in how the VMs on the Mainframes multiplexed these hardware resources among themselves.

History of Virtualization

IBM Mainframe Virtualization

The earliest pioneer of modern virtualization technology was IBM. IBM invented virtualization more than 40 years ago.

IBM started with virtualization in the 1960s with the M44/44X project. This was developed at the IBM Thomas J. Watson Research Center in Yorktown, NY. The foundation of this technology was an IBM 7044 (M44) scientific computer. A large number of virtual machines of the type IBM 7044 or extensions (44X) could be run on this hardware. The computer used advanced virtual memory and multiprogramming technologies.

IBM extended their virtualization capability by introducing a successor to M44 (7044) series by introducing IBM 7094 series computer. The computer basically ran the FMS-Fortran Monitor System in the main 7094 and a copy on the virtual machine in a background facility.

In the late 1960s, IBM introduced the first successful virtual machine operating system, the CP-40 which was geared for the System/360 Mainframe. A revision of CP-40 was introduced by the name of CP-67 and was later implemented as S/360-67 and finally as S/370.



History of Virtualization

The operating system that ran on these machines was named as VM/370. The new operating system had a special component called the VMM (Virtual Machine Monitor) which was capable of running many virtual machines, with larger virtual memory running on virtual copies of the hardware. Each virtual machine ran an unmodified copy of the operating system with good performance. The virtual machine was capable of running UNIX and other operating systems.

One of the unique features of the IBM virtualization technology was that the virtualization is part of the system's firmware. The hypervisor sits in the firmware layer and provides an integration between the hardware and the VMs running an OS.

IBM PowerVM Virtualization

In addition to Mainframes, IBM currently provides virtualization on Power Servers – the midrange UNIX systems. The Power Servers are capable of advanced virtualization mechanisms some of the notable features are : micro-partitioning, Advanced memory sharing, Live partition mobility, Virtual IO Server for IO Virtualization.

The first Power Server incorporating Advanced Power Virtualization (APV) was shipped in 2004. APV was rebranded to IBM PowerVM in 2008.

History of Virtualization

Extending Virtualization to x86

- The virtualization on x86 Architecture was introduced in the year 1985.
- In 1985, the Locus Computing Corporation, in cooperation with AT&T introduced Simultask; a virtual machine monitor for running Intel 80286 based guest operating systems, under a UNIX System V host. The system later renamed as Merge, and developed as Merge/386 made use of the Virtual 8086 mode provided by 80386 architecture. It was possible to run multiple simultaneous virtual 8086 machines on a single hardware machine.
- In 1997, Connectix released the first version of “Virtual PC” for MAC platforms
- In 1998, VMware filed an US Patent 6,397,242 for virtualization techniques for x86 architecture and subsequently, a VMware Virtual Platform was introduced for IA32 architecture.
- In 2000, FreeBSD introduced FreeBSD Jails for OS Virtualization
- In 2001, Connectix building on its initial success on Virtual PC launches the first version for Microsoft Windows.
- Year 2003, marked the release of the first open source hypervisor for x86 machines called Xen Hypervisor. The company XenSource that developed the hypervisor was later acquired by Citrix. Citrix is currently one of the major virtualization solution providers in the x86 market.

History of Virtualization

- In 2006/2007, Virtual Iron released Virtual-Iron, an x86 bare-metal hypervisor for enterprise customers. VirtualBox was also introduced as an opensource alternative under the GPL license.
- In the recent years, a new technique to provide virtualization on Linux/x86 has been developed. This project is called the “KVM” Project. The source code for kvm is part of the main linux kernel tree. KVM relies on the support from x86 hardware (including the Intel VT or AMD-V support) for better performance. In absence of hardware support, Qemu is used in emulating the required hardware components.
- KVM is now part of the standard linux kernel distributed along with a standard Linux distribution – Redhat, Novell, Ubuntu, Debian etc.

History of Virtualization

Hardware Support for x86 virtualization

- Early x86 processors had no built-in support for Virtualization. The virtualization was achieved using a software-only hypervisor using complex techniques to multiplex resources among virtual machines. The performance of these systems was reasonable but not suitable for putting in production. The virtual machines were primarily used in the test and development teams or in places where performance was not a qualifying criteria.
- Hardware Assisted Virtualization began to take shape from 2005, which implemented some of the commonly used functions into x86 hardware.
- Intel introduced Intel VT-x and AMD introduced AMD-V to support virtualization in hardware.
- Pentium 4 (Model 662 and 672) were the first Intel processors to support VT-x.
- AMD Athlon processors starting from Athlon 64 ("[Orleans](#)") supported AMD-V, the technology to support Virtualization in hardware.
- As time progressed, in addition to hardware-assist for CPU virtualization, advanced hardware techniques to virtualize memory, and IO were introduced through different chipsets. A basic principle behind these techniques was to provide a shadow copy of the hardware to each virtual machine and to have a bypass implemented in the hypervisor to access these copies directly.

History of Virtualization

Impact of Virtualization

Virtualization plays a key role in industry today. With exponential increase in Information technology, the need for more powerful and larger datacenters continues to grow.

Heavy reliance on physical infrastructures in the initial years of information explosion has resulted in unmanageable, power and space hungry data centers. A large part of IT effort is spent on managing the ever growing physical footprint.

This complexity inherent in today's data centers leaves a lot of scope for errors, security risks and disasters that may lead to loss of valuable and critical data.

Keeping all this in mind, an effort has been made to make these infrastructure assets more traceable, manageable and amenable to change. Also, the associated cost to manage an IT infrastructure has to go down to sustain the growth of IT.

History of Virtualization

The impact of virtualization can be classified under two major headings: cost and manageability.

Cost Impact

Traditional physical infrastructures consume lot of power to keep IT running. However, a large percentage of the infrastructure remains unutilized for a myriad of reasons

- **Security/Segregation of critical applications:** The applications are segregated on different physical servers for security/criticality reasons. If only these applications can be closely packed on a single server we can bring up the utilization levels on that physical server. In most cases, a physical server runs with a utilization of no more than 20%, thus about 80% of capacity on the server remains unutilized.
- **Limited or No Monitoring:** Traditional physical systems are not monitored for performance or utilization on an ongoing basis. Monitoring/Audit of physical systems is generally done only on a periodic basis.
- **Ineffective Asset Management:** Lack of automated tools to monitor removal/addition of physical components in the infrastructure. Even addition/removal of servers sometimes goes unrecorded when done on an adhoc basis. The asset list frequently goes out of date to make any conclusive assessment on the available capacity at a later date.
- **Provisioning Turnaround:** The time required to provision a server may range from a day to a month based on the request. The provisioning may require creating rack-space for new machines, provisioning for networking and storage, installation of OS and complex patch updates which take time ranging in days to months. In the process, valuable machine time gets wasted, that could have been used for running an application.

History of Virtualization

All these reasons contribute to ineffective utilization of the physical servers thus resulting in higher cost to maintain and run a datacenter. Virtualization on the other hand eliminates most of the drawbacks of the traditional IT infrastructure by building flexibility into the infrastructure.

Since each application is running into its own virtual machine on a physical server, it becomes possible to closely pack virtual machines thus bringing up the utilization.

Monitoring is relatively easier in virtualized environments since monitoring could be done from the hypervisor with inbuilt statistics.

Typically, 60-70% of the infrastructure can be virtualized thus resulting in reduction of about 60-70% of physical servers. This enables better asset management and monitoring.

Provisioning with virtualization is faster as it is now possible to capture and store existing configuration of the virtual machine in the form of images and image libraries. These images can then be restored back to a fully functional virtual machine on demand, without repeating the OS and application configuration.

A virtualized environment can also be maintained by reduced staff due to the ease of managing the reduced physical footprint.

Virtualization saves cost on three fronts: People, Process and Technology.

History of Virtualization

Manageability Impact

A virtualized infrastructure brings more visibility into the working of an IT datacenter. It is possible to track servers on an utilization basis, move workload around to load balance machines in real-time, save on power and cooling as we have seen under the cost impact.

An important aspect of virtualization is the visibility it brings to the IT infrastructure by enabling better manageability in the system.

An improved visibility helps to plan the infrastructure better, avert any disasters, optimize and fine-tune backups, better accounting to plan for new hardware requirements and overall growth.

Virtualization is a vast field. It is now possible to virtualize all aspects of an IT infrastructure ranging from Desktops to Enterprise Servers. The decision to virtualize is solely based on the customer requirements.

History of Virtualization

Some of the areas where virtualization is not recommended:

- Legacy software/systems that are not designed for virtualization
- Resource Intensive Applications that make assumptions on specific system characteristics to operate and perform (Proprietary Applications)
- Real-Time Applications where the timings is critical. We place these applications as close to the hardware as possible to guarantee turnaround times
- Other large applications such as databases

Server Virtualization

Overview

Types of Server Virtualization

Server Virtualization can be classified based on the extent of emulation required and the interface between the hypervisor and virtual machines.

Full Virtualization

Full Virtualization refers to a set of techniques to fully simulate the underlying physical server environment. The simulation is so effective that an operating system designed to run on a physical server can be run unmodified on a fully virtualized virtual machine.

A fully virtualized virtual machine (FVVM) is completely isolated from other virtual machines on the same physical host and is unaware of their existence. Any operation executed on a virtual machine (triggered by VM-resident OS or applications) does not have any external impact on other FVVM running on the same physical server.

The hypervisor traps and effectively translates any machine instructions for IO operations or any other instructions that require direct or privileged access to the underlying physical hardware. The requests from one VM are not allowed to alter the state of another VM.

Server Virtualization

Emulation

Emulation refers to a set of techniques to virtualize a completely distinct hardware than the underlying physical hardware. Emulation could be achieved in hardware or software. Emulation refers to the use of firmware and hardware to emulate an altogether different hardware. However, in the context of virtualization, the term became popular to describe the emulation of a different architecture (in software code) on a totally different physical hardware.

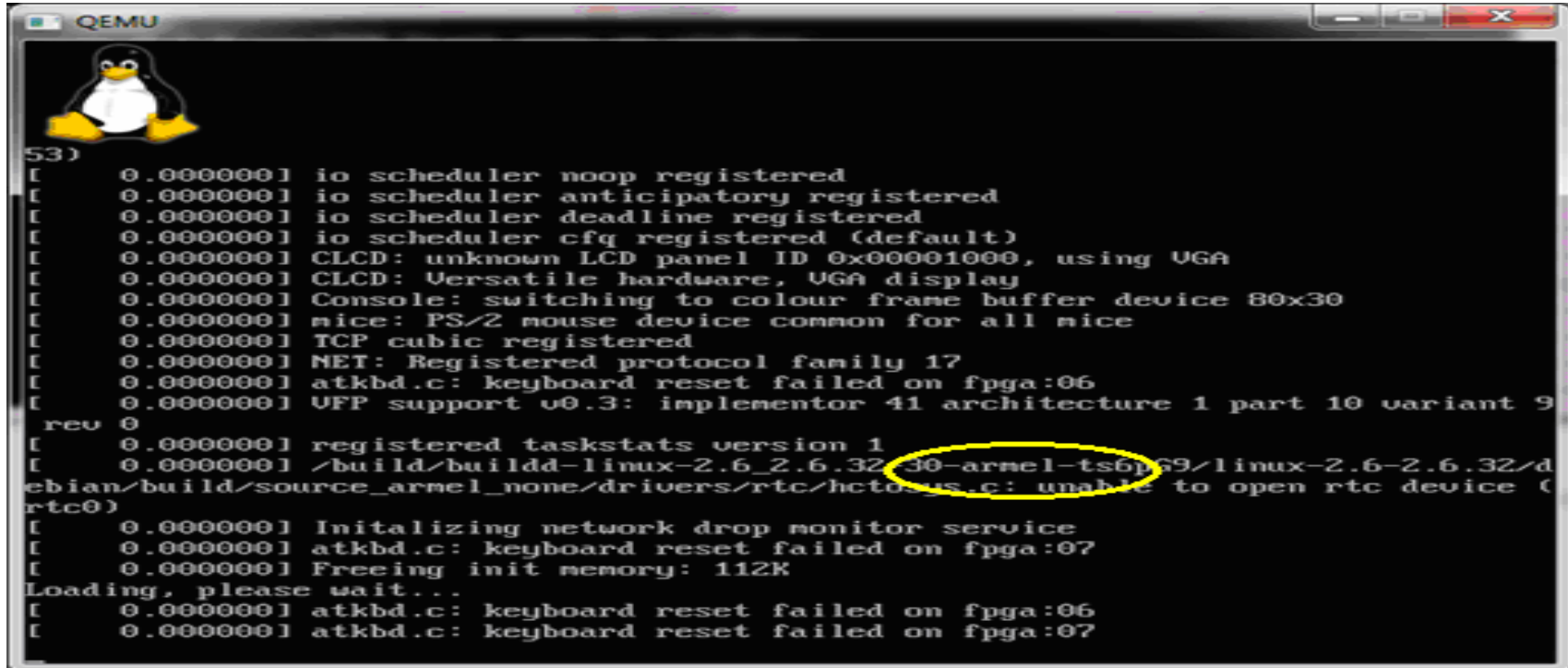
Emulation is only possible through binary instruction translation. The machine instructions of the emulated processor are translated to native machine instructions before dispatching to physical hardware.

For example:

Qemu-PPC can emulate the PowerPC architecture environment on an x86 physical machine.

Qemu-ARM can emulate the ARM architecture environment on an x86 physical machine. Shown below is a screen capture of Linux booting up on a Qemu emulated ARM architecture machine.

Server Virtualization



```
53)
[ 0.000000] io scheduler noop registered
[ 0.000000] io scheduler anticipatory registered
[ 0.000000] io scheduler deadline registered
[ 0.000000] io scheduler cfq registered (default)
[ 0.000000] CLCD: unknown LCD panel ID 0x00001000, using UGA
[ 0.000000] CLCD: Versatile hardware, UGA display
[ 0.000000] Console: switching to colour frame buffer device 80x30
[ 0.000000] mice: PS/2 mouse device common for all mice
[ 0.000000] TCP cubic registered
[ 0.000000] NET: Registered protocol family 17
[ 0.000000] atkbd.c: keyboard reset failed on fpga:06
[ 0.000000] VFP support v0.3: implementor 41 architecture 1 part 10 variant 9
rev 0
[ 0.000000] registered taskstats version 1
[ 0.000000] /build/builddd-linux-2.6_2.6.32-30-arnel-ts6p69/linux-2.6-2.6.32/debian/build/source_arnel_none/drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
[ 0.000000] Initializing network drop monitor service
[ 0.000000] atkbd.c: keyboard reset failed on fpga:07
[ 0.000000] Freeing init memory: 112K
Loading, please wait...
[ 0.000000] atkbd.c: keyboard reset failed on fpga:06
[ 0.000000] atkbd.c: keyboard reset failed on fpga:07
```

8051 microcontroller emulators are popularly used in embedded system development to emulate 8051 hardware on a x86 physical machine.

Rosetta is Apple's emulator for PowerPC architecture. The emulator ran on Apple Intel based MAC.

OVPsim Emulator can emulate varied processor architectures including MIPS architecture on a physical x86 server.

Server Virtualization

Due to additional steps involved in binary instruction translation, there is generally a performance hit involved. However, in most cases, the hardware that is being emulated is a legacy processor or an embedded processor. These processors, in their physical form run at very low clock frequencies that are only a fraction of that of the host processor. Hence, if an old hardware/embedded system is being emulated on a faster host, the performance is typically more than the emulated hardware. On the other hand, if the emulated hardware compares to the host hardware in terms of original performance, the emulation performance would be at best reasonable or suffer from bottlenecks for the purpose involved.

Emulation is used for the purpose of running old legacy applications in their native OS environments while making use of the cutting edge high performance hardware available today. This eliminates the effort to modify or port these applications to current OS/Hardware technologies. Another reason to use emulation is to test/pre-debug embedded systems or new hardware before going in for actual production

Simulation

Simulation refers to techniques that create an exact virtual footprint of the underlying hardware for the purposes of virtualization.

The main reason for virtualization-by-simulation is isolation and consolidation of OS and applications.

Multiple VMs run on a single physical hardware that closely matches the VM logical characteristics.

Simulation requires limited binary instruction translation, since the VM is simulating a similar processor and most of the less privileged instructions can be simply passed to the underlying hardware processor for execution. Only instructions that are highly privileged (or might affect the state of other virtual machines) are controlled by the hypervisor and executed on a sharing basis.

This relieves the hypervisor from binary instruction translation, thus resulting in comparatively higher performance than emulation. When we refer to Full Virtualization in the industry, we generally refer to simulation.

Server Virtualization

Hardware Assisted Virtualization

Hardware Assisted Virtualization (HAV) use features provided by the hardware to improve performance of the simulated virtual machines. The approach use features provided by the host physical processor to improve efficiency of virtualization thereby improving the performance of the virtual machines.

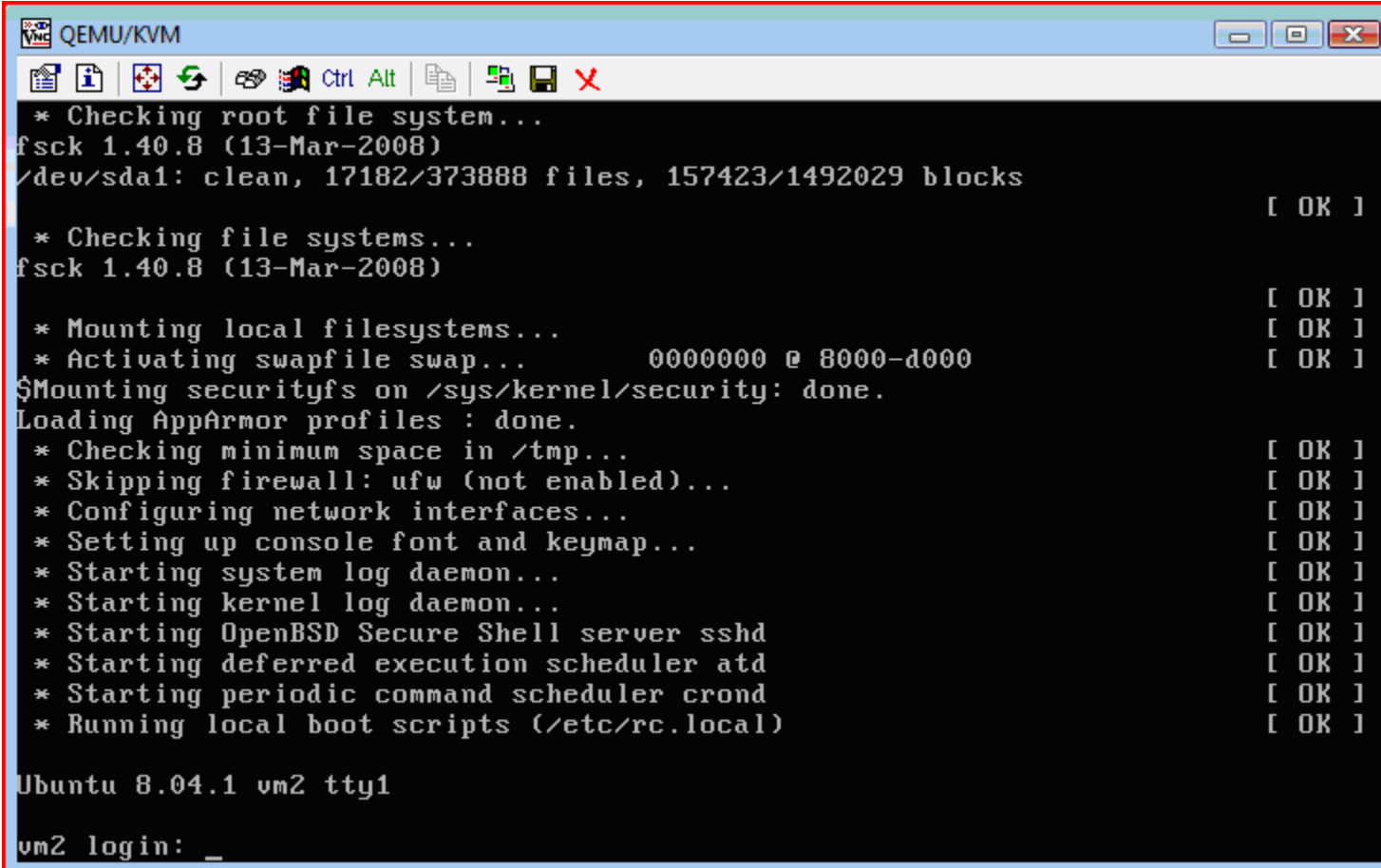
Intel and AMD implemented the hardware features required for HAV in their latest processors. The hardware features were named Intel VT-x and AMD-V respectively.

As seen earlier, full virtualization allows VMs to run in isolation without altering the state of another VM. A pre-requisite to achieve this, is to trap specific instructions that could affect the state of other VMs. Earlier x86 processors lacked the capability of generate a trap on these instructions and they have to be simulated in software. This resulted in performance hit to the VMs. After the introduction of Intel VT-x and AMD-V the hypervisor could rely on the hardware to generate the trap, thereby cutting down on the time required to virtualize and simulate the traps in software. This improved efficiency and performance of the virtual machines.

A common drawback of this technique was that not all hardware supported HAV. Hence, the widespread deployment of this was limited in presence of legacy or old hardware. But today, almost all new processors supported HAV, hence the hypervisors could make use of it at a larger scale.

Server Virtualization

The following example is of a KVM Virtual Machine running Linux 32-bit in Full Virtualization mode with Hardware Assisted Virtualization.



```
QEMU/KVM
* Checking root file system...
fsck 1.40.8 (13-Mar-2008)
/dev/sda1: clean, 17182/373888 files, 157423/1492029 blocks
[ OK ]
* Checking file systems...
fsck 1.40.8 (13-Mar-2008)
[ OK ]
* Mounting local filesystems...
[ OK ]
* Activating swapfile swap... 00000000 @ 8000-d000
[ OK ]
$Mounting securityfs on /sys/kernel/security: done.
Loading AppArmor profiles : done.
* Checking minimum space in /tmp...
[ OK ]
* Skipping firewall: ufw (not enabled)...
[ OK ]
* Configuring network interfaces...
[ OK ]
* Setting up console font and keymap...
[ OK ]
* Starting system log daemon...
[ OK ]
* Starting kernel log daemon...
[ OK ]
* Starting OpenBSD Secure Shell server sshd
[ OK ]
* Starting deferred execution scheduler atd
[ OK ]
* Starting periodic command scheduler crond
[ OK ]
* Running local boot scripts (/etc/rc.local)
[ OK ]
Ubuntu 8.04.1 vm2 tty1
vm2 login: _
```

Server Virtualization

Para Virtualization

Para Virtualization requires specialized hypervisor support in the form of an application programming interface (API) that could be used by virtual machines to request services from the hypervisor.

This specialized support from the hypervisor is known as the Hyper-call or Para-API.

The virtual machine operations could be run in the virtual context (simulation) or the Hyper-call context. The performance of operations in the virtual context is slower and suffers from performance degradation. It would be better if these could be run in the hypervisor context through the API calls. These results in significant improvement in performance compared to a fully virtualized machine.

However, the guest operating systems hosted in a virtual machine have to be modified or adapted to call these APIs instead of the regular machine instructions. Commercially available operating systems with closed-source are generally not amenable to such a setup.

Linux and FreeBSD provided required modifications as part of their kernel tree to run on Para-virtualized setups.

The most common example of a Para-virtualized machine is a Xen-VM running in Para-virtualized mode.

A specialized version of the Linux kernel is required for installation and the boot up of the Linux OS on Xen VM.

Server Virtualization

Hypervisors

Hypervisors may be described as a layer in software or firmware handling the execution of virtual machines. By definition, it may reside on bare-metal hardware or be a module in a standard operating system.

The primary purpose of a hypervisor is to share the underlying hardware resources by presenting a virtual hardware platform to the guest operating systems. The hypervisor also maintains strict isolation and ensures non-interference between virtual machines.

Hypervisors are commonly designed to run on bare-metal and in most cases use the Hardware-Assisted Virtualization techniques, if supported by the underlying hardware.

Most hypervisors on boot-up presents a control domain to the administrator. This console can be used to manage the lifecycle of virtual machines. The console typically allows for creation, modification, monitoring, migrating and deletion of virtual machines. The control-domain presents a virtual store where the configuration of virtual machines may be stored and run-time snapshots may be kept. This allows for easy management of the virtual machines residing on the control domain.

Ring Levels on x86 processors : X86 processors have built-in execution domains called rings. On an x86 processor, there are four ring levels – 0, 1, 2, and 3.

Level 0 is the highest privileged ring level. Highly privileged instructions on x86 require ring level 0 to run. Operating systems, like Linux and Microsoft Windows run in ring level 0. The user space applications are run in Ring Level 3. Other levels are not used. On a virtualized platform, a hypervisor resides directly on the bare-hardware and the operating systems (multiple instances) reside on their respective virtual machines. The hypervisor runs in Ring Level 0 and the VM operating systems run in Ring Level 3. Any request from a virtual machine to execute a highly privileged instruction results in a trap exception to the hypervisor. The hypervisor on receiving the trap accepts the request from the virtual machine and runs it in Ring 0, on the behalf of the virtual machine. All requests for privileged instruction execution are controlled by the hypervisor.

Server Virtualization

Types of Hypervisors

Based on the architecture, the hypervisor can be classified as follows:

Type I Hypervisors

Type I hypervisors run directly on the bare hardware. The hypervisors are designed to be lightweight and offer more performance advantages than the Type II hypervisors with some exceptions.

Examples of Type I Hypervisors : **VMware ESX Servers**: ESX/ESXi runs directly on x86 hardware and provides a virtualized platform for VMs to run. **Citrix Xen Server**: Citrix Xen runs on x86 hardware and provides the same benefits as the ESX. Additional features implemented by Xen Server is the inbuilt Para-Virtualization support in addition to Full Virtualization.

Type II Hypervisors

Type II hypervisors reside within a mainstream operating system and provide a separate layer over OS to abstract virtual services for virtual machines. In most cases, the hypervisor layer can bypass the OS layer to offer certain direct-to-hardware virtual services. The direct-to-hardware services require support from the underlying hardware.

Examples of Type II Hypervisors : **KVM**: Kernel Virtual machine is designed as a module in Linux kernel. KVM uses the existing Linux OS framework to provide virtualization services. KVM is enabled only in the presence of Intel VT-x hardware on Intel platforms or AMD-V feature on AMD platforms. In the absence of these features, Qemu is used, which simulates an x86 virtual platform. Qemu can also be used to emulate other architectures on Linux/x86 hardware. The most common of these is the ARM, PowerPC, MIPS, Sparc etc. **VMWare Player, VirtualBox, Bochs**: These solutions use emulation to build a virtual platform over the existing Linux or Windows based x86 systems.

Server Virtualization

IBM PowerVM Hypervisors

IBM provides the PowerVM solution which is the leading virtualization solution for AIX, IBM- i and Linux environments on IBM POWER technology.

PowerVM provides a Class-I hypervisor with advanced features like Micro-Partitioning, Dynamic Logical Partitioning, Shared Processor Pools, Shared Storage Pools, Integrated Virtualization Manager, Live Partition Mobility, Active Memory Sharing, NPIV etc.

Server Virtualization

Common Considerations in Server Virtualization

Listed below are the common concerns and considerations involved in virtualizing the server infrastructure in an organization.

SPOF in Server Virtualization

Server virtualization is increasingly being used for consolidation of the physical infrastructure for a myriad of reasons – increased utilization, better manageability, flexibility, easy sizing etc. However, consolidation using virtualization also raises concerns on the availability of the infrastructure. Since, multiple physical servers are converted and consolidated to virtual machines to run on a single physical server, this raises concerns on the reliability of the single underlying physical hardware. The underlying physical server becomes a single point of failure (SPOF) for all the virtual machines hosted on it. This increases the risk multifold as compared to traditional physical infrastructure wherein, there was usually one application running on a physical server. The risk was low since the failure of the server would impact only one application. With an increase in the system configuration of the physical servers, it is now possible to place hundreds of VMs on a single physical machine. Solutions need to be found to mitigate concerns on the failure of the underlying physical server. The most popular approach is to have a redundant physical server of almost same capacity on standby, to adopt the VM in case the primary server fails. Virtual machines can also be run on both the physical servers in a mutual takeover configuration.

This configuration makes good use of the standby server by keeping it utilized, while providing availability to the other physical server. In case of failure, one server can take-over the VMs from the other server.

There are multiple solutions available in the market to provide the highly available infrastructure. Some examples: IBM PowerHA, HP Serviceguard, VMware-HA etc.

Server Virtualization

Patch Management

Patch Management is an important consideration when moving to a virtualized server infrastructure.

Virtualization allows an administrator to take snapshot of a running virtual machine into a file. This snapshot could be used to provision the machine at a later date. There's no need to repeat the configuration on the VM as the configuration could be cloned from a VM snapshot. Incremental patch updates may be done on the active or inactive VM and the same could be saved as an updated snapshot. Virtualization relieves the administrator of the burden to repeat the configuration and patch updates by providing a simple VM save-restore model.

The snapshot or VM image library/store saves administration time involved in keeping the servers up-to-date or provisioning new servers. The servers could be provisioned in a matter of minutes as compared to 6-8 weeks in traditional infrastructures.

While Server Virtualization optimizes and saves significant cost on most of the system administration tasks, it may also lead to proliferation of unaccounted-for VM images and mismanaged VM libraries or store. Proper management of the VM-store and processes are required to manage and maintain the provisioning, updation and recycling of the VM libraries.

Virtualization management tools and VM lifecycle management tools in the market address these concerns and provide a streamlined workflow to manage the virtualized infrastructure.

Server Virtualization

Some of the commonly available tools in the market for VM management are as follows:

- Parallels Virtual Automation
- Microsoft System Center Virtual Machine Manager
- VMM – Virtual Machine Manager for managing Xen and KVM Virtual machines

Migration of existing IT infrastructure

With an increase in adoption of Virtualization, more and more customers would want to move their applications to virtual infrastructure. An important part of this process is virtualizing the physical servers involved in running the application. To achieve this, an exact replica or a virtual machine (with similar OS, configuration, patch updates) has to be created to host the application. This is achieved through a popular mechanism supplied as part of any virtualization solution in the marketplace, called the P2V (Physical to Virtual).

Using P2V, the configuration of the physical server is transformed to an equivalent virtual image configuration. The virtual image can then be booted up and the application hosted on this.

With large datacenters, the P2V process can be automated and finished early with limited downtime and impact to end-users.

Server Virtualization

Licensing

Traditional Licensing for large applications was tied to the number of physical CPUs/cores running the application. When these applications are moved to a virtual platform, it becomes possible to run multiple copies of the same application in multiple VMs but still pay for a license limited by the number of physical CPUs on the machine.

This was realized by the application companies as a major loss in revenue. The licensing terms have changed ever since and now for most applications, the license is tied to the number of virtual CPUs running the application.

Licensing is an important consideration for virtualizing the server infrastructure. Important license terms and conditions have to be checked on an application to application basis to estimate the cost involved. The software license cost generally overshadows the hardware costs in the long run. Hence, the cost involved to license has to be properly worked out before planning to move.

Increased adoption of virtualization in the competitive market has resulted in favorable license terms for companies going in for applications designed for virtualized infrastructures.

Server Virtualization

Desktop Virtualization

Desktop virtualization refers to the technique of creating an abstraction of desktop clients or end-user computing equipment. This is not very different from virtualizing a physical server. The process involves creating a logical abstraction or a virtual image of the desktop and placing it on a centralized physical server.

The end-user has an option to access their virtual desktops hosted on the centralized server using remote desktop client (RDP, VNC etc.). This technique allows for consolidation of several desktops

in the company to a single centralized physical server. The end-user client equipment runs a light-weight client with minimal resources. Thin clients are about 10-20 times cheaper than desktops.

Depending on the solution and context, desktop virtualization can mean hosting critical desktop applications or placing the entire desktop environment of end-user desktops on the centralized server. The end-users can then access this applications and data from the centralized server without ever installing the application on their local desktops.

Desktop virtualization is achieved using a set of hardware-software technologies and systems commonly termed as VDI (Virtual Desktop Infrastructure). The idea behind VDI is to create a virtual instance of each desktop and store it on a centralized server. The applications hosted on the virtual instance can be triggered or run from any thin device (a thin client or a smart mobile device) thus enabling the user to execute the applications from remote location without requiring physical access to the type of hardware required to run the application.

A common requirement for enabling desktop virtualization using VDI is the availability of reasonable network bandwidth between the end-user client and the centralized server. The desktop virtual instances located on the central server can be easily powered down when not in use by the end-user. This kind of monitoring is generally not easily possible in case of physical desktops which are left running even when not in use.

Server Virtualization

Benefits of Desktop Virtualization

Desktop virtualization offers the same benefits as the Server virtualization from the cost, consolidation and utilization perspective. Specific benefits offered by desktop virtualization are as follows

- Easy management of remote desktop instances

Since the desktops are now centralized and located on a remote server, it is easy to keep the desktops up-to-date using advanced virtual machine monitoring and management software.

Easy snapshots enable easier backup of desktop environments. These snapshots can also be used to spawn new desktop instances for new users without going for a long installation and configuration cycles.

Easy to power-up or power-down desktops based on need and usage trends. This allow for a granular resource management on the centralized server. The freed up capacity can be used for other useful purposes or for accommodating new set of desktop users

- Improved desktop availability

The centralized server can be configured and monitored for high availability. This ensures that the desktop instances hosted on the server will have limited or no downtimes. Individual downtimes of desktops are also avoided. Also, the workflow involved in getting a desktop serviced is reduced.

- Application deployment and Licensing

Run-time optimization on usage of licenses can be done as it is easy to track the desktops that are currently powered off. Multiple simultaneous users of an application can be easily estimated and controlled to arrive at better licensing terms from different vendors.

Instead of going for a desktop based license for an application, a server based license for the application turns out to be lot more economical.

Server Virtualization

- Easy of application access

Hosted desktop instances can be accessed anywhere anytime from an end-user device like a smart phone or a thin client. This improves the mobility of the workforce.

- Sharing of performance-intensive applications

Access to performance intensive applications which were traditionally hosted on high-end machines were constrained by their location and the number of simultaneous users. With the centralization of these desktop machines, the applications can be hosted on a central server, and can be accessed over a network anywhere anytime by any number of users.

Constraints in desktop virtualization

- Network bandwidth: An important resource to enable desktop virtualization using VDI is the availability of a reasonably fast network. The network infrastructure has to be sized based on the number of simultaneous clients accessing the central desktop server. Any bottleneck in the network will have an impact on large number of users. This limits the widespread adoption of VDI especially when the end-users are geographically dispersed over many locations.
- Security of the network : Security of the network connecting the end-users with the desktop server is important as any kind of SPI (sensitive personal and business information) moving over the network can be easily compromised if required security infrastructure is not in place.
- Graphic Intensive Applications : Certain applications like graphic animations, movie editing, 3d graphics, games require quick response times when it comes to display and input response. These applications are not recommended for VDI.
- Application requiring direct access to peripherals or embedded debuggers /programmers used in the hardware design sector that require direct access to host connected serial, usb ports are not well tested on VDI.

Server Virtualization

Types of Desktop Virtualization

Desktop virtualization may be classified based on the back-end VDI technology, methods used and the extent of virtualization.

Centralized Server Method : In a centralized method, the desktop virtual machines are kept on a central server. The end-user equipment could be a thin-client capable of presenting a remote display to the end-user, very similar to a RDP protocol client. The hosted virtual desktops may be provisioned on-demand, be maintained in current configuration for longer runs, or be saved and archived in a desktop-repository on the server. One major requirement for this kind of model is a good network connection. With increasing number of users and the geographic dispersion of users, this model may not scale-out so well.

Shared Load Method : In this method, it is possible for the end-user client to take a copy of the hosted virtual desktop and run it locally. This enables the users to have an option when the network connection is not up to the mark or there is a need to do heavy graphic processing on the desktop requiring fast response with no packet loss. The checked-out desktop instances may be checked back in to the server using an incremental synchronization method on a periodic basis. This method brings down the dependency on the network and facilitates running applications with special response requirements. Remote execution of the desktop instances is also supported. One of the major drawbacks of this method is when the instances are run locally; the end-user equipment must be capable enough to run the load.

Client Hosted Method : In this method, the virtual instances reside and run on the end-user equipment with no exceptions. Access to the server is only required to manage the desktop instances and keep the associated configuration data on the server. End-user equipment with capable hardware is a pre-requisite for this kind of virtualization.

Server Virtualization

Anatomy of Server Virtualization

In this section, we'll take an example to understand the internals of a standard Server Virtualization solution. For our purpose, we'll be referring to an open-source virtualization product called Xen Server Virtualization. Xen Server started as a project at University of Cambridge and was later acquired by Citrix Company.

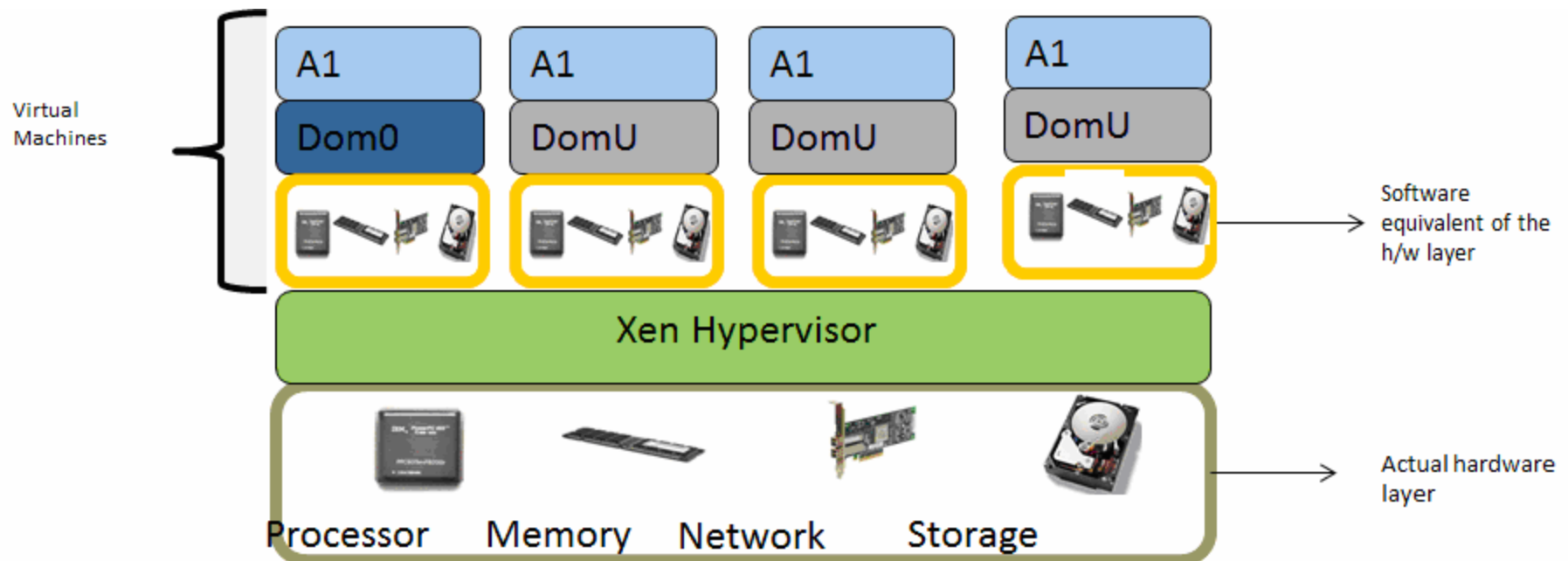
Xen Server allows multiple guest OSes to run on virtual machines managed by an underlying layer called the Hypervisor. Xen Server provides bare-metal server virtualization. This basically means that the hypervisor resides over the hardware. Hence, it is a Type-I Hypervisor. The hypervisor has complete control on the four major hardware resources in the system – CPU, Memory, Network and Storage. The hypervisor multiplexes these resources among the virtual machines.

There are three major layers in a Xen Server:

1. **Hardware Layer** : The hardware architectures supported by Xen are x86, x86_64, ia64 and ARM. Xen will generally make use of the virtualization capabilities provided by the hardware. While running on x86 and variants, it typically uses either Intel-VTx or AMD-V virtualization features. Recently, there are additional features supported in hardware called the SR-IOV for IO Virtualization. These will be discussed in later chapters.
2. **Hypervisor** : The Xen hypervisor provides the required virtual platform for defining and running the virtual machines. The hypervisor supports two kinds of virtualization – Para and Full Virtualization. In case of Para Virtualization, the Xen hypervisor provides the Hyper-call APIs which the guest machines use to request services from the hypervisor. Para-Virtualization is not a pure-virtualization; rather it is implemented as a set of APIs. The virtual machine OS hosted on Para-Virtualized VM requires modification to use the Hyper-call APIs. Only OSes with explicit support for Para-Virtualization can be run in this mode. Full Virtualization provides full emulation of the underlying platform with help from the hardware. No OS modification is necessary to run on a fully virtualized Xen machine.

Server Virtualization

3. **Domains Layer** : A differentiating factor of any virtualization solution is the platform and the manageability tools it provides to the system administrator. Xen Hypervisor boots into a control domain (Dom0), a default OS, typically a version of Linux. This control domain provides the administrator with a platform to create, manage, store and delete and track virtual machines. The control domain has complete access to the hardware on the physical machine. The control domain can be seen as the first virtual machine on the physical system. However, starting with the second virtual machine, the virtual machines are termed as DomU. The DomU are the actual user defined virtual machines. Using the control domain, the administrator can tailor make the configuration of DomU. A DomU could be a para-virtualized machine or a fully virtualized machine, as discussed earlier.



Storage Virtualization

Overview

There is an ever growing demand on storage infrastructure to store, query and analyze this information.

Traditional computers used internal or tape drives to store and backup information. Over the last few decades, specialized external storage systems have been used to manage information. These specialized external storage systems are known as Disk Arrays.

Disk arrays have provided basic advantages over the internal disk drives. One major advantage is that these allowed sharing of the storage among several machines or over the storage network. Another advantage was these arrays were built with internal redundancy that allowed for mirroring of the data. So, more than one copy of the data could be stored in real-time. When one copy of data fails, the other copy is still available. These were known as RAID Arrays.

Storage Virtualization

Benefits of Storage Virtualization

There were major shortcomings of the conventional storage systems

Interoperability: With the spread of storage systems, different vendors developed their own storage system designs and the interoperability between these storage systems became a major constraint to moving data across in reasonable time. Significant computing power gets used to retrieve, move or store data. Also, the interoperability caused compatibility issues between the servers and storage systems. A constrained infrastructure does not scale so well and lead to a lot of wastage.

Manageability: The heterogeneous storage systems had their own proprietary implementations of standards which may or may not work with other storage vendor equipment. These dependencies also constrained the development of industry-standard manageability software and tools, to better track and optimize the storage infrastructure.

Scalability: With the interoperability constrained and the usage of legacy storage protocols, these storage systems could not be scaled on demand or required major workarounds to accommodate increased storage requirements.

To counter the shortcomings of conventional storage systems, new protocols and implementation mechanisms were required to build more flexibility, manageability and interoperability into the storage systems. This was achieved by building a logical layer (hardware independent) over the actual storage hardware. The logical layer design was achieved using a set of distinct features collectively known as Storage Virtualization. The features built into the logical layer may vary from vendor to vendor, but almost all of them ensure the following:

Interoperability: Adherence to industry standard protocols for storing and moving data across, for example: SAN, NAS, Fiber Channel, iSCSI. This ensured that the logical layer can talk to heterogeneous systems irrespective of their internal implementations, as long as these adhere to certain standards.

Manageability: The logical layer provided a standard interface to manageability software to probe, interact with the underlying storage systems, irrespective of their internal physical implementations.

Scalability: By virtualizing the storage, manageability has become hardware independent allowing for addition, removal of additional storage hardware on demand with limited or no constraints.

Storage Virtualization

Types of Storage Virtualization

The commonly used protocols for storage in the industry are classified broadly in three categories:

- **Host based storage virtualization:** This refers to the components built into the host server operating system and the host hardware to enable storage virtualization.
- **Storage based virtualization:** This refers to the use of exclusive host-independent storage protocols to access storage systems. The virtual layer is built into the storage firmware to enable the separation of the hardware from the logical abstraction.
- **Network Based:** This method uses network based protocols to access storage systems. The storage systems basically host a file system and the files on the file system can be accessed over the network using a network protocol similar to NFS.
- **Hybrid Model:** This method uses a combination of Network and Storage protocols to access storage. Remote disks can be presented to a host server as local disks using iSCSI. All storage IO to the iSCSI disks is encapsulated as network packets and moved over the network connection between the host and the storage server.

All the above protocols meet the basic demand of interoperability since these adhere to industry standards for operation. Storage Virtualization can be achieved using any one or combination of the methods mentioned above.

Storage Virtualization

Host Level Storage Virtualization

The host based storage virtualization refers to a set of techniques that use host server OS and hardware to achieve the aims of manageability, scalability and interoperability.

Common host-based mechanisms used for virtualizing storage are as follows :

LVM (Logical Volume Manager)

Earlier operating systems provided a device node for every storage device connected to the system (either locally or on the storage array). An application that intends to use the device has an option to either access the device in a raw mode or through the file system mode. In either case, once the device file is specified in the application code or in the filesystem configuration, it is difficult to change it without actually going in for change in the application or filesystem configuration files. For example, an application opens a device file `/dev/sda1` for writing data on Linux. Assume the device `sda1` is of 4GB capacity. Over time, as the application writes to this device, the device runs out of capacity. The only option to increase capacity is to add another device `/dev/sda2` to the system and change the configuration of the application to use `sda2` device. Assume, the application would want to read `sda1` before writing to `sda2`, this creates a problem since the application is designed to use one disk at a time. Let's say after changing the application code, the application can now use two disks at a time. What happens when the second disk runs out of space. So, there's no transparent way of adding storage without significantly impacting the application.

Logical Volume Manager is an important feature for virtualizing the underlying storage. Logical volume manager creates an abstraction over the disks in the form of logical volumes and volume groups. An application that wishes to use the storage is allotted the storage in the form of logical volumes. The application code can be configured for logical volumes. Logical volumes has the capacity to expand or contract in capacity based on whether the disks are being added or being removed from it. However, the logical volume name remains the same irrespective of the number of disks underneath. Since the application only makes use of the logical volume name, the application configuration does not need to change when the disks are added or removed.

Storage Virtualization

Logical Volume management brings in flexibility in managing storage assigned to an application. It virtualizes the underlying storage by creating a logical layer and separating the physical components from the logical components. Any change in the physical characteristics does not impact the logical layer and is hidden from the applications.

Host Based Mirroring

The abstraction created using Logical Volume Management, also allows an administrator to implement Host Based Mirroring. Typically, a logical volume in a volume group can be created in a mirror configuration. The logical volume, thus created will act in a RAID 1 (mirror configuration), i.e. the data written on one physical volume will get replicated on the other physical volume in real-time.

A basic extension of the mirror configuration is to have one physical volume created out of local devices, and the other physical volume created out of remote devices (presented through an iSCSI mechanism). This allows for a Disaster Recovery configuration or a simple high availability configuration wherein data written to the local disk gets replicated to a remote disk.

The end-application does not need to be aware of this special configuration, it is still writing to a logical volume. This logical volume, in turn is specially configured in a mirror configuration.

Thus, using host-level virtualization completely hides the storage hardware intricacies from the applications and makes the application-storage interaction simple to manage and maintain.

Storage Virtualization

Storage Level Virtualization :

Storage level virtualization refers to the implementation of logical layer in the storage array firmware.

The logical layer separates the physical components in the storage array and provides an abstraction for access to the storage. The logical layer can talk to heterogeneous storage systems from multiple vendors without exposing the details to the end user host or application.

The logical layer abstraction also enables better manageability of the storage system by allowing the storage allocation to be done in logical units rather than in actual physical units. The abstraction takes care of mapping the logical units to the actual physical units.

Actual implementation of the logical layer/abstraction may vary. Typically, a physical storage array is managed through redundant storage controllers. The storage controller builds an abstraction of the underlying storage hardware. A common technique is to logically categorize the physical disks into LUNs (Logical unit identifier). A logical unit identifier can be defined over a fraction of a disk or could comprise multiple disks, coupled in redundant configuration or in striped configurations for performance reasons. A LUN thus defined forms a basic unit of allocation for the host server. The LUN can be presented to the host server for use.

The separation of the physical disk from its logical definition formed the basis of storage virtualization within a single array.

With the proliferation of storage disk arrays and the need for more storage, it wasn't possible to limit ourselves to just one storage array. A typical datacenter would generally have multiple storage arrays to accommodate the growing storage needs.

Storage Virtualization

Multiple storage arrays also made storage management difficult for the system administrators as it required individual controller configurations for each array. Moreover, it became difficult to move storage volumes within the datacenter as each is restricted to a particular storage array.

SAN or Storage Area Networks provided a solution to this problem. An advanced protocol like the Fiber Channel protocol is used to form a network of storage arrays. The host and the storage still used SCSI protocols for communication but over a Fiber Channel.

This greatly improved the manageability of the storage infrastructure and unified the storage array infrastructure in the datacenter. It is now possible to move storage around without actually physically shifting or connecting the storage to a host. Since the storage is now available over the network, it was only a matter of simple reconfiguration to reassign storage to another host, if need be.

SAN also greatly enabled the setting up of disaster recovery infrastructures. It is now possible to set up a secondary storage at a distant location and made it easy to keep it in sync using the storage array communication through Fiber Channel over IP or iSCSI.

To enable this, additional logical layer to enable Fiber Channel over IP was developed which virtualized the storage communication over IP networks.

SAN enabled better manageability and accessibility to heterogeneous storages separated over larger distances while still maintaining the block level SCSI communication standards for communication between the servers and storages.

Storage Virtualization

Network Level Storage Virtualization

One of the other popular techniques to provide storage virtualization is called NAS or Network Attached Storage.

NAS provides a file system level abstraction of the storage, i.e. the remote storage file system can be locally mounted by a host as any other file system. The IP network is used to manage any communication or meta-data transfer between the NAS Client and the NAS Server.

NAS uses commonly used network protocols to provide access to remote file systems for example: NFS

A NAS Server is a specialized computer optimized for storage function. The server is designed to provide a fast path access to its attached storage over the IP network. This calls for an efficient network connection and an efficient storage distribution algorithm. A NAS Server typically presents the storage in the form of file-system to NAS Clients.

NAS (using NFS) builds a file-system abstraction on the end-user NAS client, thus enabling the use of the remote file-system as if it locally mounted.

An advanced form of virtualization in NAS is called the NAS server virtualization. In this case, the NAS itself virtualizes into multiple virtual NAS servers, each hosting a set of file systems for serving to different clients.

In other case, the NAS can provide an abstraction over the set of file systems hosted by NAS. This way NAS can re-route host requests to the correct filesystem.

Network Virtualization

Overview

The ever increasing complexity of the networks imposes the following constraints:

1. The management of a network infrastructure tends to get complex when there are different physical devices and medium involved.
2. Also, with improvements in technology, the networks have become capable of transmitting 10 Gpbs or more data in a second over one channel. To accommodate low bandwidth users alongside high bandwidth users, there must be a way to split up the bandwidth on the network on a static or dynamic basis to ensure overall optimum utilization.
3. Also, with the server and storage being virtualized, the network must be capable of communicating with virtualized servers and storages.
4. Security is another major concern in distributing information. To securely distribute or share information over a public unsecured network requires building up an encryption channel over the public network.

There's a need to virtualize the network infrastructure to make it more manageable while conforming to open standards for manageability.

Network Virtualization

VPN

Security is an important aspect of network communications. Information may be rendered completely useless if it cannot be shared or transmitted with the right levels of confidentiality. This kind of information includes but is not limited to confidential company data, sensitive private information, finance data, medical transactions and others.

The public network is insecure. By default, all the data transmitted over the public network or the internet can be seen or modified by anyone. The end-users or the communicating parties (communication endpoints) have to take special steps to encrypt the communication at their end to ensure confidentiality of data. Also, the parties can setup a continuous encrypted channel to communicate or share data on a continuous basis. This channel is generally destroyed once the communication session is completed. The channel can be later established again between the parties to share information.

The security of this channel can be achieved in various ways

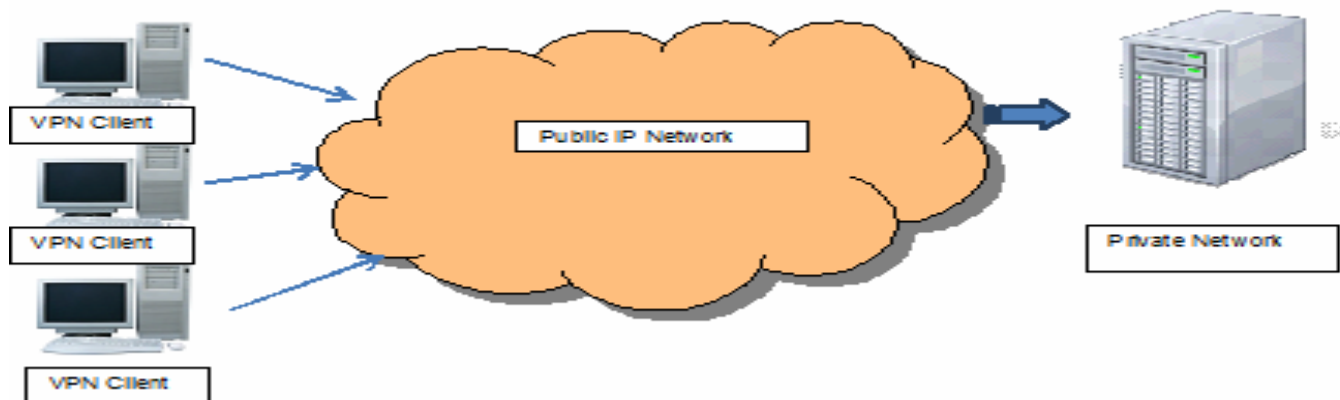
- Use of intermediate network hardware – routers to encrypt and decrypt the data on the channel. The end-user computers(or endpoints) do not require any special software or hardware to set up the secure channel
- Use of end-user software or hardware clients – In this case, a secure server is set up at the destination and a secure client is set up at the client-end. The client can communicate or connect to the network at the server end by establishing a connection with the secure-server.

Network Virtualization

With an increase in movement of workforce, the organizations need a way to connect their employees to the main network in a safe, secure manner, while ensuring that the employees see no major differences in how the network resources get accessed or used.

Even major stock exchanges need to ensure that there is a secure communication channel that allows the stock depository account holders access to the current real-time stock prices and news.

The technology to set up a secure channel is called Virtual Private Network. As the name implies, we are not setting up a new physical network. Any existing network like the public internet is used. However, VPN sets up a logical layer over the existing network using a secure tunneling protocol. A secure tunneling protocol can be implemented at the endpoints or on the network as seen above. Either way, we are setting up a secure channel for communication.



Network Virtualization

VPN or Virtual Private Networks connect remote users with their central company servers over the public network while maintaining security and confidentiality of data that is being transmitted. This gives the effect of being connected directly to the company LAN. All operations that can be performed on the company LAN can be performed on the VPN as well.

VPN does not require any dedicated leased lines to maintain confidentiality. VPN creates a logical network over and above the physical public network that is already in place. This is achieved through a range of technologies that are available from Layer-1 to the application layer of the network protocol stack.

Some of the common ones are: IPSec, SSL/TLS, Secure Shell(SSH), Microsoft's Secure Socket Tunneling Protocol (SSTP), CISCO Layer 2 Forwarding Protocol and the more recent Layer 2 Tunneling Protocol etc.

VPN is a cost effective mechanism to establish a network as compared to secured leased lines as it improves utilization and cut down bandwidth costs. It also transfers security responsibility back to user hands.

A common open source VPN implementation is OpenVPN. This is based on the SSL (Secure Socket Layer) protocol. This allows users to connect to remote network using standard security authentication protocols. All traffic over the OpenVPN channel is encrypted using SSL.

Network Virtualization

VLAN

A traditional LAN is a collection of hosts that are physically placed together and connected to the same switch ports. The hosts in a LAN share a common broadcast domain, a gateway and a common subnet.

One of the primary limitations of a LAN is that the machines have to be placed and physical connected together. Wouldn't it be good if we could group the machines not by their physical location but by their function?

For example: If we want to group machines in a university. We want to group all the machines belonging to a department together. The machines/users may be scattered in different labs, professor may have their own desktops or student might have laptops that connect to the university network. By grouping all these geographically dispersed machines together into a common entity, it'll be possible to manage and maintain the machines much better. It would also be possible to come out with department specific statistics regards to the bandwidth use. Sharing of data and files securely within the department becomes a lot simpler since all these machines belong to the same logical group. Hence, the definition and configuration for security and manageability becomes a lot simpler.

Virtual LAN or VLAN provides a mechanism to achieve the goal of logically grouping the machines irrespective of their physical location in the network.

Network Virtualization

This is a massive improvement over traditional LANs and offers the following advantages

- **Manageability:** VLAN greatly improves the manageability of physically dispersed machines by grouping these under a single logical network. IP assignments and allocation/de-allocation of IP resources becomes simpler to manage and track.
- **Data Sharing:** Data sharing within a single logical network is lot easier when compared to sharing data across different physical network. Easier firewall rules allow unrestricted sharing of data within the VLAN. This also simplifies setting up of firewall rules.
- **Tracking Usage:** Resource tracking is easier or could be made more granular since the machines are operating in a single logical network.
- **Reduction of broadcast traffic on switch:** A set of ports on a switch can be designated as belonging to VLAN1 and the remaining set can be assigned to VLAN2. Any broadcast traffic on VLAN1 will not interfere with VLAN2 thus isolating the traffic at a more granular level and improving efficiency.
- **VLANs are also used to share a high network bandwidth channel among several different VLANs.** The logical separation of traffic due to VLAN ensures that the traffic of one VLAN will not interfere with traffic on another VLAN sharing the same channel.
- **Server Virtualization and VLAN:** Communication between the virtual machines on the same physical host can be managed internally using a hypervisor defined VLAN between the virtual machines. As a result, the traffic from one virtual machine (for example: running a webserver) can be routed internally to another virtual machine (for example: running a database server) on the VLAN. Only when the VM need to communicate to an external VM (located on another physical host) does it need to use the physical NIC on the machine. Also, it is possible to define multiple VLANs on the same physical host thus allowing for more granular grouping of virtual machines on the physical host.

VLAN plays an important role in making the traditional LANs more flexible and geographically dispersed.

Application Virtualization

Overview

Challenges

Listed below are the common challenges in using applications in the traditional install, use, and update model

- Applications with the correct version have to be installed locally on machines with supported operating system and architecture. The same process has to be repeated over a large number of computers in an organization.
- In case, an application requires patch updates, the application need to be patched separately on each machine.
- License usage tracking of applications is complex, as it involves special software tools that run on individual desktop to check on the installation status and the usage metrics
- Applications thus installed can't be made available easily to users on travel. The process is cumbersome involving re-installation of the OS and transfer of licenses to the mobile equipment.
- During hardware refresh cycles, the applications require reinstallation and reboots

Application dependency on the underlying operating systems and platforms is a major reason behind the complexity of maintaining applications.

Application Virtualization

The solution

If an application is made independent of the underlying platform, it is possible to do away with most of the difficulties and complexity of application lifecycle.

An application can be made platform independent by building a logical layer over the target platform. This logical layer presents a consistent interface to the applications irrespective of the underlying platform.

A very good example of the logical layer is the “Java Virtual Machine (JVM)”.

JVM presents a consistent interface to a java program irrespective of the underlying platform. A java source gets converted to byte-code on compilation. The byte-code can then be run on any available JVM platform without any modifications or installation and is independent of the underlying OS. The JVM platform is available for all major operating systems – Microsoft Windows, Linux, UNIX platforms etc.

Another example could be the CLR (Common Language Runtime) built over .NET framework.

The applications built for the logical layer need not be installed and can be provisioned on-demand locally or from a remote server.

This opens up new opportunities to streamline the application lifecycle activities. The applications may be placed on a remote server. The clients can connect to the remote server and place requests for an application. The application can then be streamed to the user desktop. The user need not install the application to actually use it.

This also opens up opportunities for better license management and better usability tracking since the applications are hosted on the centralized server.

Whenever, a user moves to a new desktop or uses a laptop on the move, all that the user need to do is to connect to the application server and request for the required application. This relieves the end-user of the installation and configuration of the application on the local machine before it could be used.

Application Virtualization

Architecture

Application Virtualization Infrastructure can be designed in-line with desktop virtualization techniques and involves a server-client model for servicing applications.

The steps involved in the application lifecycle are as follows

- An user requests an application from the remote application server
- The application server streams the application on request from the user
- The user desktop runs the application in a virtual environment that shields the application from the underlying platform
- On completion of work, the application is removed from the user's desktop
- The application server updates the application usage counters and the simultaneous counters for accounting purposes
- The application is kept up-to-date with the latest patches and license renewals on the centralized server. This relieves the end-users of license and patch management.
- New applications are added to the central-server and the information on these is propagated to the end-users. These applications can be used as soon as these are setup on the central-server. The end-users need no changes on their desktop to use the new applications. This saves time proportionate to the number of end-users in the organization
- Users on the move can establish a VPN connection to the central server to request and access the application

Application Virtualization

Benefits of Application Virtualization

Application virtualization provides the following benefits to an IT organization.

- Application virtualization builds a virtual layer on the user-desktop which allows the applications to run and execute without any major dependency on the specifics of the OS and the platform.
- Common examples are Java virtual machine, Microsoft .NET framework, Linux emulation layer – Cygwin, Interpreter layers from perl, python etc.
- These solutions provide varying degree of independence from underlying platforms and allow an application to run on multiple platforms
- Provides streamlined migration of user-applications from one OS platform to another. With application virtualization it is possible to mix and match heterogeneous OS platforms to run distributed applications.
- Logical isolation of applications from the operating system offer distinct advantages over native applications. This protects the applications from malicious or poorly written code that may hamper the system or degrade system's performance.
- When compared with VDI (Virtual Desktop Infrastructure), application virtualization is a light weight solution since only the application code is virtualized. In case of VDI, the entire stack comprising of hardware, software and applications is virtualized. This saves on precious network bandwidth and also requires smaller data store to store virtualized applications

Application Virtualization

- Applications can be deployed on-demand and in seconds through application streaming techniques
- Usage of the applications hosted on the central server can be easily tracked. Simultaneous use of application can also be tracked for license purposes. This helps in fine-tuning the overall licensing requirements for the applications.
- Application or application sub-features that are not frequently used can be removed from the central server. This saves on resources and license costs for these applications.
- Application can also be made geo-aware. Depending on the user's location, the appropriate language version of the application can be streamed to the end-user computer. For example: If a user has requested an application from China, the Chinese version of the application can be streamed to the user's desktop.

Case Study on Virtualization

Customer IT Landscape

We'll take an example of a company named "FiberCoils Co". FiberCoils is involved in the manufacture of fiber optic cables for the broadband and telecom sectors. The company maintains an extensive supply chain to procure raw materials, extensive dealership network to market finished products etc. The company maintains a datacenter to consisting of about 50 high-end servers. These 50 servers overall are required to maintain the day-to-day IT activities of FiberCoils. A list of functions served by these servers is as follows:

- Mail Servers: Provides email services to all Fiber-Coils employees, internal suppliers, dealers and logistics personal (Q: 1)
- Web Servers: The webserver maintain information on various products and services of the company and provide front-end for data-entry. (Q.1)
- Database Servers: The database servers maintain the company's database. Each division has their own set of database servers to store information (Q.1)
- ERP Systems: The ERP systems provide functions for supply chain management, financial management, sales management, business intelligence, asset management, invoice tracking, Employee payroll, HR etc. (Q.3)
- Firewalls and Security Systems: These systems provide end-to-end IT security to the datacenter and maintains compliance of these systems with the existing compliance laws (Q.1)
- Factory Systems: The factory server systems control the critical systems required for manufacturing processes (Q.2)

Case Study on Virtualization

- Test and Development Environment: These systems serve as the test-bed for software upgrades, patches and staging of new software to production systems (Q.1)
- Remote Desktops and Servers: Remote desktop for use by offices that are located at remote places, upload data to the main datacenter on a periodic basis. (Q.30)
- Disaster Recovery Site: The disaster recovery site datacenter is an exact replica of the critical systems running at the primary site (Q.10)

All the system functions listed above are currently running on a physical footprint or physical servers.

Case Study on Virtualization

Triggers for virtualization

The customer is planning to optimize the current infrastructure and improve overall manageability, while reducing costs. In addition to this overarching goal of optimization, the customer had additional triggers to begin looking at the latest trends in IT, which includes virtualization.

- A large set of servers in the customer's datacenter are approaching the end of their maintenance lifecycle. The customer plans to renew/refresh these machines at the next hardware refresh lifecycle
- The growth of the customer IT datacenter has largely been ad-hoc, primarily driven by immediate need and requirements of individual departments in the organization. This has created several silos within the IT infrastructure. The customer is now planning to have a better control on the IT procurement and manage cost and cash-flow efficiently.
- There are no specific standards followed for managing the servers. A large part of the administration is still manual requiring several system administrators to manage the growing data-center.
- A large set of machines run at an utilization level of 0-20% maximum. The existing resources could not be utilized efficiently due to the inherent inflexibility in the infrastructure.
- Various departments with the company have set up their own security systems and security policies which have resulted in less transparency at a company level. Resources could not be shared easily and require lengthy approval processes.
- About 10% of the machines in the datacenter are legacy platforms running legacy applications.
- The IT datacenter is running out of floor-space. Additional space to accommodate growing needs is not available unless a new datacenter is set up for this requirement.

Case Study on Virtualization

Server Sizing

All qualified servers for virtualization will be classified under two major categories – hypervisors and VMs. A section of physical servers will be configured as hypervisors. The sizing for the hypervisors would be based on the following points

- Add up the resource requirements for all the qualifying physical servers.
- Add 35% headroom to the calculated total capacity to handle peak loads
- Add 10% extra capacity to the calculated total capacity to account for hypervisor and management overhead
- Get a list of physical high-end servers that match the total capacity required. These servers will be configured as hypervisors and use to host other VMs. The OS and applications hosted on these physical servers prior to virtualization will be deployed as VM on these servers.
- Remaining physical servers can be released or freed-up for other uses
- In case of a DR infrastructure, the physical servers can be consolidated on the DR site using the same procedure.
- In case of a highly available infrastructure, a VM corresponding to the passive node will be created and placed on a different hypervisor than the one hosting the primary node VM.
- Restrictions on the movement of VMs on the hypervisor nodes will effect sizing and additional physical nodes / capacity need to be accounted for, in this case.

Case Study on Virtualization

Server Criticality

A set of servers that serve critical functions in the organization can be retained as physical servers. A server can be marked as critical if it qualifies for one of the following criteria

- It serves a critical function in the organization, with heavy reliance on hardware and system(OS) level features. For example: a database server. The database server makes extensive use of OS and hardware facilities, in some cases directly. Virtualizing these servers leads to major downgrade in performance due to the extra emulation layer required or a slower virtual IO mechanism in VM.
- Legacy Applications and Systems: Legacy systems running legacy applications are not easy to port to open standards. Most of the virtualization solutions are designed around open-standards. Legacy systems also have a huge dependency on the underlying OS and hardware for operation.
- Servers that constantly run at a higher utilization about 95% of the time.
- Security systems that require hard-isolation to comply with existing compliance laws.

Case Study on Virtualization

Provisioning

A central idea behind virtualization is to bring down the average provisioning time of servers. FiberCoils has seen a rapid growth in demand and reliance on IT in the last few years. This demand will continue to grow in the future with the growth of the organization. New demand will require new servers. The current provisioning time for a physical server in FiberCoils is about 6-8 weeks. Virtualization can bring down the provisioning times to about 1 day. The following areas would see a drastic improvement in turnaround time once virtualization is deployed.

- A snapshot of a VM can be captured and placed in a central datastore. The snapshot will have the entire configuration saved right from the hardware, OS and the customer specific application. In future, if there is a need to bring up a server with the same configuration, it would be as easy as to copy the VM snapshot and power it on. Any number of copies can be created from the saved snapshot, thus resulting in a drastic time savings for provisioning the new servers.
- The remote desktops used by FiberCoils can be consolidated onto a central server using Desktop Virtualization. Addition of new desktop would be as simple as adding a VM to the central desktop server and sending the details to the end-user.
- Hypervisor based monitoring tools can be used to track the running status of VM, and the application monitoring can be used to track the running status of the applications on the VM. The monitoring tools provide an up-to-date run-status, thus eliminating the delay in detecting server crashes and down-time. This reduces overall down-time in the organization.

Case Study on Virtualization

Proximity and Locality

- Virtualization enables dependent servers to be placed together on the same physical server and colocated locally to enable better communication and reduction in communication and IO latency. FiberCoils could make use of locality in the following ways:
- The application servers and the database server VM can be colocated together. The server VM will have a VLAN between them on the same hypervisor. Hence, the traffic between these servers will pass through the VLAN without being routed on the physical network interface. This way the network latency can be reduced.
- The firewall, proxy and the VPN servers can be colocated on the same hypervisor, thus allowing for reduced network latency between these servers. This is implemented through VLAN.

Case Study on Virtualization

Transition Tools for Virtualization

A virtualization vendor provides its own custom tools for consolidating a physical infrastructure. However, these tools can be broadly classified under the following major categories.

- Virtualization assessment and planning toolkits: These toolkits in-effect take the current IT landscape as the input and current constraints and provide a plan for virtualization and consolidation
- TCO calculator: These worksheets provide mechanisms for calculating the total saving and the cost of ownership after moving to virtualization
- P2V: Physical-to-Virtual tools are used to move from a physical state to a virtual state. These tools enable an administrator to create a virtual copy of the physical server for deployment.
- Manageability Tools: These tools provide mechanisms for creating, managing, storing and deleting virtual machines in the datacenter. In addition to the basic functionality, the tools provide mechanisms for monitoring and ongoing optimization of the current IT infrastructure.

Cost Savings

Typical cost saving experienced by FiberCoils after moving to virtualization is in the range of 40%. This includes saving on the datacenter space, physical servers, operational costs and easy manageability of the IT infrastructure.

Introduction to Cloud Computing

Overview

Cloud provides a mechanism for delivering IT as a service to business with the following characteristics:

- You do not need to own a datacenter to use hardware or software services
- You have an option to subscribe or unsubscribe to services delivered in the cloud.
- The cloud service provider will provide you a SLA for the services defined in the service catalog.
- The cloud services could be accessed using a standard interface
- The cloud services are available on demand
- It can scale up or down based on your usage. The excess capacity goes into a general pool and used by someone else
- Monitoring allows for more granular optimization of usage of resources in the cloud
- The subscriber gets charged for the services subscribed and the periodic usage
- A large part of cloud operations is automated. The process is known as orchestration.

Central to cloud technology is the information technology service framework – i.e. delivery of information technology as a service. The importance of information and the technologies supporting the creation, management and distribution of information cannot be underestimated in this age. An important aspect of cloud is that it ties a business need to a technology definition. With Cloud, it becomes easier to convert a business requirement directly to a technology requirement. A Service Catalog defines what a particular instance of cloud is designed to provide to end-users. A configuration management system provides a central repository for all the configuration items present in the cloud. In short, Cloud defines a framework to deliver IT as a service in the most efficient and the fastest way possible without a need to actually own the resources required to achieve the same, all along providing a level of transparency and monitoring not feasible in earlier paradigms.

Introduction to Cloud Computing

Virtualization and Cloud

Virtualization and Cloud are sometimes confused as one technology in certain scenarios. Virtualization and Cloud are two distinct technologies. However, in certain aspects these are related to each other. The following points demonstrate the overlapping aspects of Virtualization and Cloud.

- Virtualization is only an enabler for Cloud. Virtualization provides the required infrastructure flexibility to cloud by virtualizing the resources which allows for easy provisioning and management of these resources across hardware pools
- Virtualization is only one of the eight major building blocks of Cloud.
- Virtualization is most relevant to IaaS (Infrastructure as a Service). However, PaaS(Platform as a Service) and SaaS (Software as a Service) could largely be achieved without virtualization. Increasingly, PaaS and SaaS units are being packaged as VMs for easy deployment.
- The real difference between Cloud and Virtualization comes from the business aspects of Cloud and the Service management framework built into Cloud. Virtualization is a technology without no direct business benefits other than saving cost to business. On the other hand, Cloud connects a business requirement to technology and can be used to map and fulfill a business requirement.
- A virtualized infrastructure has limited automation built-in. However, Cloud based infrastructure is built around complete automation and orchestration.

The elasticity and the flexibility in the Cloud are built on the foundation of Virtualization. It is recommended and an industry practice to first consolidate a physical infrastructure using Virtualization before moving in for a Cloud Transformation.

Introduction to Cloud Computing

Virtualization and Cloud both brings saving to the business due to consolidation and the flexibility and increased manageability. Listed below are the areas and the relative savings

Parameter	Virtualization	Cloud
Utilization	Typically 60-70%	Typically 60-70%. However, it is possible to share resources across pools (physical hosts) and maintain the utilization levels. Spare capacity available on demand
Provisioning	Manual, 1 day	Automatic, On-demand within minutes or few hours
Monitoring	Comparative ease in monitoring using automated tools. However, need manual intervention to take care of any failures	Typically automated. No manual intervention required due to advanced orchestration capabilities built into the cloud
Sizing	Easier to resize. However, manual intervention required to resize for new requirements	On-demand automatic rescaling of the resources
Staff for Administration	Reduced number of Full Time employees	Typical reduction in number of employees required to manage the infrastructure. On an average, FTE reduction is generally in the range of 1 administrator for 400 cloud instances
Cost	Initial hardware cost reduced due to sharing of hardware assets and due to increased utilization	In most cases, the initial hardware cost is almost negligible. There'll be running cost on a monthly basis.
Optimization	Easy to share resources and re-balance loads on the virtual machines on the same host. However, re-balancing across physical hosts require advanced features and planned downtime	Easy to share resources and re-balance loads across resource pools. No manual intervention required to move resources or resize resources for an application.

Introduction to Cloud Computing

Anatomy of Cloud

A cloud is composed of eight major components.

1. Provisioning and Configuration Module
2. Monitoring and Optimization
3. Metering and Chargeback
4. IT Service Management
5. Orchestration
6. CMDB (Configuration Management Database)
7. Cloud Lifecycle Management Layer
8. Service Catalog

Each component serves a specific function in Cloud. Not all components are present in every Cloud solution available in the market. Cloud vendors may choose to implement a subset of features to suit their end-user requirements.

The choice of components may also depend on the Delivery model (discussed later).

Some of these components may also be partially implemented depending on the deployment model used for Cloud.

Introduction to Cloud Computing

The function of each module or components of Cloud is as follows

- **Provisioning and Configuration** : Provisioning and Configuration layer forms the lowest layer of cloud and typically reside on bare hardware (as firmware) or on the top of the hypervisor layer. The function of this layer is to abstract the underlying hardware and provide a standard mechanism to spawn instance of virtual machines on demand. The layer also handles the post-configuration of the operating systems and applications residing on the VM.
- **Monitoring and Optimization** : This layer handles all the monitoring of all server, storage, networking and application components in Cloud. Based on this statistics, it could perform routine functions that optimize the behavior of the infrastructure components and provide relevant data to the cloud administrator to further optimize the configuration for maximum utilization and performance.
- **Metering and Chargeback** : This layer provides functions to measure the usage of resources in Cloud. The metering module collects all the utilization data per domain per user. The module gives the Cloud administrator enough data to measure ongoing utilization of resources and to create invoices based on the usage on a periodic basis.
- **Orchestration** : Orchestration is central to Cloud operations. Orchestration converts requests from the Service management layer and the monitoring, chargeback modules to appropriate action items which are then submitted to provisioning and configuration module for final closure. Orchestration updates the CMDB (Configuration Management Database) in the process.
- **Configuration Management Database (CMDB)** : CMDB is a central configuration repository wherein all the meta-data and configuration of different modules, resources is kept and updated on a real-time basis. The repository can then be accessed using standard protocols like SOAP by third party software and integration components. All updates to CMDB happen in real-time as requests get processed in Cloud.

Introduction to Cloud Computing

- **Cloud Lifecycle Management Layer (CLM)** : The Layer handles the coordination of all other layers in Cloud. All requests internal and external are addressed to the CLM layer first. CLM may internally route requests and actions to other layers for further processing.
- **Service Catalog (SC)** : Service Catalog is central to the definition of Cloud. SC defines what kind of services the cloud is capable of providing and at what cost to the end-user. SC is the first thing that is drafted before a Cloud is architected. The Service Management Layer consults SC before it processes any request for a new resource.

Introduction to Cloud Computing

Benefits of Cloud

Cloud provides tangible business benefits to business. It saves cost to the business thus improving the bottom line. It also adds value to the existing business processes by incorporating new functions for increasing efficiency, flexibility, manageability and improved transparency.

Listed below are some of the major benefits of Cloud

- Provisioning is automated and on-demand and can be done on a self-service basis. The provisioning typically takes from few seconds to few hours. Also, the demand for resources can be estimated well in advance to plan for procurement of hardware.
- Utilization typically is around 60-70%. The freed up resources goes into a pool which can be assigned transparently to other users. It is also possible to downscale and upscale based on demand.
- Elasticity allows scaling up and down on-demand.
- Scalability is on-demand. Capacity can be planned and operational expenses can be fine-tuned to meet the current demand.
- Availability: Typically the VM instances are not tied to any particular hardware. Rather they are designed to run over a range of hardware. Hence, it is possible to restart the instance on secondary hardware if the primary hardware fails. This happens automatically and transparently.
- Capital Expenditure: Depending on the cloud deployment, a customer saves about 40% in upfront capital expenditure required to procure hardware. The operational expense can be further fine-tuned based on demand, thus resulting in much higher savings.

Introduction to Cloud Computing

- Chargeback allows for more granular monitoring of usage of resources in terms of cost. This opens up avenues for further optimization.
- Monitoring in cloud allows for further optimization of resources for maximum utilization and reduced wastage. This results in higher savings over time.

Based on the cloud implementation, the savings in Cloud may vary. There are multiple factors that may impact the features of Cloud and the savings and benefits inherent in Cloud.

Introduction to Cloud Computing

Cloud Delivery and Deployment Models

Cloud may be classified based on the structure and the layer at which these operate.

Deployment Models

Based on how and where the Cloud is set up and deployed, the cloud may be classified into three major deployment models

- **Private Cloud** : An organization may choose to build a Cloud within their datacenter. The organization purchases own hardware and software to set up the Cloud. The main intention behind this kind of Cloud is to deliver cloud service to internal departments within the organization. Security could be a major factor contributing to the decision to set up a cloud in-house. This type of cloud is known as Private Cloud.
- **Public Cloud** : This is a more general form of Cloud. It is deployed to provide cloud-services to the general population irrespective of their organization affiliation. The services are generally available through a website using an on-demand payment and subscription mechanism. Public Cloud is considered less secure than Private Clouds. From an end-user perspective, there's no capital expenditure involved in setting up a public cloud. The end-user pays only a monthly subscription fee based on the usage.
- **Hybrid Cloud** : As the name indicates, the Cloud is set up to handle a fraction of the workload on Private Cloud and a fraction of the workload on the Public Cloud. Typically, a customer would normally place their production workload on the Private Cloud and use Public Cloud for development and test environments. The workload can be moved between the Private and Public section of the Hybrid Cloud based on demand. The non-production workload on the public cloud moves back to the private cloud when the private cloud is less loaded. Hybrid Clouds combine the benefits of public and private cloud and help further optimize the capital and operational expenses of running a workload.

Introduction to Cloud Computing

Delivery Models

Another classification of Cloud is based on the layer at which a cloud service is delivered. Based on the delivery model, the cloud may be classified into four major categories

- **IaaS (Infrastructure as a Service)** : The cloud provides infrastructure services to the end-user on a subscription basis. The infrastructure services may include custom designed virtual machines, storage and backup infrastructure, tape backup as service etc. This type of cloud deals primarily with the hardware resources and services. Some common examples are Amazon EC2, Rackspace etc.
- **PaaS (Platform as a Service)** : This type of cloud provides platform services to end-users on a subscription basis. The platform services may include the webserver stacks, middleware or other similar platforms. Some common examples are: the apache-php-perl webserver platform, IBM websphere as a service, tomcat servlet container as a service, moodle LMS, Microsoft Azure etc.
- **SaaS (Software as a Service)** : This type of cloud provides software services to end-users on a subscription basis. The end-user customer is not required to maintain any license for using the cloud provided software. All updates to the software is taken care by the cloud service provider. For example: Google Apps etc.
- **BPaaS (Business Process as a Service)** : This type of cloud provides a particular business process as a service along with the staff that is required to run the process activities. The end-user is not required to hold any license or hire any staff for using the Business Process services. This kind of service provides an end-to-end business process coverage for a business on an on-demand subscription basis. For example: Providing business analytics as a service to end-customers, tailor made for the business.

Introduction to Cloud Computing

Cloud Transformation Roadmap

An organization planning to transform their infrastructure to Cloud move through three major stages of evolution

- 1. Consolidation and Virtualization :** This stage involves moving from a pure physical infrastructure to a virtual infrastructure. This is achieved through virtualization and consolidation of all physical assets. At this stage, basic asset management, monitoring and optimization can be built in using the tools provided by virtualization. The incident, change, asset management remains the same as it was done for the physical infrastructure.
- 2. Automation and Optimization :** Over time, the customer would want to optimize and automate their complete infrastructure to further reduce provisioning and turnaround times. Automation also allows for ease in moving the resources on an on-demand basis. A centralized database to record the configuration of various components in the infrastructure is set up. This is known as the configuration management database or CMDB. If there are multiple datacenters, a federated CMDB may be setup to achieve the same. Over time, the requests for resources may be standardized into templates. A user may request a service from the templates to get faster service and reduce time spent on describing the service request. These templates are collated in the form of service catalog.

The customer may improvise the methods used for monitoring and centralize the same to record information in the CMDB. An event driven mechanism to automate the processing of the monitoring and configuration events may be set up to enable faster resolution of problems in the infrastructure. A basic per-user usage tracking mechanism may be set up at this stage to track individual utilization.

Introduction to Cloud Computing

- 3. Integration of Service Management :** The last stage involves integration of Service Management processes with Cloud. Service Management involves setting up a Service Catalog for the services available in the Cloud. These services may be used on a self-service basis and on-demand. The stage involves integration and working of all the eight layers of the cloud architecture.

Thank You

